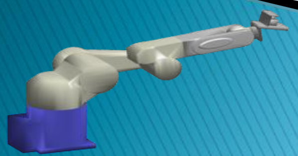
Week 2

Schedule

- Practice Quiz in private channel with designated tutor
  - Tutor create Zoom and post it (or Teams group video chat)
  - Individually do the individual quiz
  - Practice Teams quiz in Zoom breakout groups
- Lab 1 exercise revision and Q&A
- Introduce Lab Assignment 1
- Working through Lab 2 exercise together

Quizzes (weeks 3,5,7,9)

- Quiz starts right at the beginning of class time
- Log into personal PC and setup
- Must share your video feed with your tutor's private channel
- Individual Quiz
  - First 30 minutes
  - Worth 4%
- Team Quiz (groups of 2, **max 1 group of 3**)
  - Next 20 minutes
  - worth 1%
- Must attain 80% in every quiz (can redo if required)
- Mark for subject is only the first in-class attempt
- Practice quiz in Week 2
- Onramp course before Quiz 1



Overview of Week 3: Quiz 1

- To access
  - Must do Matlab Onramp & upload PDF certificate to Onramp assignment

- Two quizzes worth 5% in total
- Individual quiz
  - worth 4%
  - will go from lab start time for 30 minutes
  - 10 questions (1 question from each category)
  - No talking
- Group quiz
  - worth 1%
  - will start immediately after the individual quiz
  - will go for 20 minutes
  - Groups of 3 people or less
  - 20 questions (2 question from each category)
  - Lots of talking within group

Question Categories (10)

- Rotation Matrix
- Translation Matrix
- Chain transformation matrix multiplication
  - Translation then Rotation
  - Rotation then translation
- Euler Theorem
- Car On a Track
- UAV (Unmanned Aerial Vehicle)
  - Safe
  - Unsafe
- Teach Control of Robot Model
- Robot Model Joint Limits

…but how can I prepare?

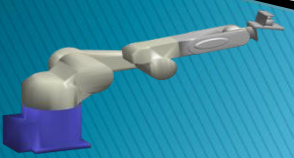
- Complete Expected Prework for Labs 1 & 2
- That is:
  - Be familiar with assumed Math & Matlab material
  - Download & install Matlab Robotics Toolbox
  - Read textbook
    - Chapter 1 (pages 1–6): “Introduction”
    - Sections 2, 2.1, 2.2, 2.3 (pages 15–41): “Representing Position and Orientation”, and
    - Section 7.1 (pages 137–141): “Describing a Robot Arm”
  - Watch video on D&H Parameters
  - Complete and understand the Week 1 Lab Exercises
  - Attempt the Lab 2 Exercises

Lab 1 Review (Q1)

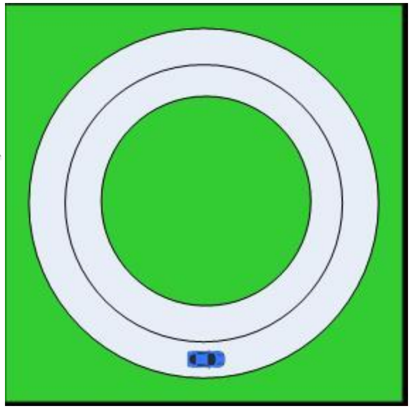
1. Start Matlab

2. Download and install Peter Corke's modified robotics toolbox (See UTSOnline)

3. Run the demos
   (type "rtbdemo")
   to show the arms and get an idea what is possible.



Lab 1 Review (Q2) Car driving on a circular track



1. Download and show the image
2. Plot a transform representing the car
3. Change the leg lengths of the transform
4. Incrementally multiply car transform by offset transforms

```
car1Tr = se2(300, 550, 0);
```

Then iteratively update $T_{car} = T_{car}T_{move}T_{turn}$,

where T_{move} is move forwards and T_{turn} means turn

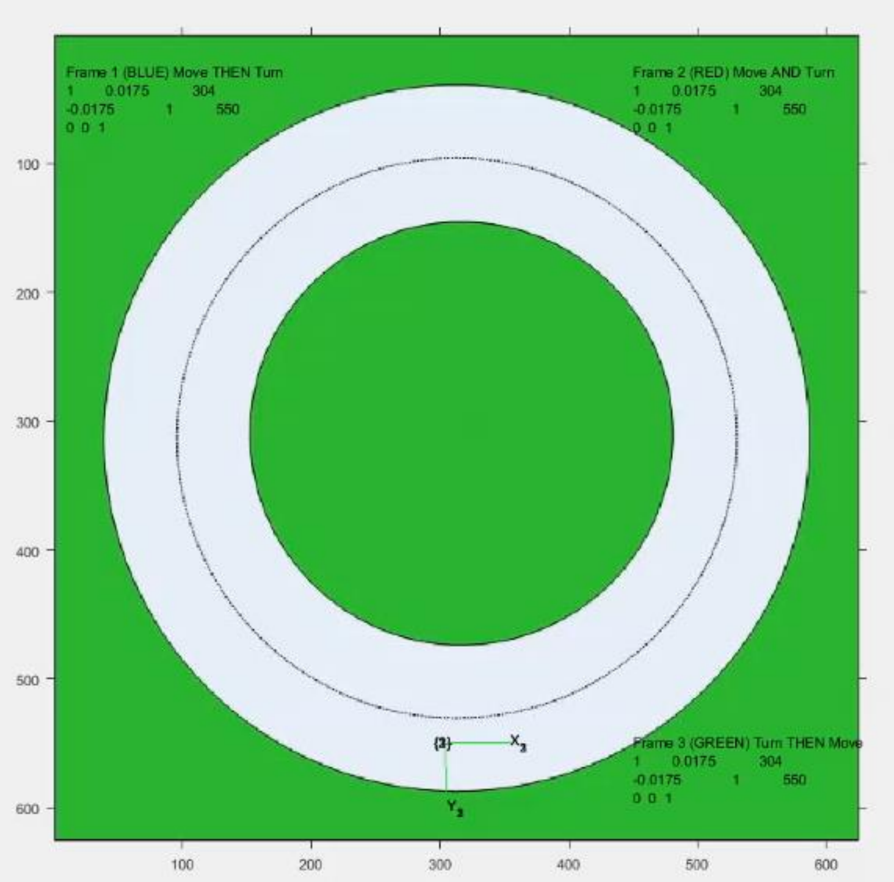
```
car1MoveTr = se2((pi * 484)/Increments, 0, 0);
car1TurnTr = se2(0, 0, -2*pi/ Increments);
```

Why not use the equation of a circle?
(how about weird shaped tracks? sensors inputs?)

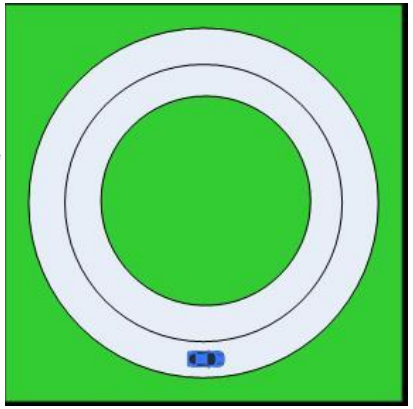
Does $T_{car}T_{move}T_{turn} = T_{car}T_{turn\&move} = \cancel{T_{car}T_{turn}T_{move}}$? where

```
car1MoveAndTurnTr = se2((pi * 484)/Increments, 0, -2*pi/Increments);
```

Interestingly Rotations (top left 2x2) same, but Position (top right 1x2) is different.
All the same after the full revolution



Lab 1 Review (Q2) Car driving on a circular track



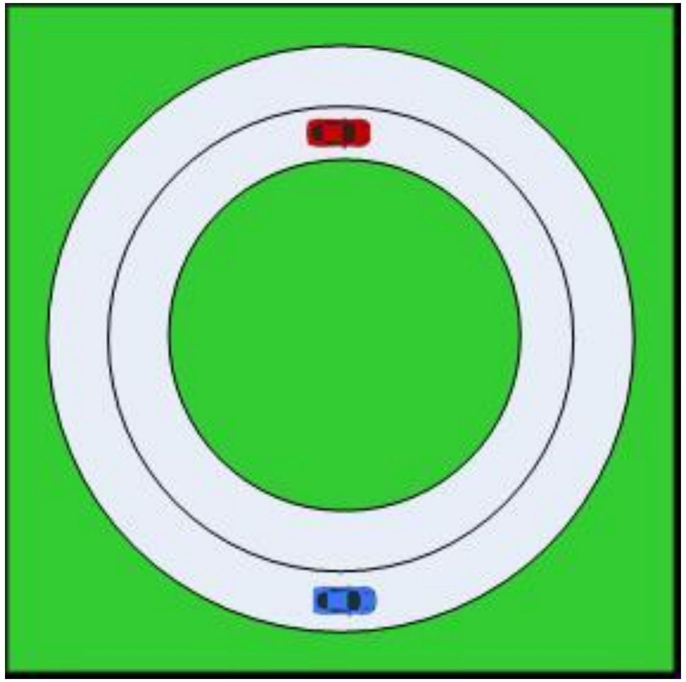
5. Display the transform of the car using the Matlab “text”.
   Note that plots can be cleared:
   - h = text(0,0,’hi’)
   - try delete(h); end

Command Window

New to MATLAB? See resources for Getting Started.

```
>> help trplot2
 trplot2 Plot a planar transformation

  trplot2(T, OPTIONS) draws a 2D coordinate frame represented by the SE(2)
  homogeneous transform T (3x3).

  H = trplot2(T, OPTIONS) as above but returns a handle.

  trplot2(H, T) moves the coordinate frame described by the handle H to
  the SE(2) pose T (3x3).

  Options::
  'axis',A            Set dimensions of the MATLAB axes to A=[xmin xmax ymin ymax]
  'color', c          The color to draw the axes, MATLAB colorspec
  'noaxes'            Don't display axes on the plot
  'frame',F           The frame is named {F} and the subscript on the axis labels is F.
  'text_opts', opt    A cell array of Matlab text properties
  'handle', h         Draw in the MATLAB axes specified by h
  'view',V            Set plot view parameters V=[az el] angles, or 'auto'
                      for view toward origin of coordinate frame
  'length',s          Length of the coordinate frame arms (default 1)
  'arrow'             Use arrows rather than line segments for the axes
  'width', w          Width of arrow tips

  Examples::

        trplot2(T, 'frame', 'A')
        trplot2(T, 'frame', 'A', 'color', 'b')
        trplot2(T1, 'frame', 'A', 'text_opts', {'FontSize', 10, 'FontWeight', 'bold'})

  Notes::
  - The arrow option requires the third party package arrow3 from File
    Exchange.
  - When using the form TRPLOT(H, ...) to animate a frame it is best to set
    the axis bounds.

  See also trplot.
```

Lab 1 Review (Q3) Another car….



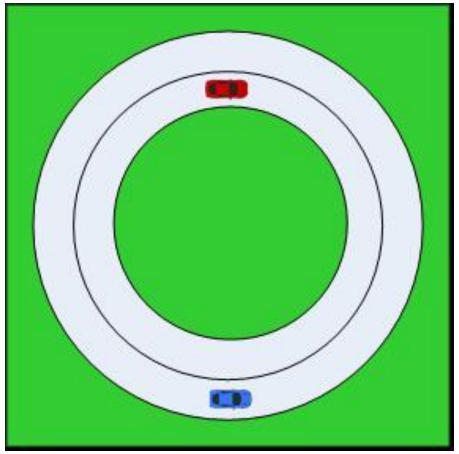
1. Plot transform for a second car

2. Make the second car incrementally drives in the opposite direction to the first car

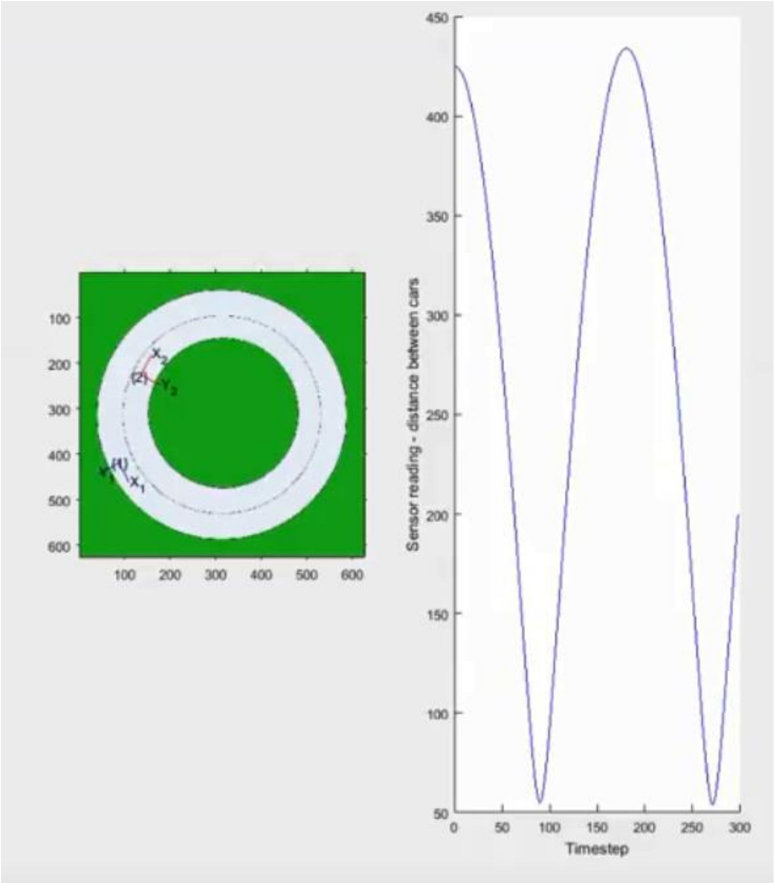
3. Determine the relative transform between the two cars at each time step

$$T_{car1to2} = T_{car1}^{-1} T_{car2} \quad \text{so} \quad T_{car2} = T_{car1} T_{car1to2} = \cancel{T_{car1}} \cancel{T_{car1}^{-1}} T_{car2}$$
$$T_{car2to1} = T_{car2}^{-1} T_{car1} \quad \text{so} \quad T_{car1} = T_{car2} T_{car2to1} = \cancel{T_{car2}} \cancel{T_{car2}^{-1}} T_{car1}$$

Lab 1 Review (Q4) Bonus….

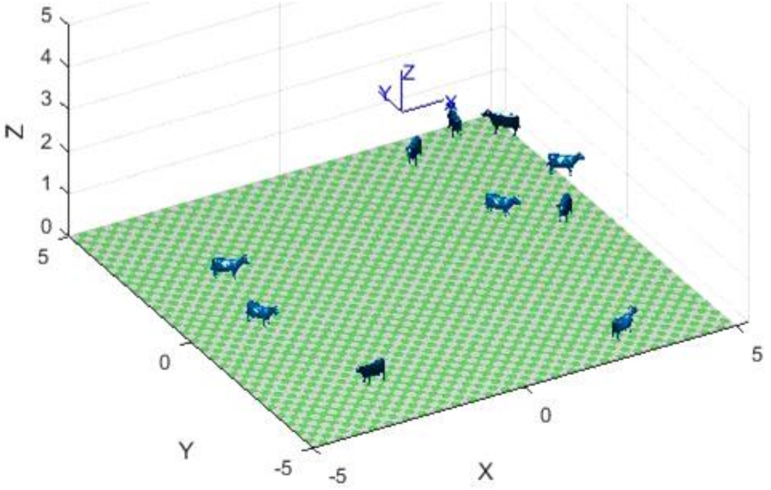


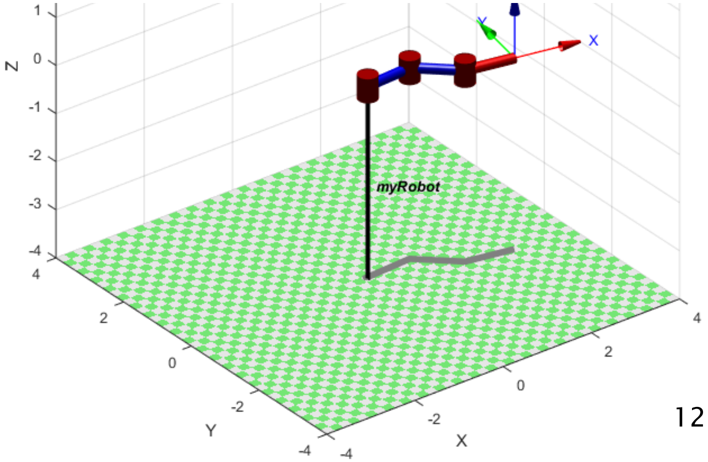
- Plot a graph of the distances between the two cars as they drive around the track
  - Do movements from Q3
  - Compute distance between two points
  - Plot on a graph



Introduction to Lab 2

1. Consider an UAV flying to monitor cattle, and plot a 3D transform
2. Plot a cow herd and make the cows move
   Cow model is “cow.ply”
3. Plot the cows and the UAV (a coordinate frame moving in 3D), get transform between them, then track
4. Let’s plot and move a Robot given DH parameters!

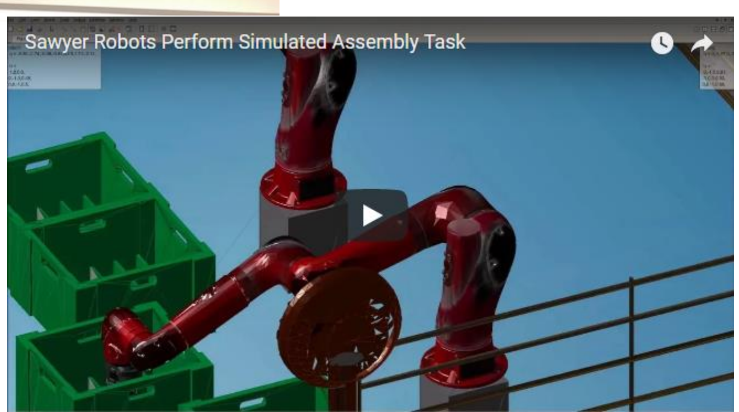




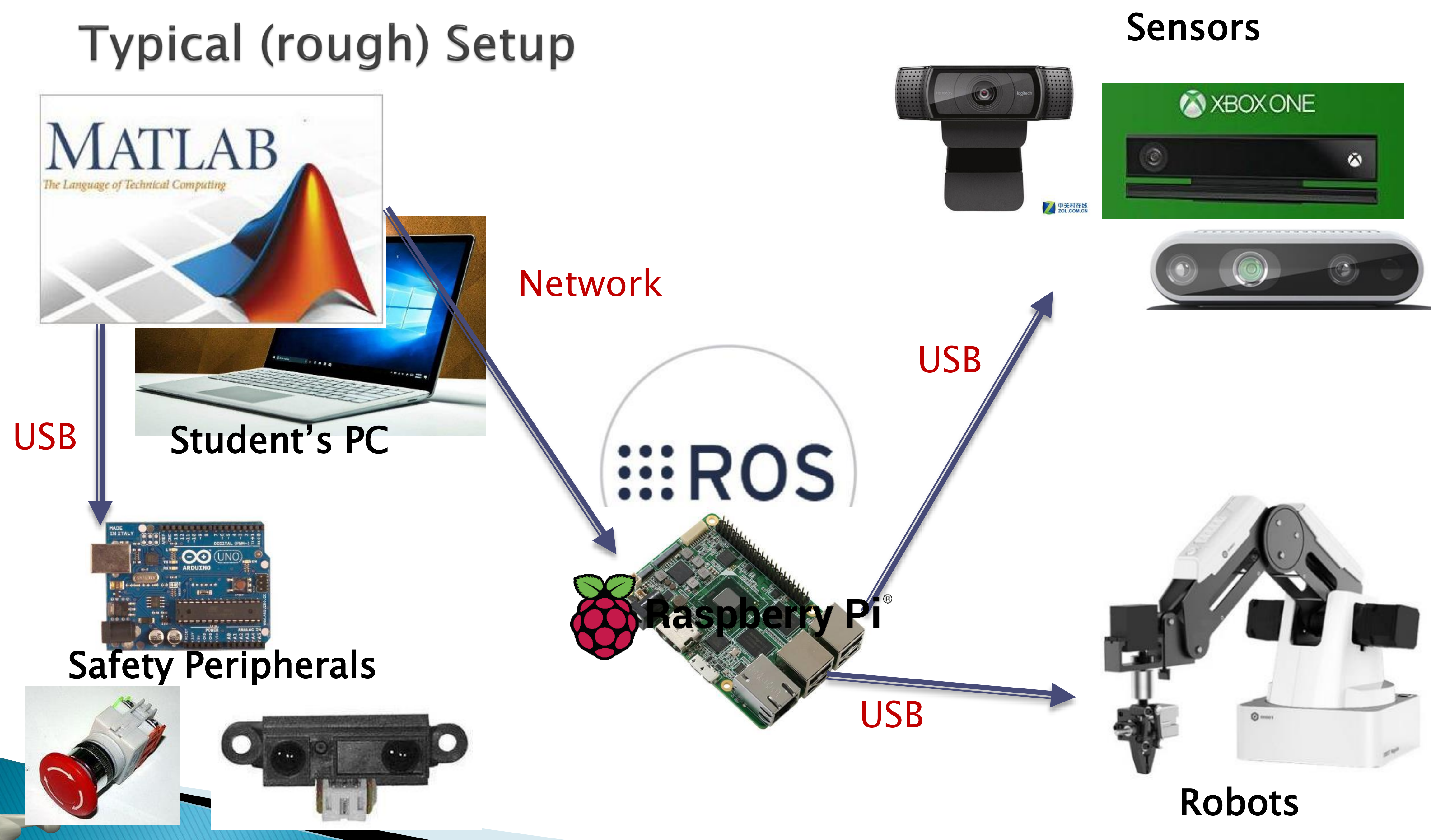
Lab Assignment 1

- Weight: 20%.
  Demo Due: Week 6 – Lab Class
  Report Due: Week 6 – 23:59 Friday
- Build a model of the robot(s), parts and environment.
- Simulate assembling plastic housing using multiple UR3 Robots
- Robot and Parts' locations will be given on the demo day

- Video compilation (from Spring 2016)
  http://youtu.be/ayzs0ZrseOw



Typical (rough) Setup



Week 2: Pre-work before Week 3 lab

- Completed week 2 lab exercises
- Read textbook
  - Section 7.2 (pages 141-146): “Forward Kinematics”
- Watch videos on D&H parameters and Forward Kinematics
  - Part 1
  - Part 2
- Attempt Week 3 lab exercises
- Work on Lab Assignment #1 (due week 6)

Note/slides from the textbook

- Textbook readings for week 2:
  - Sections 2.1, 2.2, 2.3 (pages 19–41): “Representing Position and Orientation”, and
  - Section 7.1 (pages 137–141): “Describing a Robot Arm”

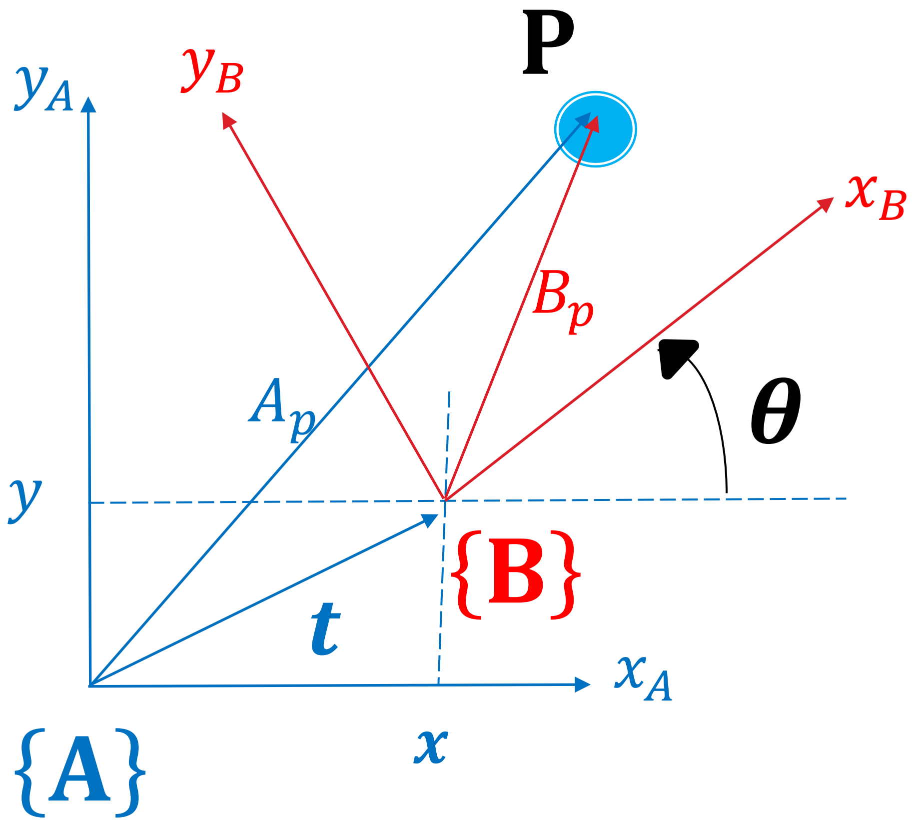
- Although it is better to read the textbook, some notes (in slide format) have been summarised below

2.1 Representing Pose in 2-Dimensions

- We use a Cartesian coordinate system or coordinate frame with orthogonal axes denoted x and y and typically drawn with the x-axis horizontal and the y-axis vertical.
  - The point of intersection is called the origin.
  - Unit-vectors parallel to the axes are denoted $\hat{x}$ and $\hat{y}$.
  - A point is represented by its x- and y-coordinates (x, y) or as a bound vector

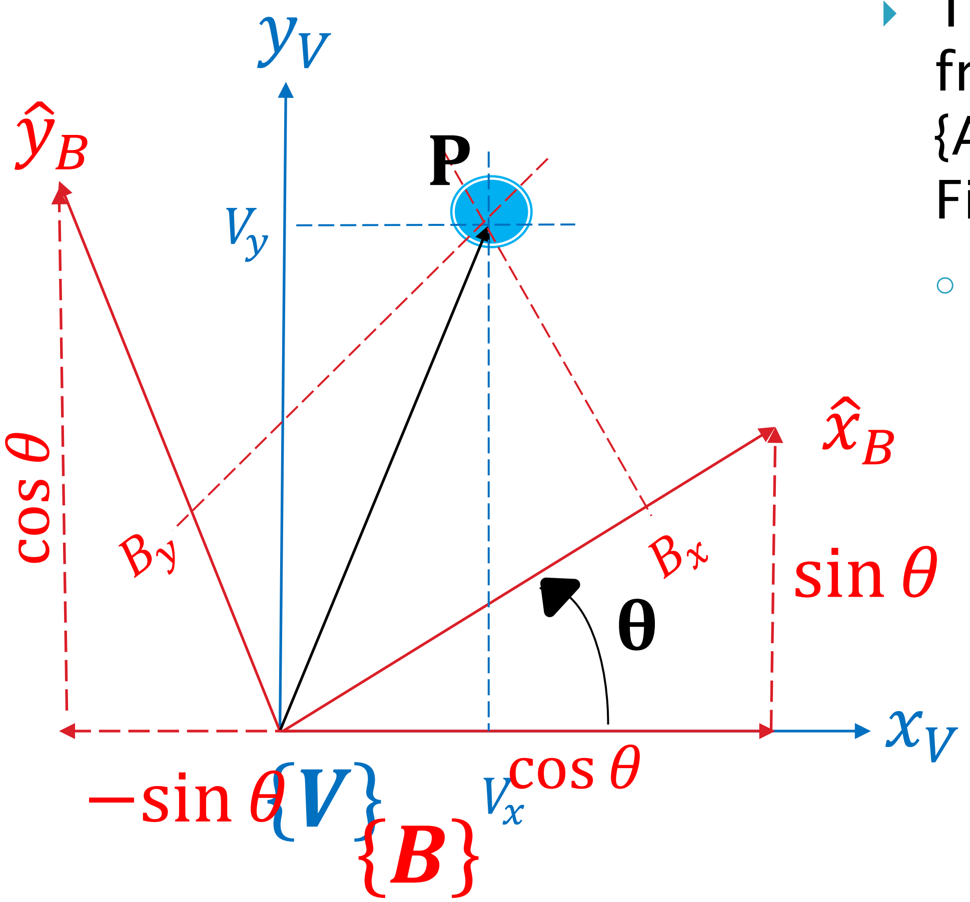
$$p = x\hat{x} + y\hat{y} \qquad (2.2)$$

2.1Representing Pose in 2-Dimensions (continued...)



- The origin of {B} has been displaced by the vector t =(x, y) and then rotated counter-clockwise by an angle θ.
  - 3-vector $A\xi_B$ ~(x, y, θ), and we use the symbol ~ to denote that the two representations are equivalent since $(x_1, y_1, \theta_1) \oplus (x_2, y_2, \theta_2)$ is a complex trigonometric function of both poses.
  - The approach is to consider an arbitrary point P with respect to each of the coordinate frames and to determine the relationship between A_p and B_p.

Product of a Row and a Column Vector



- To consider just rotation we create a new frame {V} whose axes are parallel to those of {A} but whose origin is the same as {B}, see Fig. 2.7.
  - According to Eq. 2.2 we can express the point P with respect to {V} in terms of the unit-vectors that define the axes of the frame

$$
\begin{aligned}
v_p &= v_{x\hat{x}_v} + v_{y\hat{y}_v} \\
&= (\hat{x}_v \;\; \hat{y}_v) \begin{bmatrix} v_x \\ v_x \end{bmatrix}
\end{aligned}
$$

which we have written as the product of a row and a column vector.

Product of a Row and a Column Vector (continued...)

- The coordinate frame {B} is completely described by its two orthogonal axes which we represent by two unit vectors

$$\hat{x}_B = \cos\theta\,\hat{x}_v + \sin\theta\,\hat{y}_v \qquad (2.3)$$
$$\hat{y}_B = \sin\theta\,\hat{x}_v \quad \cos\theta\,\hat{y}_v$$

which can be factorized into matrix form as

$$(\hat{x}_B \;\; \hat{y}_B) = (\;\hat{x}_v \;\; \hat{y}_v)\begin{bmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{bmatrix} \qquad (2.4)$$

Product of a Row and a Column Vector (continued…)

- Using Eq. 2.2 we can represent the point P with respect to {B} as

$$B_p = B_{x\hat{x}_B} + B_{y\hat{y}_B}$$

$$= \begin{pmatrix} \hat{x}_B & \hat{y}_B \end{pmatrix} \begin{bmatrix} B_x \\ B_y \end{bmatrix}$$

and substituting Eq. 2.4 we write

$$B_p = \begin{pmatrix} \hat{x}_V & \hat{y}_V \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{bmatrix} B_x \\ B_y \end{bmatrix} \qquad (2.5)$$

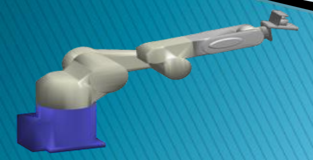
Product of a Row and a Column Vector (continued…)

- Now by equating the coefficients of the right-hand sides of Eq. 2.3 and Eq. 2.5 we write

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{bmatrix} B_x \\ B_y \end{bmatrix}$$

which describes how points are transformed from frame {B} to frame {V} when the frame is rotated. This type of matrix is known as a rotation matrix and denoted $V_{\boldsymbol{R}_B}$

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = V_{\boldsymbol{R}_B} \begin{bmatrix} B_x \\ B_y \end{bmatrix} \qquad (2.6)$$



The Rotation Matrix V_{R_B} Properties

- The rotation matrix V_{R_B} has some special properties.
  - Firstly it is orthonormal (also called orthogonal) since each of its columns is a unit vector and the columns are orthogonal.
    - In fact the columns are simply the unit vectors that define {B} with respect to {V} and are by definition both unit-length and orthogonal.
  - Secondly, its determinant is +1, which means that R belongs to the special orthogonal group of dimension 2 or
    $R \in SO(2) \subset R2 \times 2$.
    - The unit determinant means that the length of a vector is unchanged after transformation, that is, $|B_p| = |V_p| \forall \theta$.

Orthonormal Matrices

- Orthonormal matrices have the very convenient property that $R^{-1} = R^T$, that is, the inverse is the same as the transpose.
- We can therefore rearrange Eq. 2.6 as

$$\begin{bmatrix} B_x \\ B_y \end{bmatrix} = (V_{\boldsymbol{R}_B})^{-1} \begin{bmatrix} V_x \\ V_y \end{bmatrix} = (V_{\boldsymbol{R}_B})^T \begin{bmatrix} V_x \\ V_y \end{bmatrix} = B_{\boldsymbol{R}_V} \begin{bmatrix} V_x \\ V_y \end{bmatrix}$$

- Note that inverting the matrix is the same as swapping the superscript and subscript, which leads to the identity $R(-\theta) = R(\theta)^T$.

Translation Between The Origins Of The Frames

- The second part of representing pose is to account for the translation between the origins of the frames shown in Fig. 2.6.
- Since the axes {V} and {A} are parallel this is simply vectorial addition

$$\begin{bmatrix} A_x \\ A_y \end{bmatrix} = \begin{bmatrix} V_x \\ V_y \end{bmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} \qquad (2.7)$$

$$= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{bmatrix} B_x \\ B_y \end{bmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} \qquad (2.8)$$

$$= \begin{pmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \end{pmatrix} \begin{bmatrix} B_x \\ B_y \\ 1 \end{bmatrix} \qquad (2.9)$$

2.1Representing Pose in 2-Dimensions (continued...)

- or more compactly as

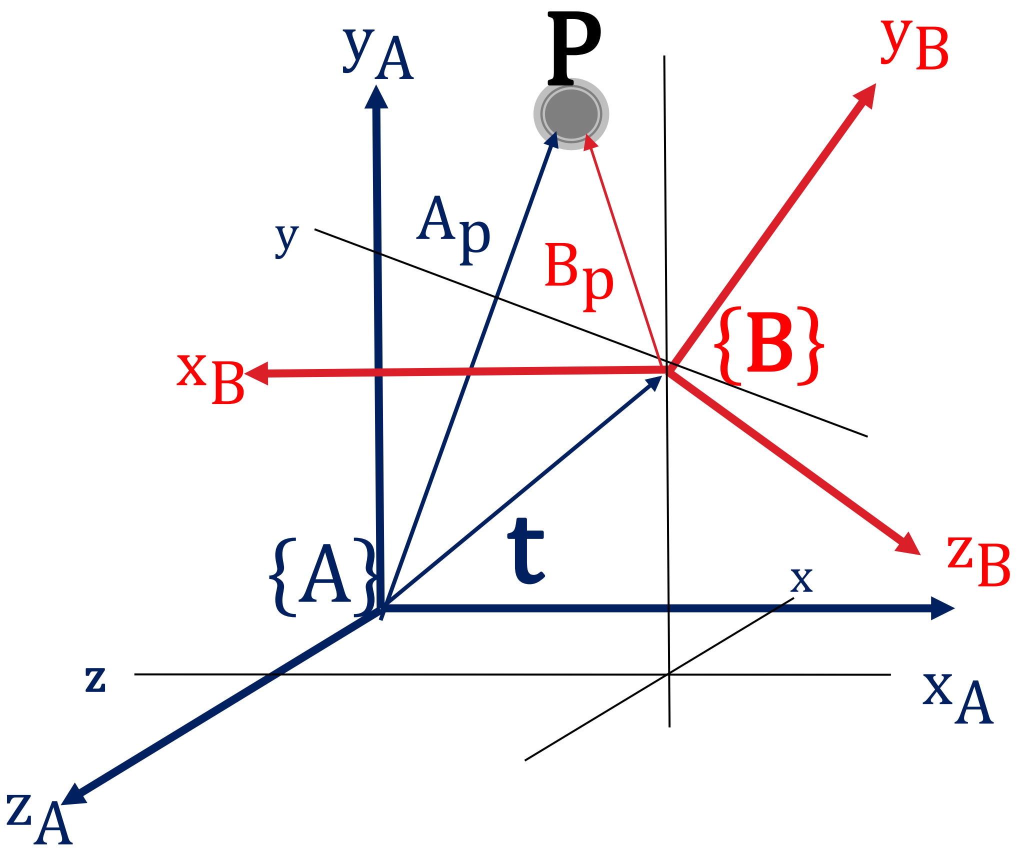
$$\begin{bmatrix} A_x \\ A_y \\ 1 \end{bmatrix} = \begin{bmatrix} A_{\boldsymbol{R}_B} & \boldsymbol{t} \\ 0_{1X2} & 1 \end{bmatrix} \begin{bmatrix} B_x \\ B_y \\ 1 \end{bmatrix} \qquad (2.10)$$

where t =(x, y) is the translation of the frame and the orientation is $A_{\boldsymbol{R}_B}$.

- Note that $A_{\boldsymbol{R}_B} = T_{\boldsymbol{R}_B}$ since the axes of {A} and {V} are parallel, coordinate vectors for point P are now expressed in homogenous form and we write

$$A_{\tilde{\boldsymbol{p}}} = \begin{bmatrix} T_{\boldsymbol{R}_B} & \boldsymbol{t} \\ 0_{1X2} & 1 \end{bmatrix} B_{\tilde{\boldsymbol{p}}} \; = A_{\boldsymbol{T}_B} B_{\tilde{\boldsymbol{p}}}$$

Two 3D Coordinate Frames



- Two 3D coordinate frames {A} and {B}. {B} is rotated and translated with respect to {A}

${A}_{T_B}$ Relative Pose Representation

- and ${A}_{T_B}$ is referred to as a homogeneous transformation. The matrix has a very specific structure and belongs to the special Euclidean group of dimension 2 or T ∈ SE(2) $\subset R^{3X3}$.

- By comparison with Eq. 2.1 it is clear that ${A}_{T_B}$ represents relative pose

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

A Concrete Representation of Relative Pose

- A concrete representation of relative pose ξ is ξ ~T ∈ SE(2) and $\boldsymbol{T_1} \oplus \boldsymbol{T_2} \mapsto \boldsymbol{T_1 T_2}$ which is standard matrix multiplication.

$$\boldsymbol{T_1 T_2} = \begin{bmatrix} \boldsymbol{R_1} & \boldsymbol{t_1} \\ \boldsymbol{0_{1X2}} & \boldsymbol{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{R_2} & \boldsymbol{t_2} \\ \boldsymbol{0_{1X2}} & \boldsymbol{1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R_1 R_2} & \boldsymbol{R_1 t_2} \\ \boldsymbol{0_{1X2}} & \boldsymbol{1} \end{bmatrix}$$

- One of the algebraic rules from page 18 is ξ ⊕ 0 =ξ.
- For matrices we know that TI =T, where I is the identify matrix, so for pose 0I the identity matrix.
- Another rule was that ξ⊖ξ=0. We know for matrices that $\boldsymbol{TT^{-1}} = 1$ which implies that $\ominus T \mapsto T^{-1}$

$$\boldsymbol{T^{-1}} = \begin{bmatrix} \boldsymbol{R_1} & \boldsymbol{t_1} \\ \boldsymbol{0_{1X2}} & \boldsymbol{1} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{R^T} & \boldsymbol{-R^T t_1} \\ \boldsymbol{0_{1X2}} & \boldsymbol{1} \end{bmatrix}$$

- For a point $\tilde{\boldsymbol{p}} \in P^2$ then $\mathrm{T} \cdot \tilde{\boldsymbol{p}} \mapsto \boldsymbol{T}_{\tilde{\boldsymbol{p}}}$ which is a standard matrix-vector product.

How To Determine Coordinate of the Point

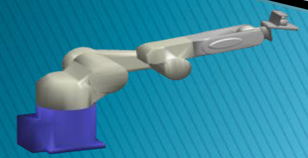
- To determine the coordinate of the point with respect to {1} we use Eq. 2.1 and write down

$$0_{\boldsymbol{p}} = 0_{\xi_1} \cdot 1_{\boldsymbol{p}}$$

- and then rearrange as

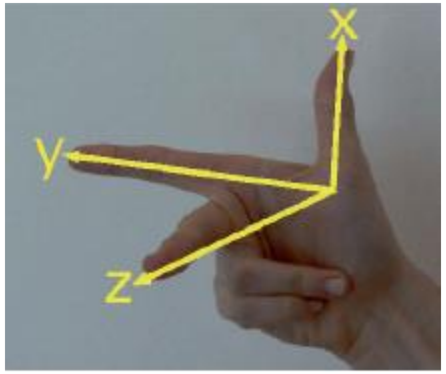
$$1_{\boldsymbol{p}} = 0_{\xi_1} \cdot 1_{\boldsymbol{p}}$$

$$= \left(0_{\xi_1}\right)^{-1} \cdot 0_{\boldsymbol{p}}$$



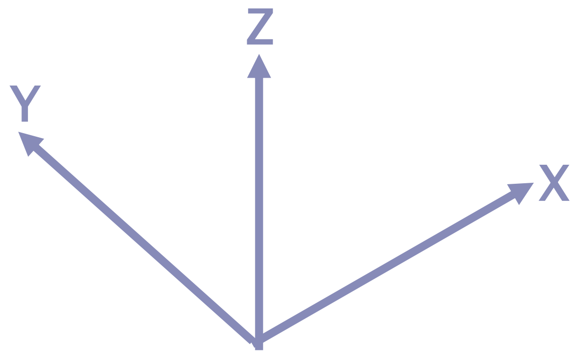
2.2.1 Representing Orientation in 3-Dimensions

- Euler's rotation theorem states that any rotation can be considered as a sequence of rotations about different coordinate axes.
  - Right-hand rule. A right-handed coordinate frame is defined by the first three fingers of your right hand which indicate the relative directions of the x-, y- and z-axes respectively.

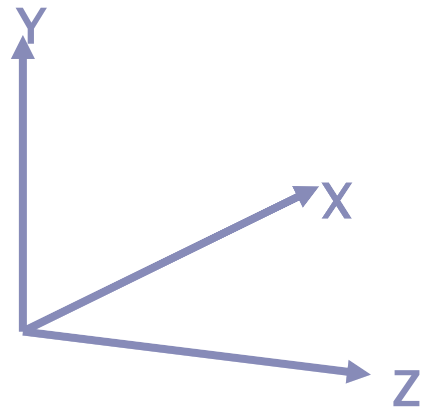


  - The ⊕ operator is not commutative – the order in which rotations are applied is very important.

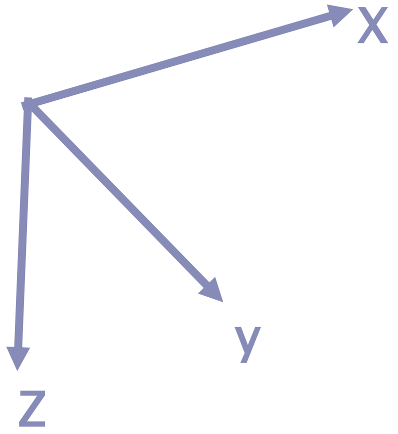
Rotation of a 3D Coordinate Frame



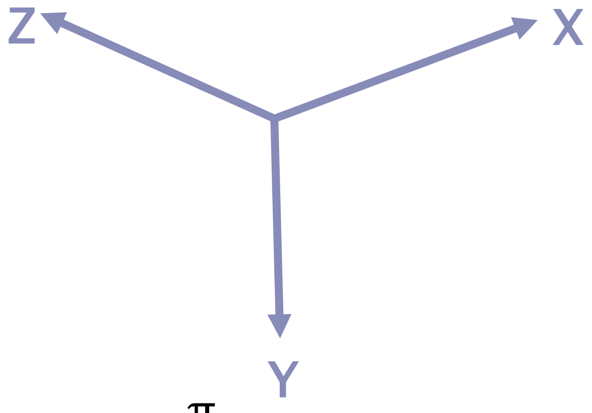
a. Original



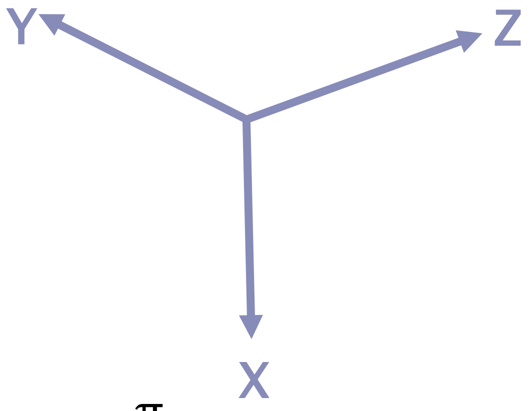
$b. \frac{\pi}{2}$ about x–axis



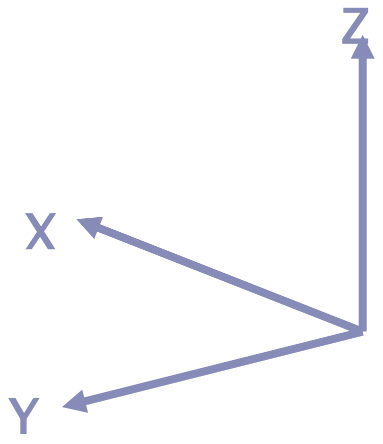
c. π about x$-$axis



d. $-\frac{\pi}{2}$ about x–axis

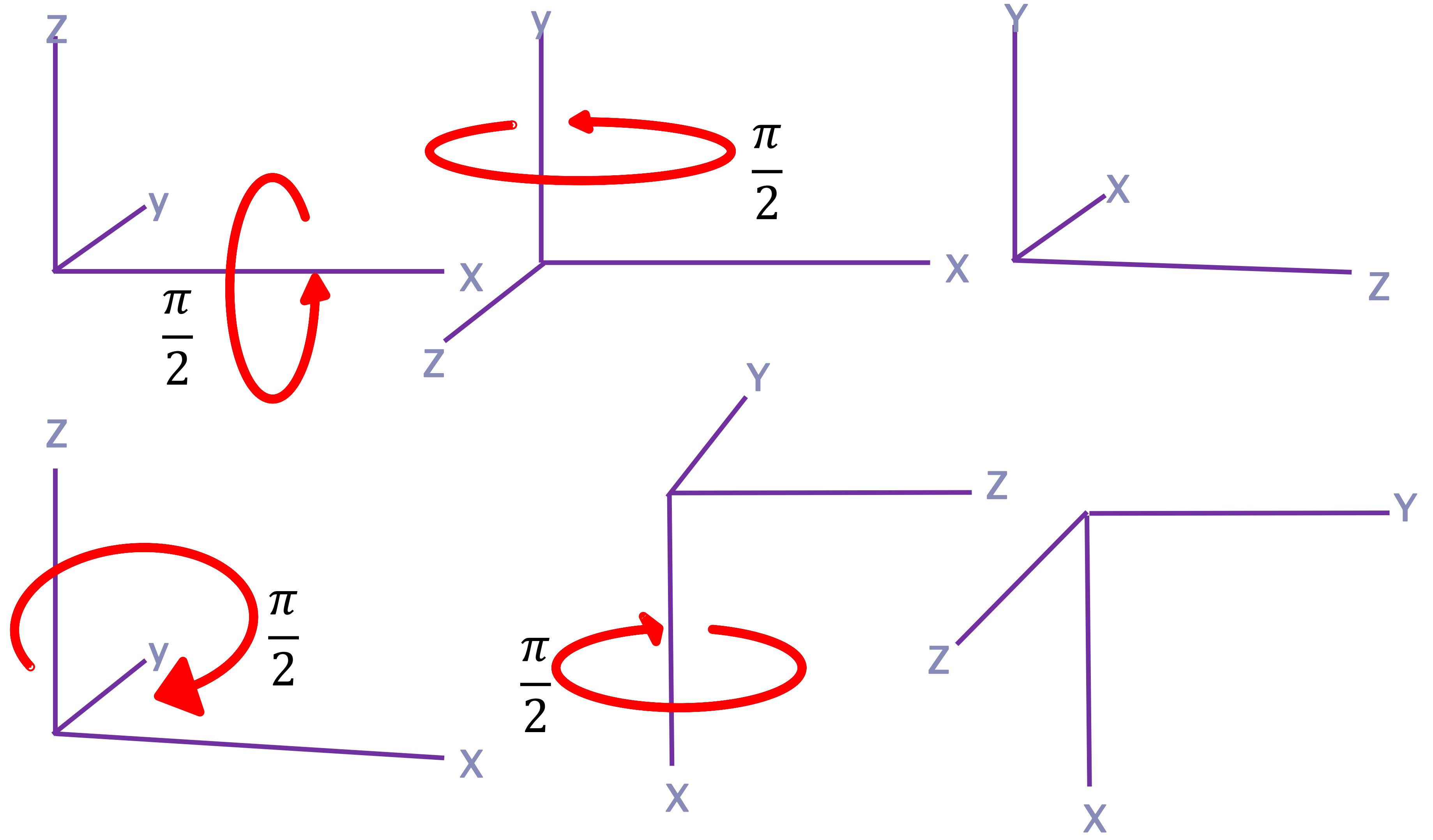


e. $\frac{\pi}{2}$ about y–axis



f. $\frac{\pi}{2}$ about z–axis

Rotation of a 3D Coordinate Frame



2.2.1.1 Orthonormal Rotation Matrix

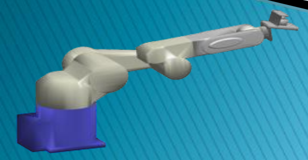
- Each unit vector has three elements and they form the columns of a 3×3 orthonormal matrix ${}^{A}R_{B}$

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = {}^{A}R_{B} \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix}$$

- which rotates a vector defined with frame {B} and {A} respectively.
- The orthonormal rotation matrices for rotation of θ about the x-, y- and z-axes are

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

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



2.2.1.2 Three-Angle Representations

- The Eulerian type involves repetition, but not successive, of rotations about one particular axis: XYX, XZX, YXY, YZY, ZXZ, or ZYZ.
- The Cardanian type is characterized by rotations about all three axes: XYZ, XZY, YZX, YXZ, ZXY, or ZYX.
  - It is common practice to refer to all 3-angle representations as Euler angles but this is underspecified since there are twelve different types to choose from.
  - The ZYZ sequence $R=R_Z(\varphi)R_Z(\theta)\ R_Z(\psi)$ is commonly used in aeronautics and mechanical dynamics
  - The Euler angles are the 3-vector $\Gamma = (\varphi, \theta, \psi)$.

2.2.1.3 Singularities and Gimbal Lock

- A fundamental problem with the three-angle representations just described is singularity.
  - Using the definition of the Lunar module's coordinate system where the rotation of the spacecraft's body-fixed frame {B} with respect to the stable platform frame {S} is

    $$^{S}\mathbf{R}_B = R_y(\theta_p) R_z(\theta_r) R_x(\theta_y)$$

  - For the case when $\theta_r = \frac{\pi}{2}$ we can apply the identity $R_y(\theta) R_z\left(\frac{\pi}{2}\right) = R_z\left(\frac{\pi}{2}\right) R_x(\theta)$ leading to

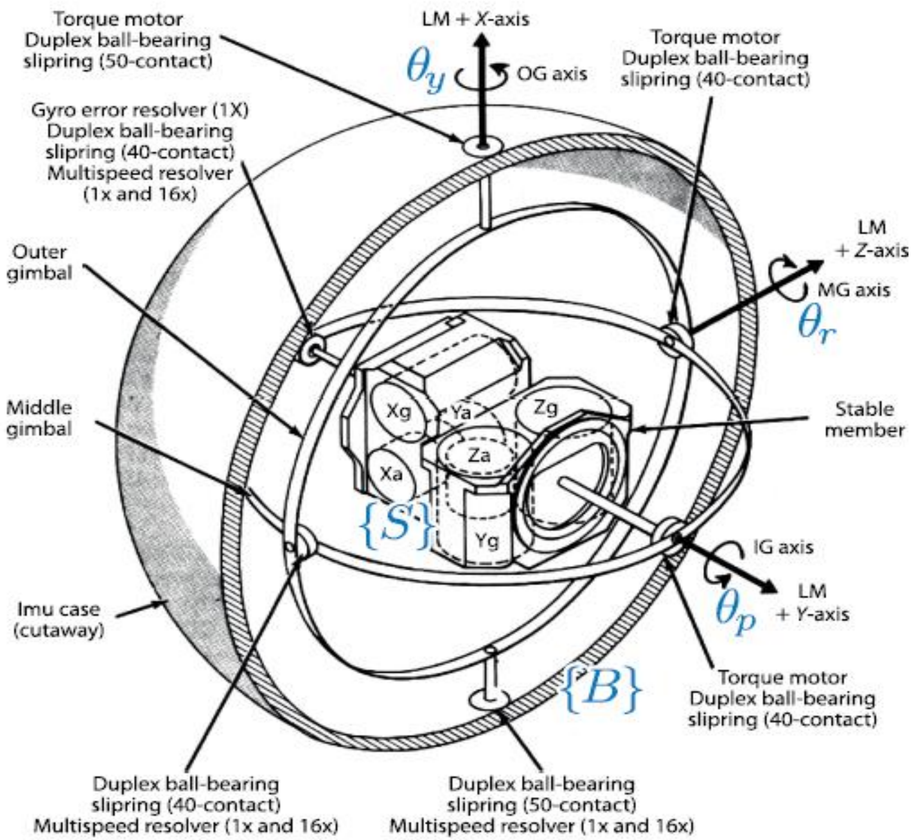
    $$^{S}\mathbf{R}_B = R_z\left(\frac{\pi}{2}\right) R_x(\theta_p) R_x(\theta_y) = R_z\left(\frac{\pi}{2}\right) R_x(\theta_p)$$

    which cannot represent rotation about the *y*-axis.

Schematic of Apollo Lunar Module

- The x–axis pointing up through the thrust axis, the z–axis forward, and the y–axis pointing right.

- Starting at the stable platform $\{S\}$ and working outwards toward the spacecraft's body frame $\{B\}$ the rotation angle sequence is YZX.



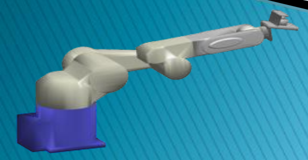
2.2.1.4 Two Vector Representation

- The approach vector: $\hat{o}=(a_x,a_y,a_z)$
- An orthogonal vector: $\hat{o}=(o_x,o_y,o_z)$
  - These two unit vectors are sufficient to completely define the rotation matrix

$$R=\begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}$$

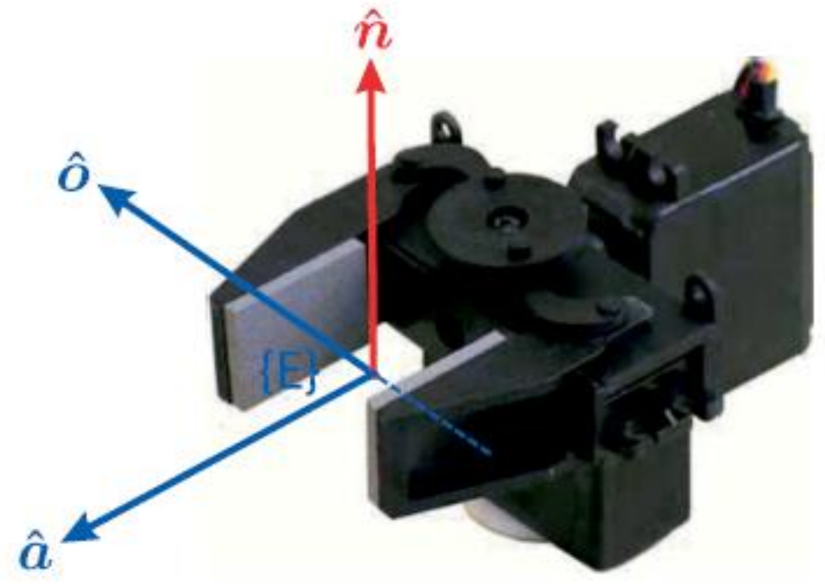
since the remaining column can be computed using $\hat{n}=\hat{o}\times\hat{a}$

  - Even if the two vectors $\hat{a}$ and $\hat{o}$ are not orthogonal they still define a plane and the computed $\hat{n}$ is normal to that plane.
  - In this case we need to compute a new value for $\hat{o}'=\hat{a}\times\hat{n}$ which lies in the plane but is orthogonal to each of $\hat{a}$ and $\hat{n}$.



2.2.1.5 Rotation About an Arbitrary Vector

- Robot end-effector coordinate system defines the pose in terms of an *approach* vector $\hat{a}$ and an *orientation* vector $\hat{o}$, from which $\hat{n}$ can be computed.
- $\hat{n}$, $\hat{o}$ and $\hat{a}$ vectors correspond to the *x*–, *y* and *z*–axes respectively of the end-effect or coordinate frame.

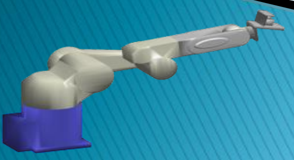


Orthonormal Rotation Matrix

- An orthonormal rotation matrix will always have one real eigenvalue
  - $\lambda = 1$ and a complex pair $\lambda = \cos\theta \pm i \sin\theta$ where θ is the rotational angle.
  - Eigenvalues and eigenvectors: $R_v = \lambda_v$ where v is the eigenvector corresponding to λ.
  - λ. For the case $\lambda = 1$ then $R_v = v$ where eigenvector v is unchanged by the rotation.
- Using Rodrigues' rotation formula

$$R = I_{3\times3} + \sin\theta S(v) + (1 - \cos\theta)(vv^T - I_{3\times3})$$

  - However the direction can be represented by a unit vector with only two parameters since the third element can be computed by $v_3 = \sqrt{1 - v_1^2 - v_2^2}$



2.2.1.6 Unit Quaternions

- The quaternion is an extension of the complex number written as

$$\hat{q} = s + v$$
$$= s + v_1 i + v_2 j + v_3 k$$

  - where $s \in \mathbb{R}$, $v \in \mathbb{R}^3$ and the orthogonal complex numbers i, j and k are defined such that $i^2 = j^2 = k^2 = ijk = -1$
  - We will denote a quaternion as $\hat{q} = s < v_1, v_2, v_3 >$

- These are quaternions of unit magnitude that is, those for which $|\hat{q}| = 1$ or $s^2 + v_1^2 + v_2^2 + v_3^2$
  - The unit-quaternion has the special property:

$$s = \cos\frac{\theta}{2}, v = \left[\sin\frac{\theta}{2}\right]\hat{n}$$

2.2.2 Combining Translation and Orientation

- Composition is defined by

$$\xi_1 \oplus \xi_2 = (t_1 + \hat{q}_1 \cdot t_2, \hat{q}_1 \oplus \hat{q}_2)$$

- and negation is

$$\ominus \xi = \left(-\mathring{q}^{-1} \cdot t_2, \mathring{q}^{-1}\right)$$

- and a point coordinate vector is transformed to a coordinate frame by

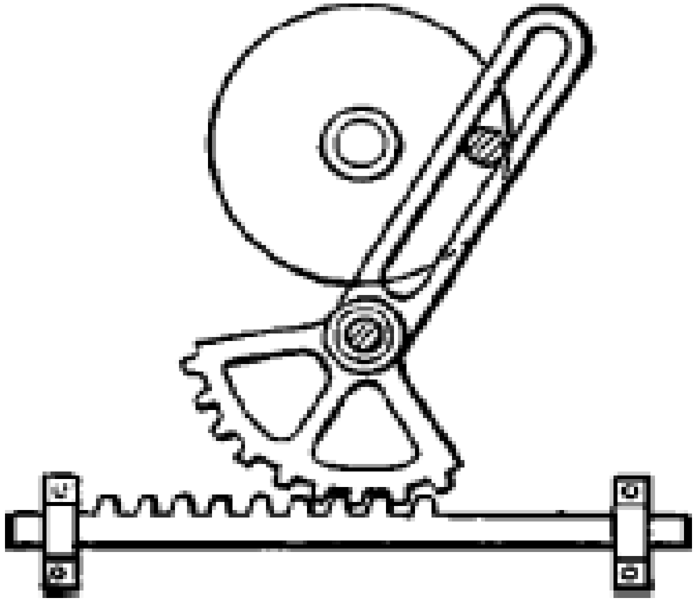
$$x_p = x_{\xi_y} \cdot y_p = \mathring{q} \cdot y_p$$

- derivation is similar to the 2D case of Eq. 2.10 but extended to account for the *z*–dimension

$$\begin{bmatrix} A_x \\ A_y \\ A_z \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{A}\mathbf{R}_B & t \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} B_x \\ B_y \\ B_z \\ 1 \end{bmatrix}$$

What is Kinematics?

- Kinematics is the branch of mechanics that studies the motion of a body, or a system of bodies, without consideration given to its mass or the forces acting on it.
- From the Greek word “motion”



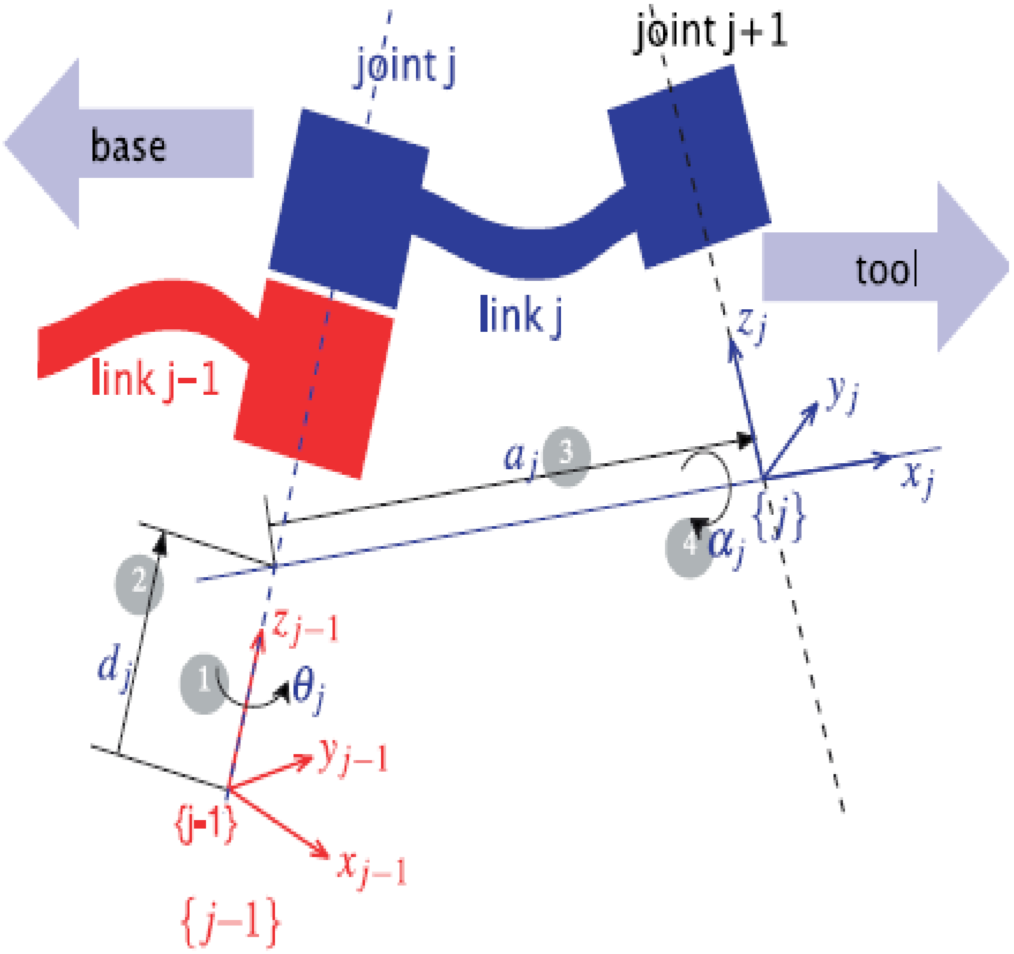
7.1 Describing a Robot Arm

- A serial-link manipulator comprises a set of bodies, called links, in a chain and connected by joints.
- Each joint has:
  - translational (a sliding or prismatic joint)
  - rotational (a revolute joint)
- Motion of the joint changes the relative angle or position of its neighbouring links.

a

b

Definition of Standard Denavit and Hartenberg Link Parameters



- The colors red and blue denote all things associated with links j−1 and j respectively. The numbers in circles represent the order in which the elementary transforms are applied

4x4 Homogenous Transformation

- The transformation from link coordinate frame $\{j-1\}$ to frame $\{j\}$ is defined in terms of elementary rotations and translations as

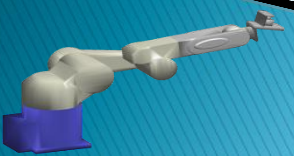
$$^{j-1}A_j\left(\theta_j,d_j,a_j,\alpha_j\right)=T_{Rz}(\theta_j)T_Z(d_j)T_X(a_j)T_{Rz}(\alpha_j)$$

which can be expanded as

$$^{j-1}A_j=\begin{bmatrix}\cos\theta_j & -\sin\theta_j\cos\alpha_j & \sin\theta_j\sin\alpha_j & a,\cos\theta_j\\ \sin\theta_j & \cos\theta_j\cos\alpha_j & -\cos\theta_j\sin\alpha_j & a,\sin\theta_j\\ 0 & \sin\alpha_j & \cos\alpha_j & d_j\\ 0 & 0 & 0 & 1\end{bmatrix}$$

- The parameters α_j and a_j are always constant.

$$q_j=\begin{cases}\theta_j & \sigma_j=0,\text{ for revolute point}\\ d_j & \sigma_j=1,\text{ for prismatic joint}\end{cases}$$



7.2 Forward Kinematics

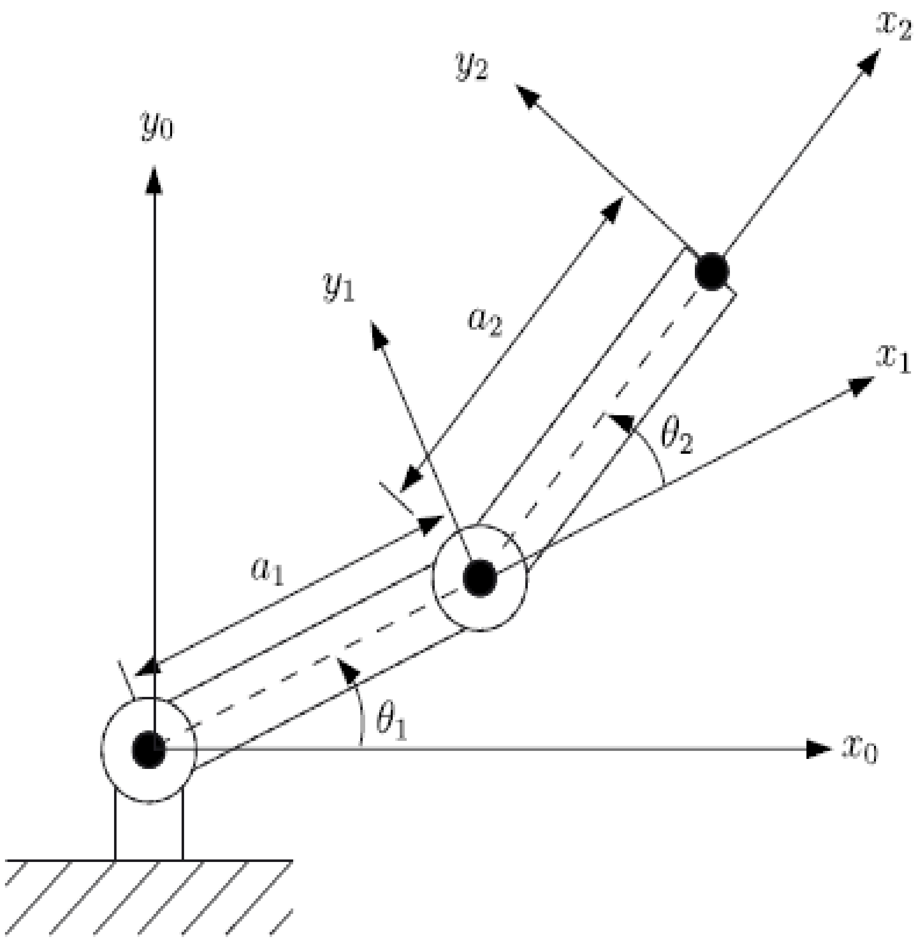
- The forward kinematics is often expressed in functional form

$$\xi_E = K(q)$$

- This is simply the product of the individual link transformation matrices which for an *N*–axis manipulator is

$$\xi_E \sim {}^0T_E = {}^0A_1 \, {}^0A_2 \ldots {}^{N-1}A_N$$

7.2.1 A 2-Link Robot



- Two-link robot structure

References

- [1] Robotics, Vision and Control. Peter Corke
- [2] Why Asimov's Three Laws Of Robotics Can't Protect Us
  Read more at https://www.gizmodo.com.au/2016/04/why-asimovs-three-laws-of-robotics-cant-protect-us/#ePkMlzk8SxTMUvEF.99