Lecture 2

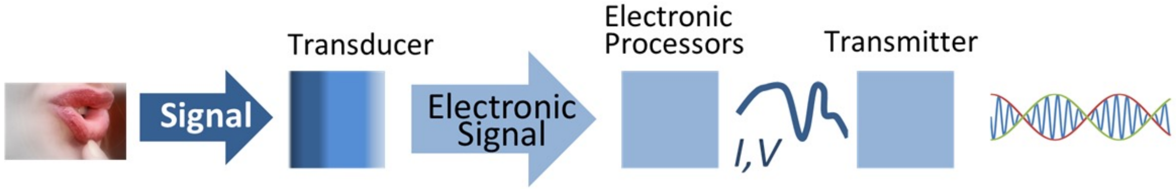
Actuators and Sensors

Matrices, Vectors and Cartesian Coordinate Systems, Frames, Mappings and Transformations

Alexandros Lioulemes, PhD

Transducer

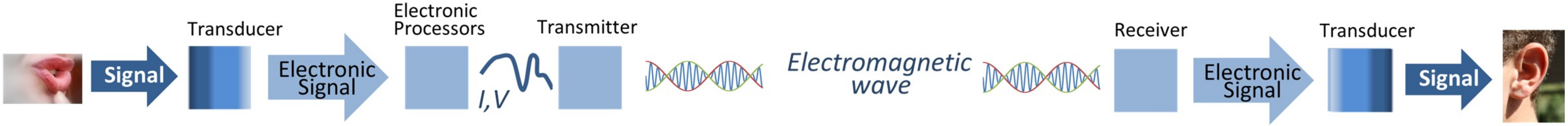
- A device that converts energy from one form to another.
- Types:
  - **Mechanical transducers** convert physical quantities into *mechanical* quantities or vice versa;
  - **Electrical transducers** convert physical quantities into *electrical* quantities or signals.
    - Thermocouple, Linear Variable Differential Transformer (LVDT)



https://en.wikipedia.org/wiki/Transducer

Transducer - Categories

- An **actuator** is responsible for moving or controlling a mechanical system;
- A **sensor** receives and responds to a signal from a physical system;
- A **bidirectional** transducer converts physical phenomena to electrical signals or vice versa;
  Antenna, Voice coils
- **Transceivers** integrate simultaneous bidirectional functionality.
  - Wireless, ultrasound transceivers



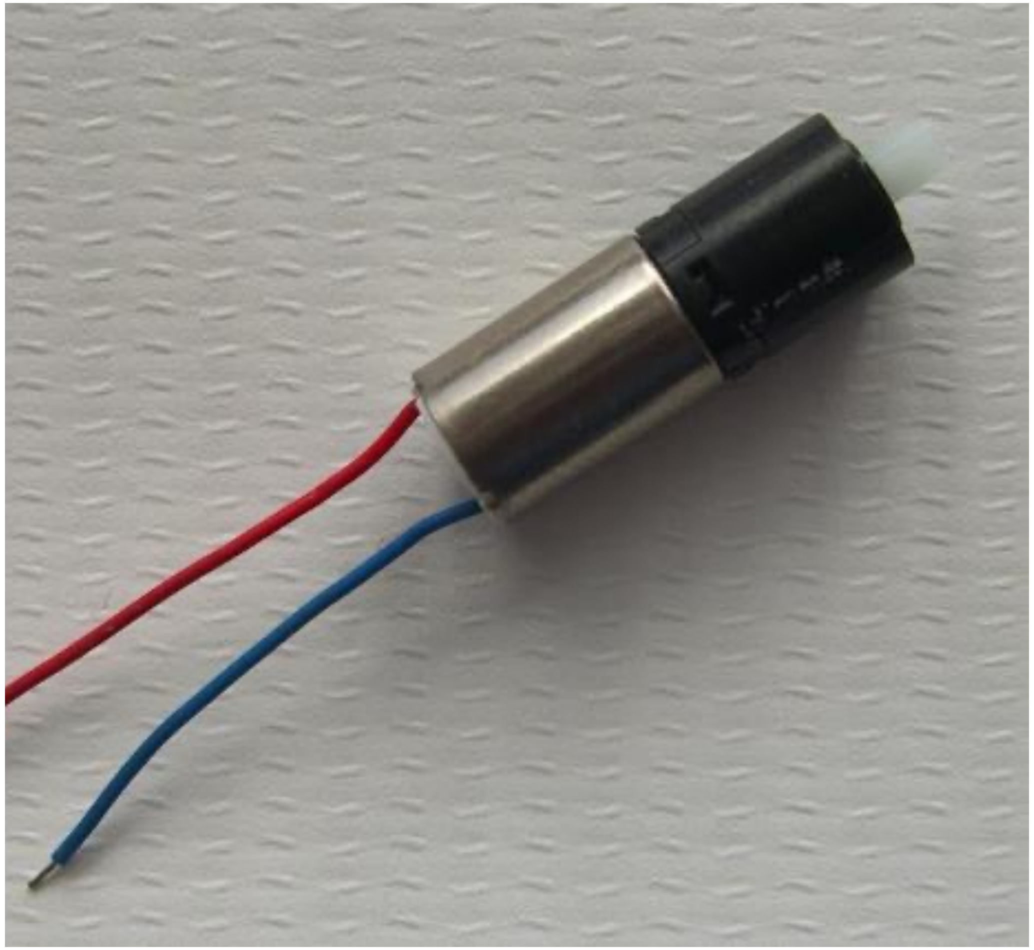
https://en.wikipedia.org/wiki/Transducer

Actuator

- An actuator is **a device that converts energy into physical motion**:
  - Linear actuators are devices that produce movement within a straight path → *force* (f)
  - Rotary actuators create a circular motion → *torque* (τ)
- Types:
  - DC motors
  - Servo Motors
  - Pneumatic actuators
  - Soft actuators
  - Hydraulic actuators
  - Piezoelectric actuators
  - Series Elastic actuators
  - Underactuation

https://en.wikipedia.org/wiki/Actuator

Actuator - Input



- **Current** is proportional to torque:

$$\tau_m = K_m i_a$$

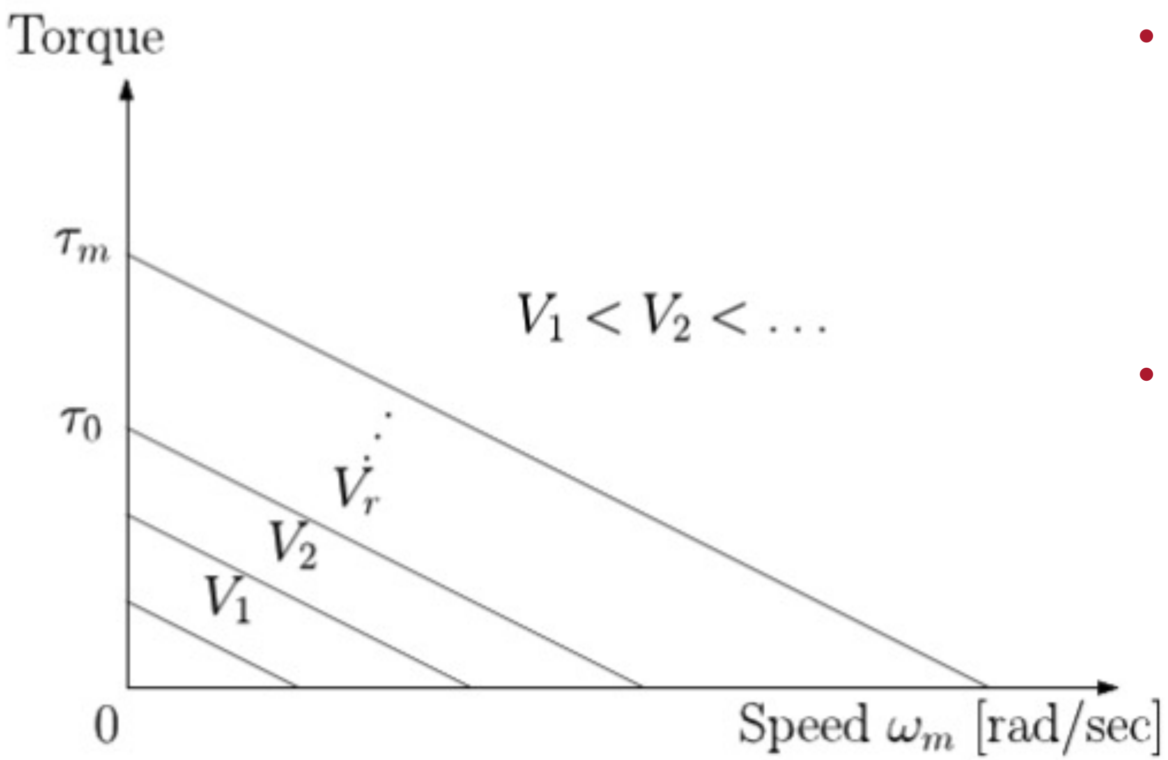
K_m is called the motor/torque constant, provided in actuator datasheet

- **Voltage** is proportional to speed:

$$V_b = K_b \omega_m$$

K_b is the back-electromotive-force (EMF) constant, again available in actuator datasheet

Actuator - Torque vs. Speed



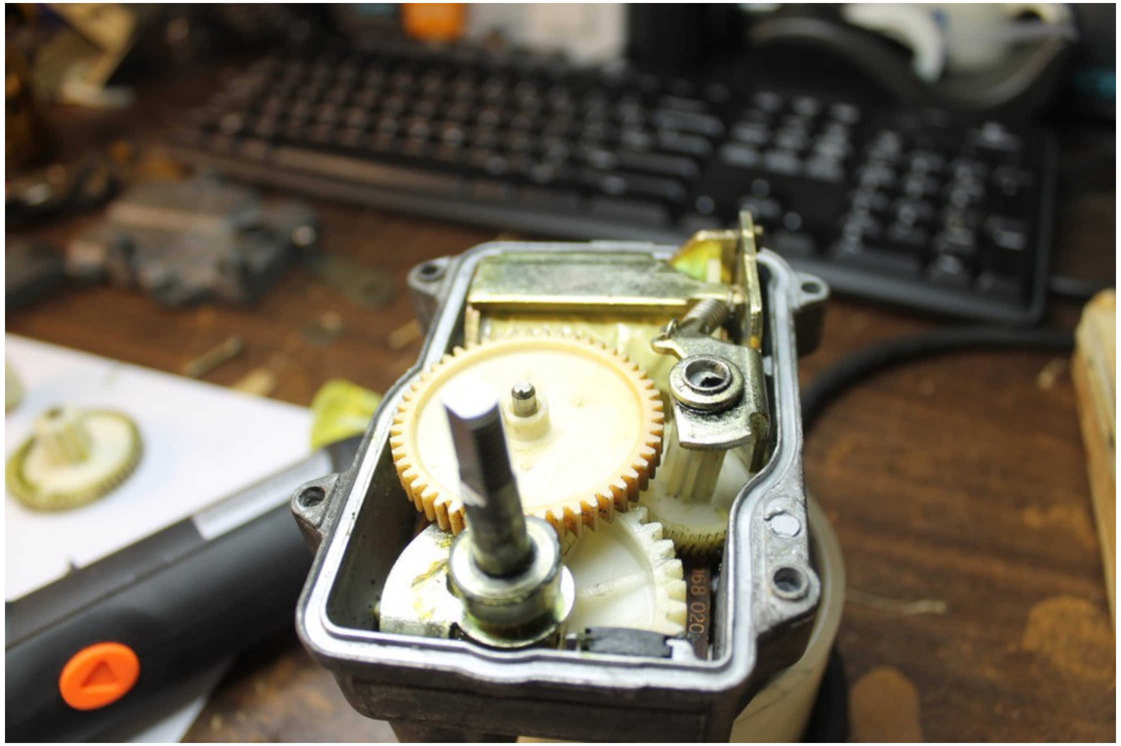
- **Current** is proportional to torque:

$$\tau_\mathrm{m} = K_\mathrm{m} i_\mathrm{a}$$

- **Voltage** is proportional to speed:

$$V_\mathrm{b} = K_\mathrm{b} \omega_\mathrm{m}$$

Actuator - Torque



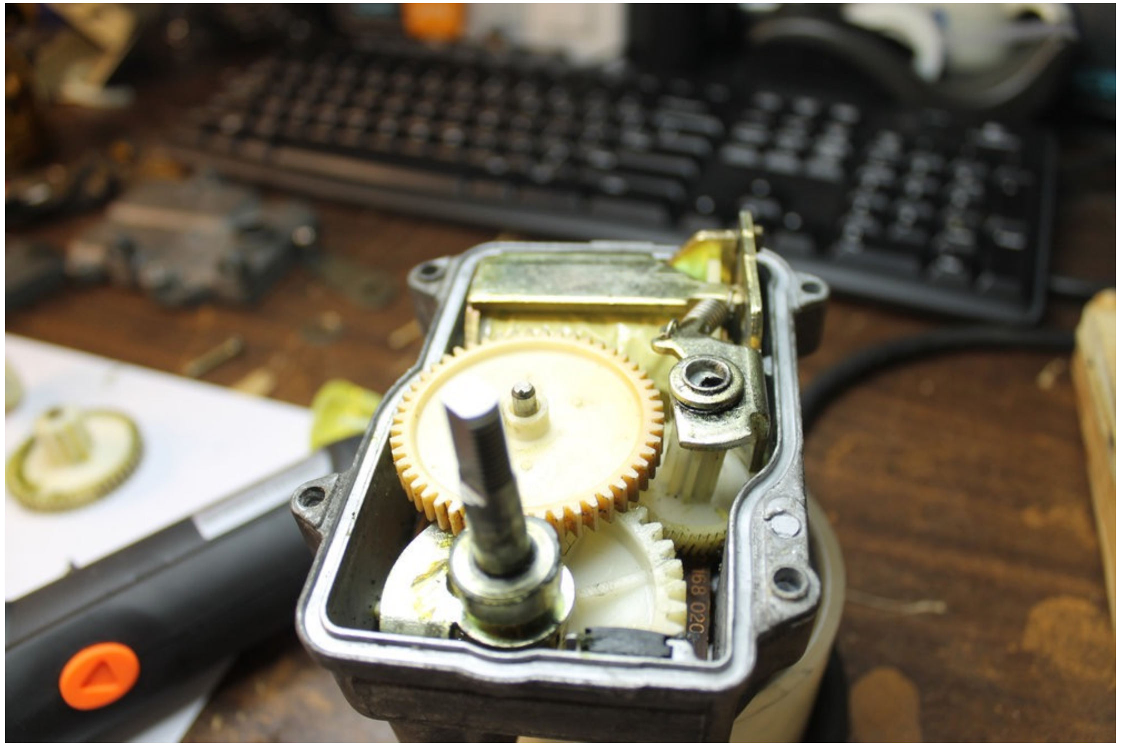
$$\tau_l = \tau_m r$$

τ_l: torque at the end of the gear

τ_m: torque of the actuator

r: gear ratio

Actuator - Velocity



$$\dot{\theta}_{\mathrm{s}} = \frac{\dot{\theta}_m}{r}$$

$\dot{\theta}_{\mathrm{m}}$: velocity of the actuator

$\dot{\theta}_{\mathrm{s}}$: velocity at the end of the gear

r: gear ratio

Actuator - Inertia

J_a: Actuator inertia

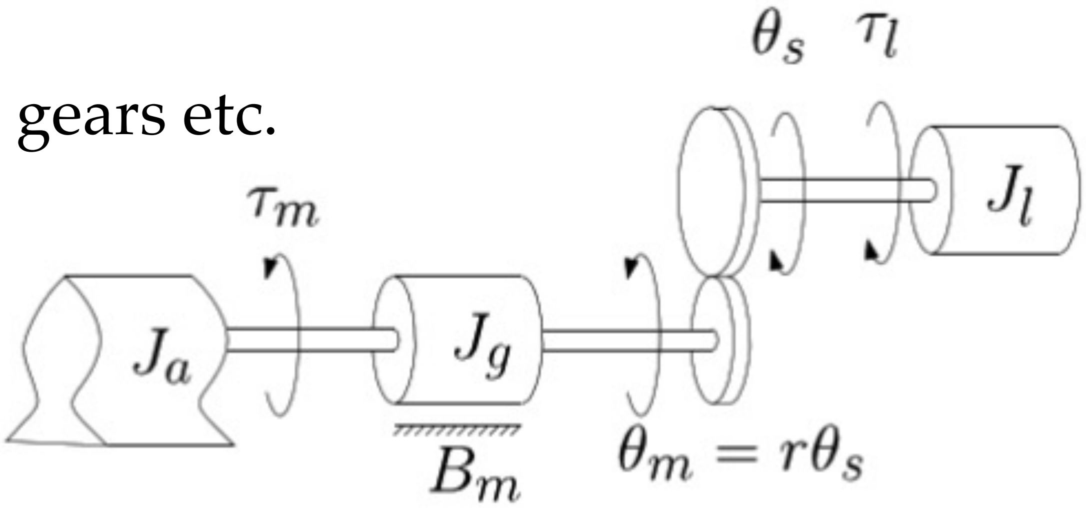
τ_m: torque of the actuator

Bm : Frictional effects related to the actuator, gears etc.

J_g: Gear inertia

τ_l: torque at the end of the gear

J_l: Link inertia

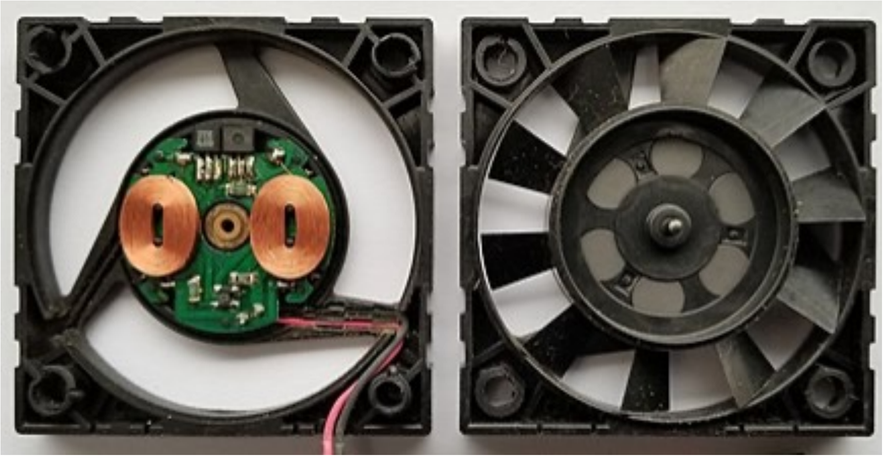


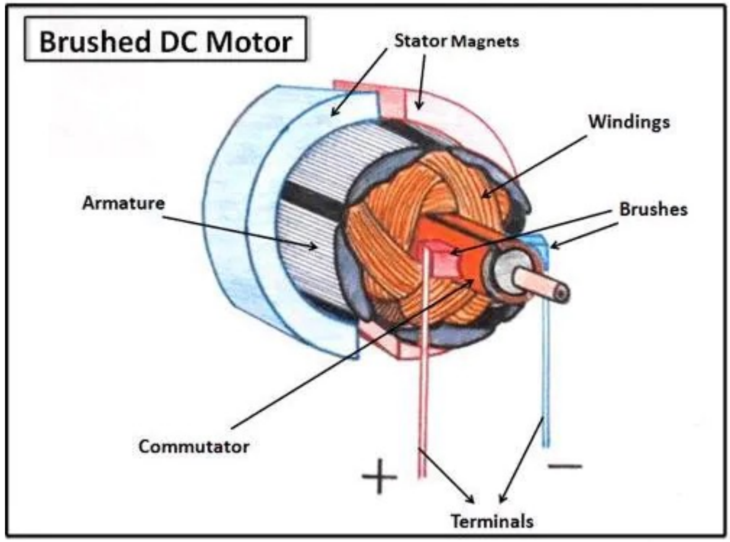
$$J\ddot{\theta} + B\dot{\theta} = U$$

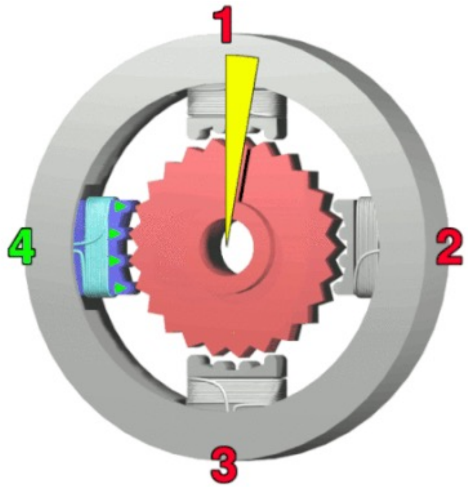
$$\boldsymbol{Cumulative\ inertia + Cumulative\ drag\ forces = Input\ (J_a)}$$

Actuator - Type - DC Motors

- Brushed motors
  - Cheaper
- Brushless motors
  - More expensive
  - Lower maintenance
  - Efficient
- Stepper motors
  - High holding torque but not as fast
  - Less energy efficient







https://en.wikipedia.org/wiki/Brushed_DC_electric_motor
https://en.wikipedia.org/wiki/Brushless_DC_electric_motor
https://en.wikipedia.org/wiki/Stepper_motor

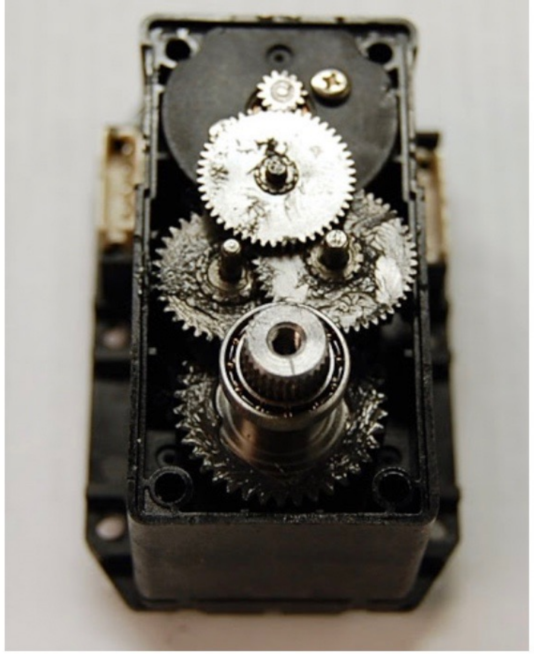
Actuator - Type - DC Motors - Controllers/Drivers

- Generates the required voltage pattern to operate the actuators
  - Current-Torque Control Mode
  - Voltage-Speed Control Mode

https://en.wikipedia.org/wiki/Motor_controller

Actuator - Type - Servo Motors

- They have their own embedded controller and gears
- Dynamixels are good examples

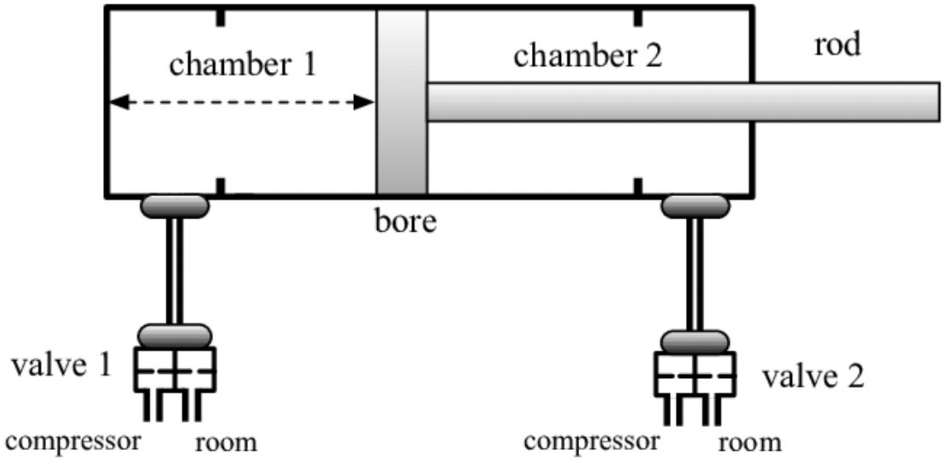


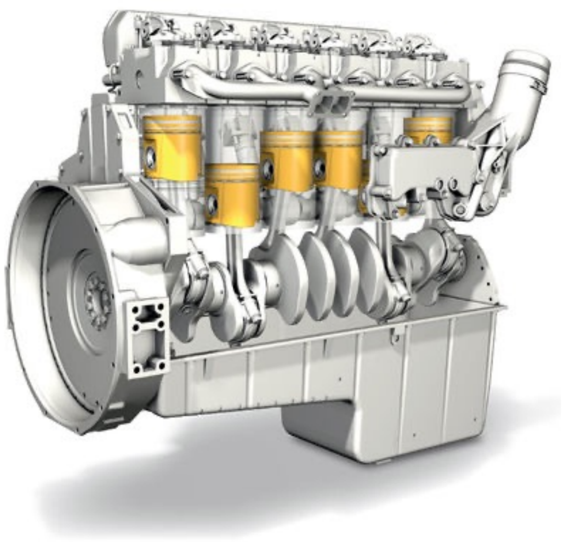
Industrial servomotors and gearboxes, with standardized flange mountings for interchangeability

https://en.wikipedia.org/wiki/Servomotor

Actuator - Type - Pneumatic Actuators

- A device that converts compressed air into a linear or rotary mechanical motion.
- Consists of pistons, cylinders, valves (ports) and compressors.
- Reliable and safe
- Applications:
  - Combustible automobile engines
  - Air Compressors
  - Packaging and production machinery
  - Railway
  - Aviation

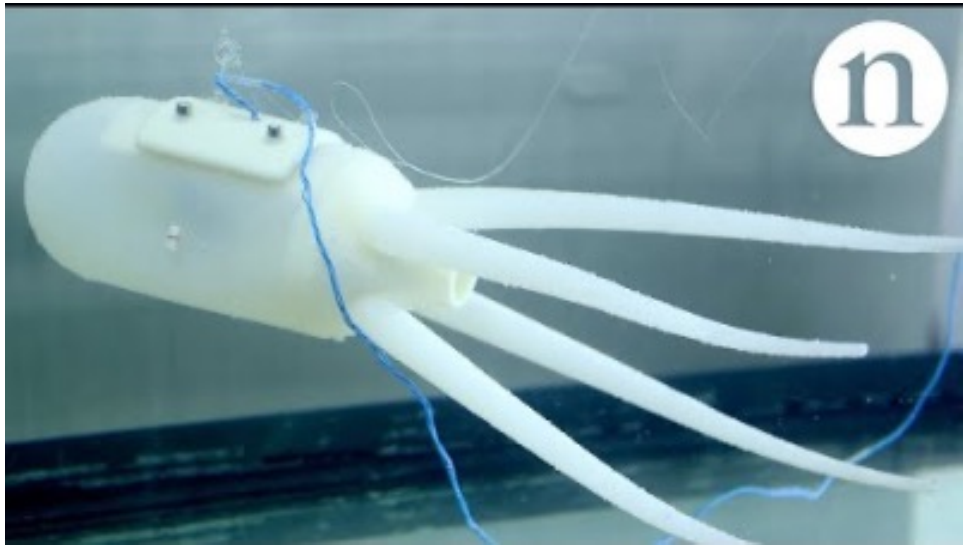


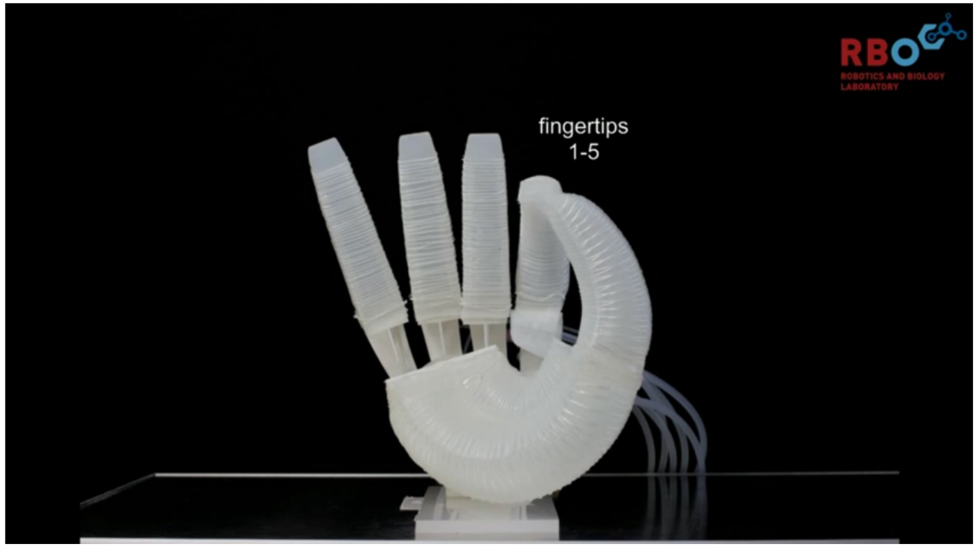


https://en.wikipedia.org/wiki/Pneumatic_actuator

Actuator - Type - Soft Actuator

- Change their shape in response to stimuli
- Applications to safety and healthcare for humans



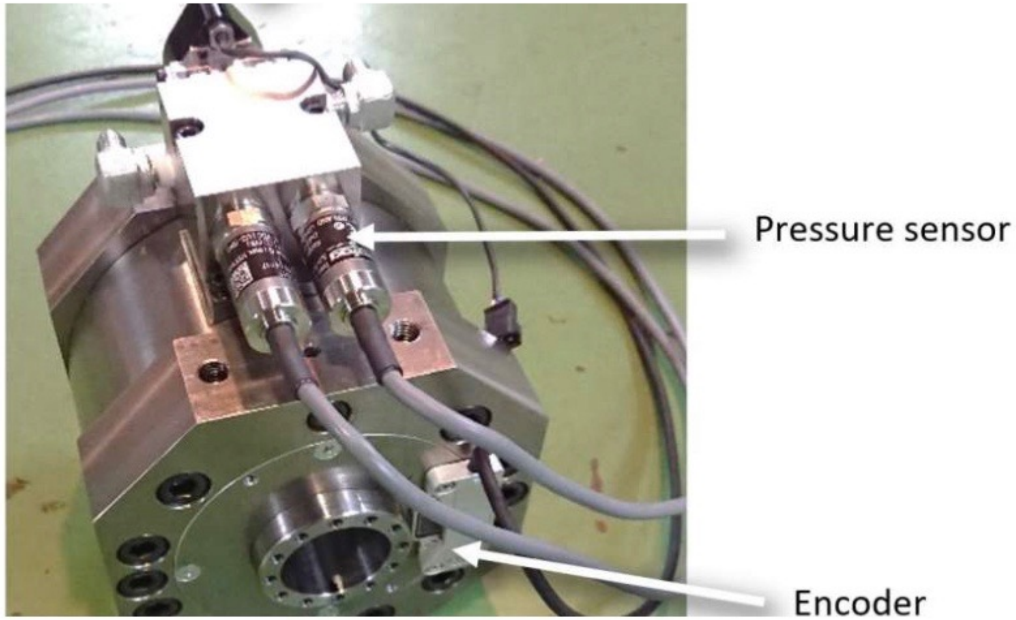


https://www.youtube.com/watch?v=A7AFsk40NGE

https://en.wikipedia.org/wiki/Soft_robotics

Actuator - Type - Hydraulic actuators

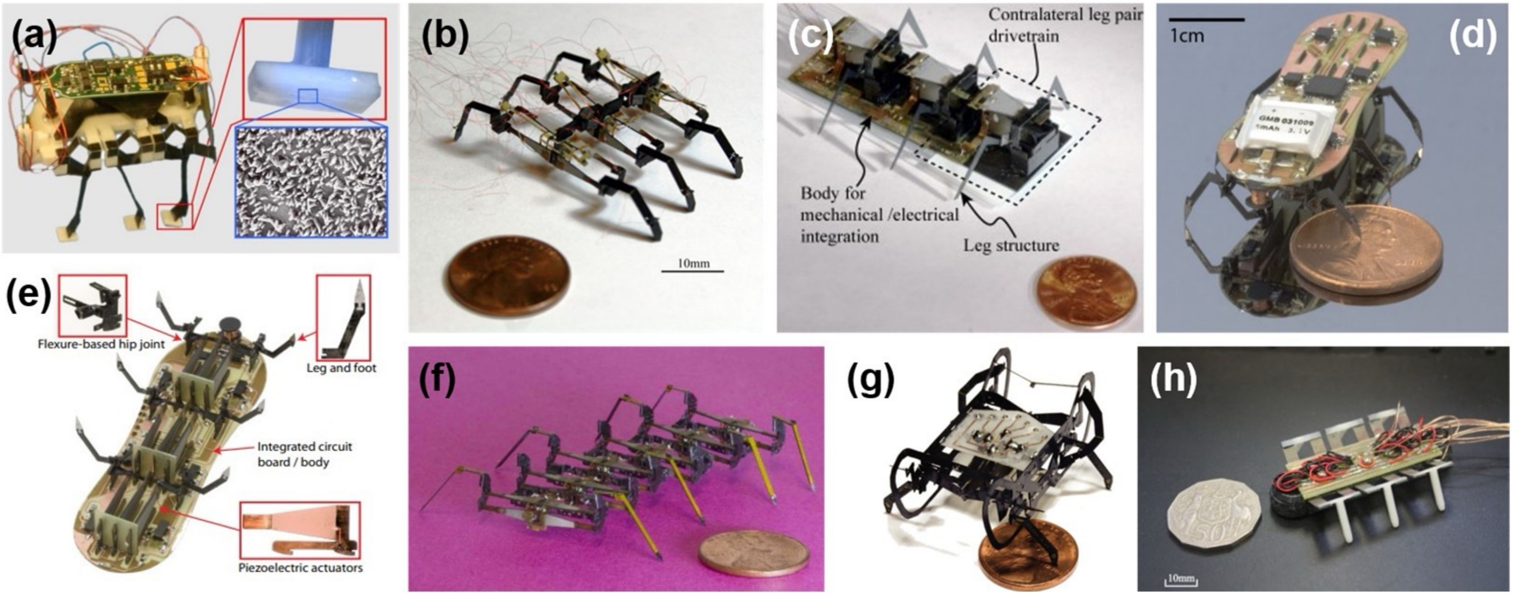
- Very strong
- Requires a pump system
- Not very common in robotics (except - Boston Dynamics) - Atlas →



https://en.wikipedia.org/wiki/Hydraulic_motor

Actuator - Type - Piezoelectric actuators

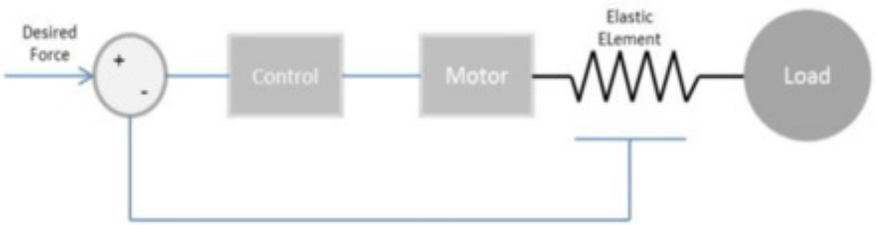
- Transducers that convert electrical energy into a mechanical displacement based on a piezoelectric effect or vice versa;
- Ceramics that expands and shrink depending on the applied current;
- Very popular in micro robotics;
  - Microelectromechanical systems (MEMS)



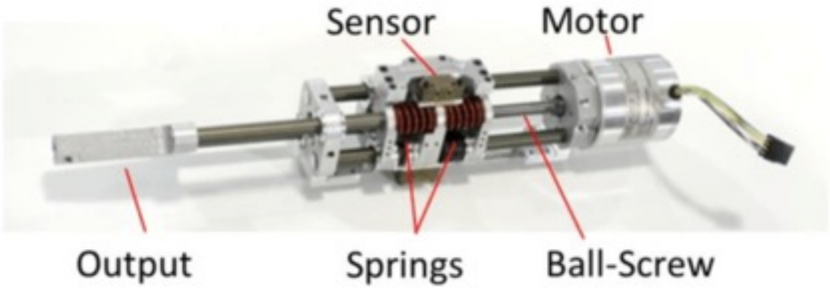
https://en.wikipedia.org/wiki/Piezoelectric_motor

Actuator - Type - Series Elastic Actuators (SEA)

- Rely on elastics between the motor actuator and the load;
- Safer for human-robot-interaction applications;
- Energy efficient and shock absorption;
- Reduce excessive wear on the transmission;



(a)



https://www.youtube.com/watch?v=gZLO2Am0Zk8

Actuator - Type - Underactuation

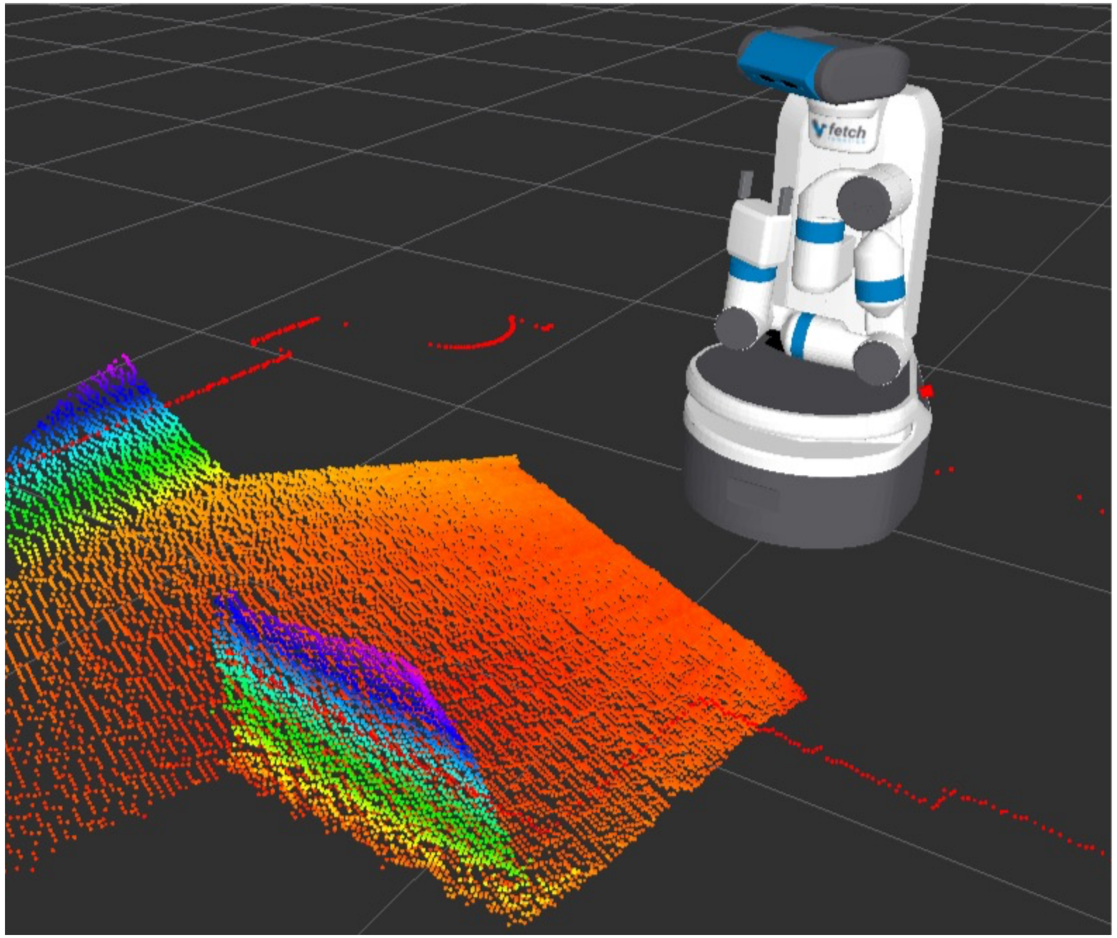
- There are less independent action variables than the degrees of freedom of the mechanical system.
- The number of actuators you have is less than the number of joints.

https://www.youtube.com/watch?v=C340gbK3sZc

https://en.wikipedia.org/wiki/Underactuation

Sensors

- Robots need to acquire information about:
  - Themselves:
    - Joint positions
    - Joint velocities
    - Joint toques
  - Their environment:
    - Locations of the objects
    - Collisions avoidance



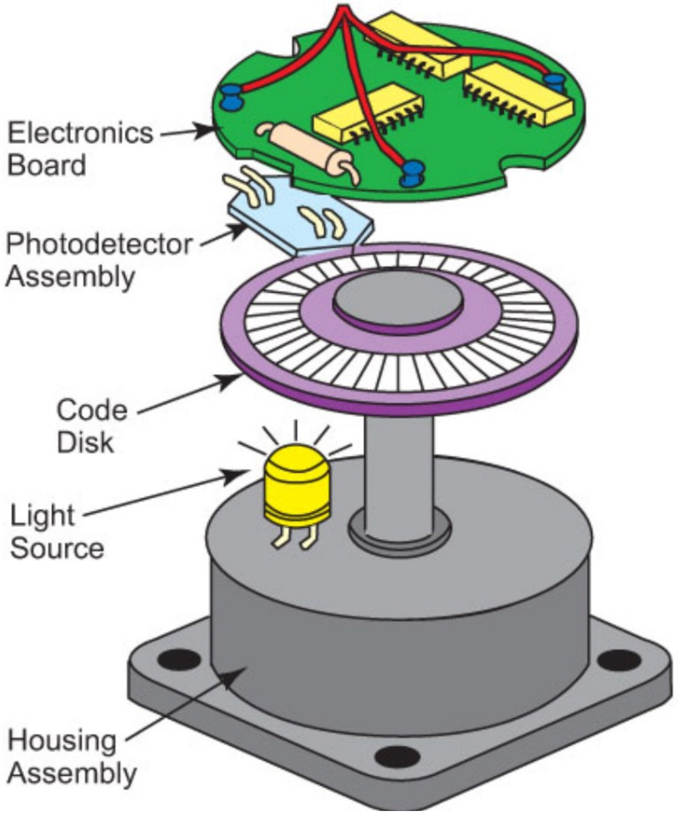
https://en.wikipedia.org/wiki/Sensor

Sensors - Types

- Proprioceptors:
  - Encoders
  - Joint torque
  - Accelerometers
  - Gyroscopes
  - …
- Exteroceptors:
  - Cameras
  - LiDAR
  - Infra-red
  - Sonar
  - …

Sensors - Encoder

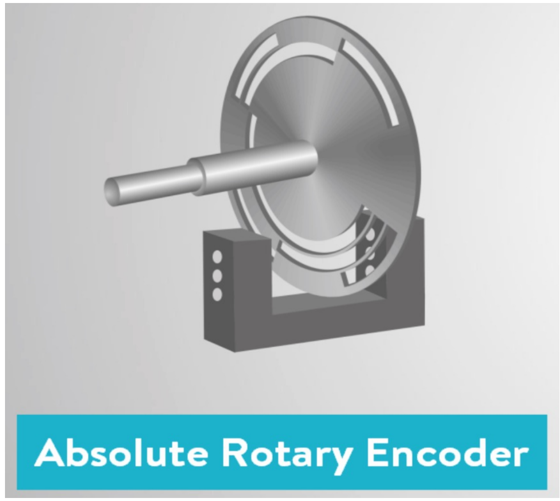
- Often used to measure the position of the joints
- The rotor is coupled with a disk
- The disk rotates together with the rotor generating square waves on a photodetector

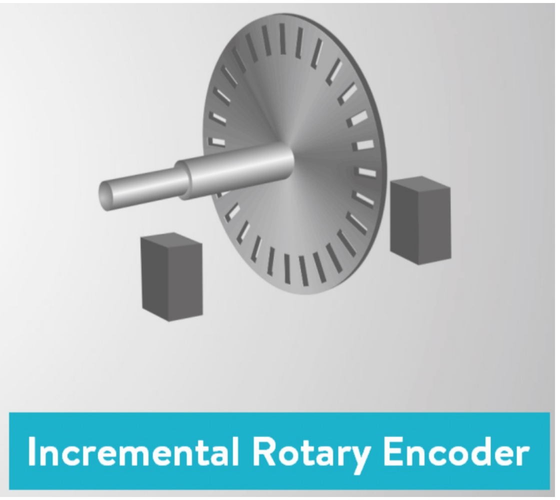


https://en.wikipedia.org/wiki/Rotary_encoder

Sensors - Encoder - Types

- Incremental
  - Assumes its start location is zero when powered on
  - Requires calibration for absolute measurements
- Absolute
  - Does not forget the zero location

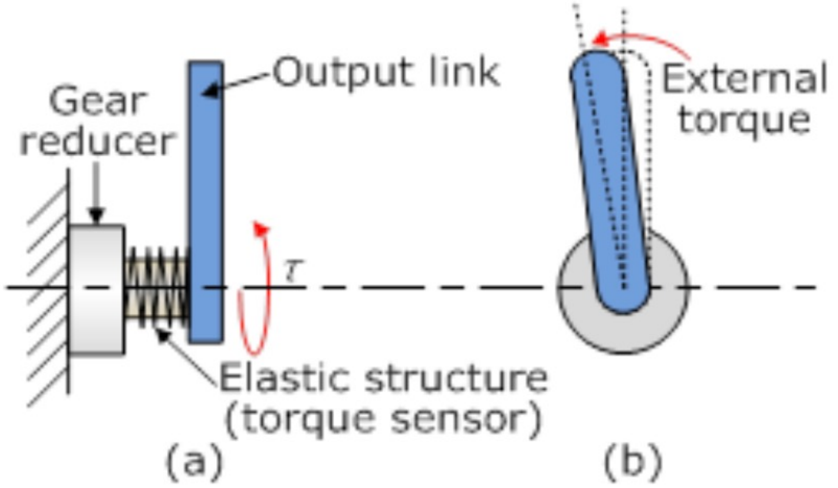




https://en.wikipedia.org/wiki/Rotary_encoder

Sensors - Joint torque

- Need for accurate force sensing of the external load at the end-effector of manipulators
- Rely on strain gauges
- They deform linearly as external forces are applied up to the elastic limits (Hooke's law)
- The structure is only sensitive to the torque applied with respect to the rotation axis.

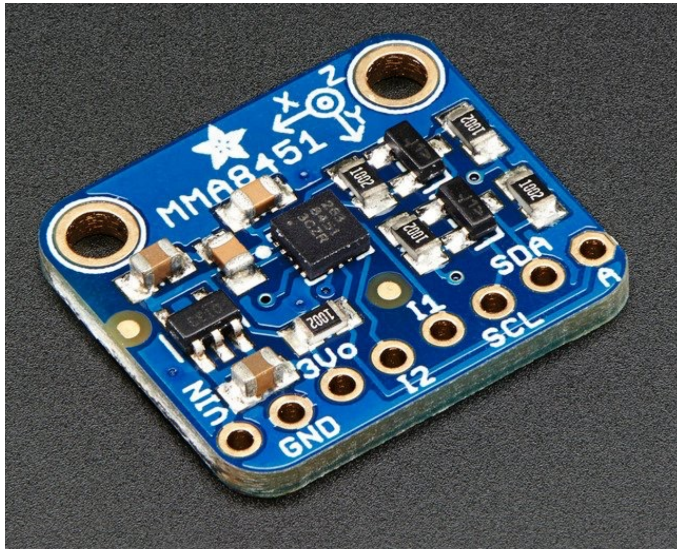


Deformation of elastic structure by external torque
(a) side view
(b) top view

https://en.wikipedia.org/wiki/Torque_tester

Sensors - Accelerometer

- Measures the acceleration of any body or object
  - Proper acceleration - instantaneous rest frame
  - Coordinate acceleration – fixed coordinate system
- Structure: is a damped mass, a proof mass, on a spring



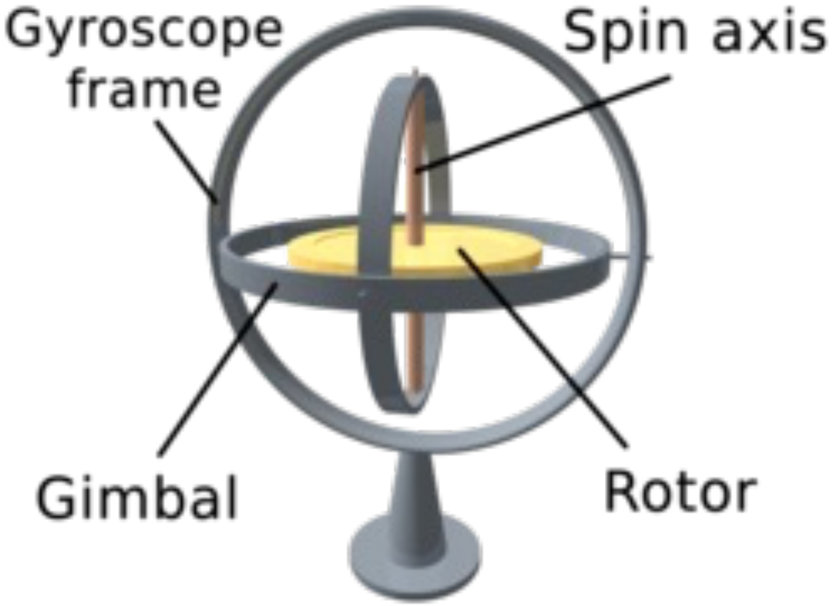
Adafruit Triple-Axis Accelerometer - ±2/4/8g @ 14-bit - MMA8451

https://www.youtube.com/watch?v=i2U49usFo10&t=1s

https://en.wikipedia.org/wiki/Accelerometer

Sensors - Gyroscope

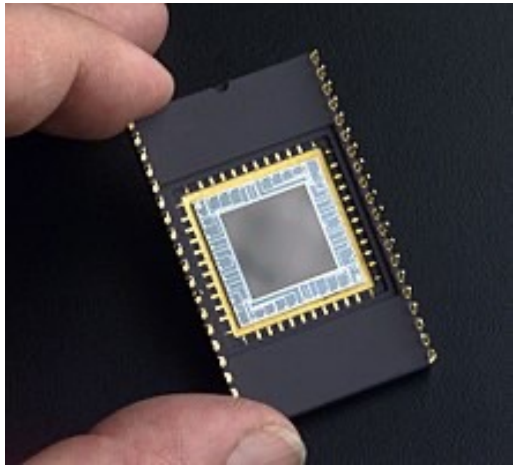
- Gyroscopes (or gyros) measure or maintain rotational motion.
- Spinning wheel which the axis of rotation is free and unaffected by tilting or rotation of the mounting.
- Wheel mounted into three gimbals providing pivoted supports for allowing the wheel to rotate about a single axis.



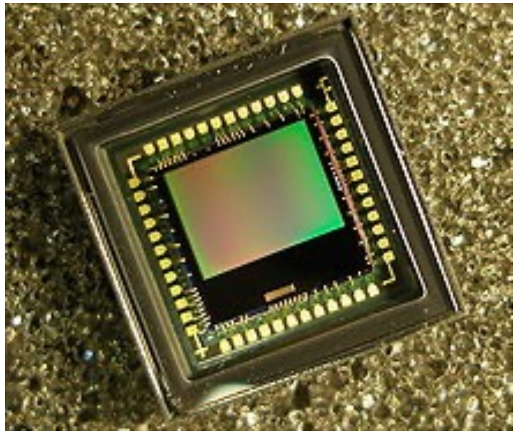
https://en.wikipedia.org/wiki/Gyroscope

Sensors - Camera

- Optical instrument that captures images.
  - 2D images
  - 3D images
- Image sensor detects and conveys information used to make an image.
  - The waves can be light or other electromagnetic radiation
- Types:
  - the charge-coupled device (CCD)
  - the active-pixel sensor (CMOS)



CCD

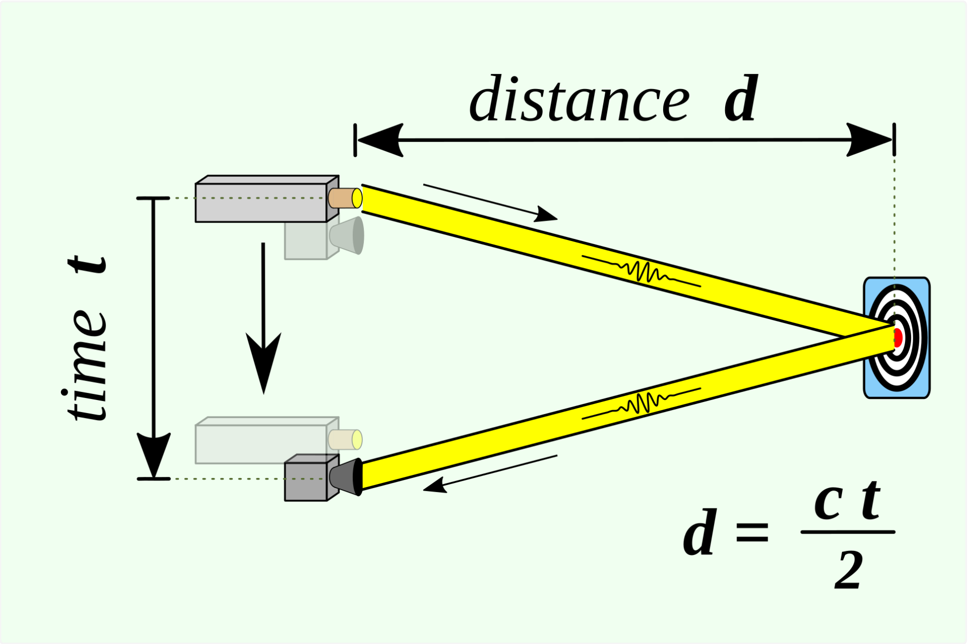


CMOS

https://en.wikipedia.org/wiki/Camera

Sensors - LiDAR

- Determines ranges by targeting an object or a surface with a laser and measuring the time for the reflected light to return to the receiver.
- Ultraviolet, visible, or near infrared light to image objects.

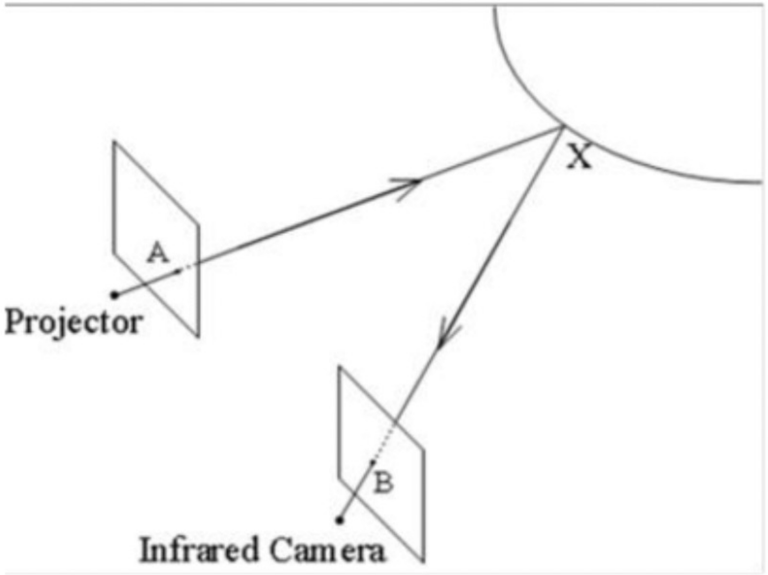


c: speed of light
d: distance between the detector and the object
t: time spent for the laser light to travel to the object, then travel back to the detector

https://en.wikipedia.org/wiki/Lidar

Sensors - Infra-red (IR)

- Electromagnetic radiation (EMR) with wavelengths longer than those of visible light (invisible to the human eye)
- An infrared projector projects a random pattern onto the scene observed by an infrared camera




https://en.wikipedia.org/wiki/Infrared

Sensors - SONAR

- SOund Navigation And Ranging (SONAR)
  - technique that uses sound propagation to navigate, measure distances (ranging), communicate with or detect objects on or under the surface of the water, such as other vessels.
- SONAR transducers emit sonic pings which bounce off of elements in the environment.
- The time it takes for the ping to return to the transducer is scaled by the speed of sound in the robot's operating medium (i.e., air or water) to determine the distance to the object.
- Types:
  - *passive* sonar is essentially listening for the sound made by vessels;
  - *active* sonar is emitting pulses of sounds and listening for echoes.

https://en.wikipedia.org/wiki/Sonar

Sensors - Calibration

- Adjustments performed on a sensor to make it as accurately, or error free, as possible.
- Reasons for sensor errors:
  - Improper zero reference;
  - Shift in sensor's range;
  - Mechanical Wear or Damage;
- Calibration prompts:
  - A new sensor;
  - After the sensor has been repaired or modified;
  - Moving from one location to another;
  - or when a specified time period has elapsed;
  - specified by the requirements (manufacturer recommendation)

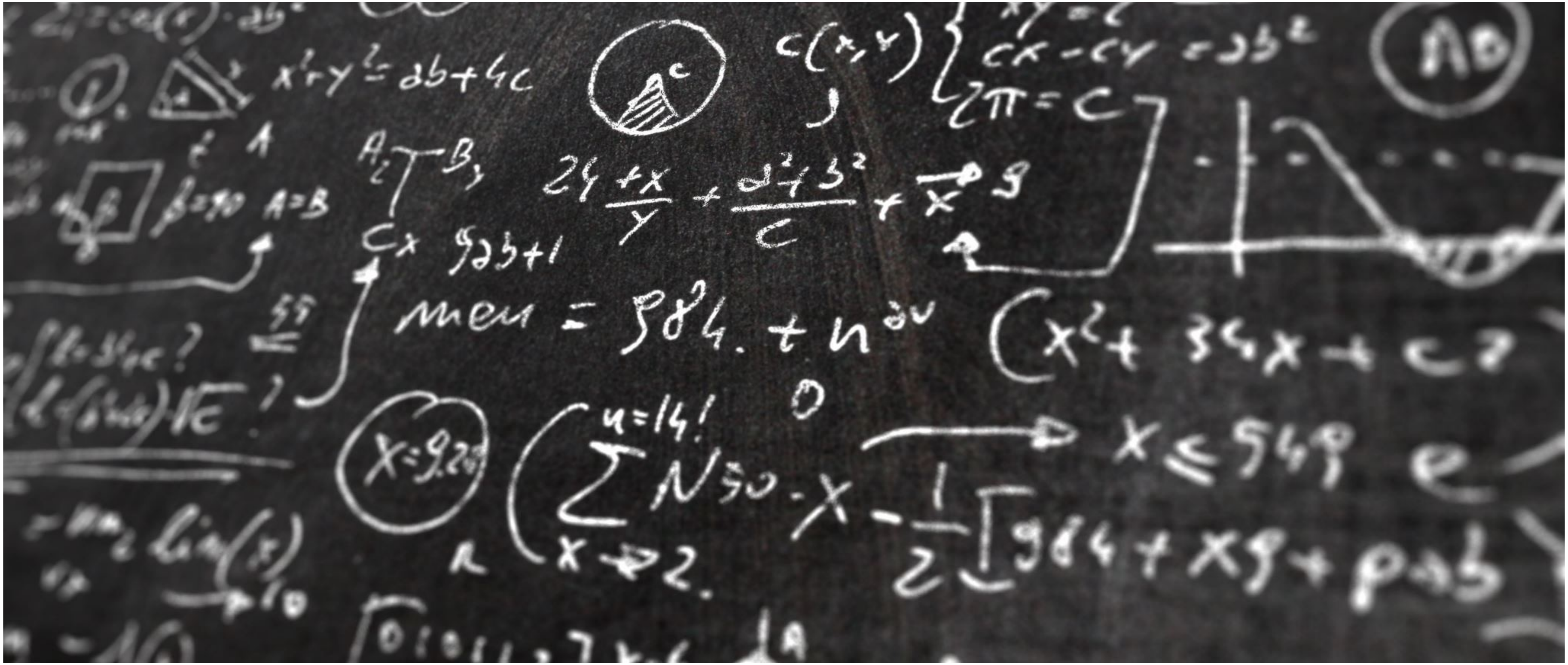
https://en.wikipedia.org/wiki/Calibration

Sensors - Calibration - Home position

- Setting Home position in a robotic arm is an example of sensor calibration

https://support.barrett.com/wiki/WAM/HardwareSetup#WAMStandardHomePosition

Math Reminder: Linear Algebra



https://en.wikipedia.org/wiki/Linear_algebra

Installation of MATLAB

MATLAB®

https://hub.wpi.edu/article/805/matlab-install-guide

Matrix

- Matrix:

$$\mathbf{R} = \left[r_{ij} \right]_{\substack{i=1,\ldots,m \\ j=1,\ldots,n}} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix}_{mxn}$$

- Transpose of a matrix:

$$R^{\mathrm{T}} = \begin{bmatrix} r_{11} & \cdots & r_{m1} \\ \vdots & \ddots & \vdots \\ r_{1n} & \cdots & r_{mn} \end{bmatrix}_{n \times m}$$

https://en.wikipedia.org/wiki/Linear_algebra#Matrices

Matrix

- Square matrix:
  - Numbers of rows = number of columns

$$A \equiv \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

- Inverse of the square matrix:
  - Inverse matrix B such that AB = BA

$$AB = BA = I_n$$

  - If *B* exists, *B* is called inverse of *A*

$$A^{-1}$$

- *A* matrix is orthogonal when:

$$A^\top = A^{-1}$$

- Identity matrix

$$I_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Matrix - Example

- Calculate the transpose matrix of A {or A^T }

$$A = \begin{bmatrix} 1 & 4 & 6 & 5 & 8 \\ 2 & 3 & 7 & 9 & 0 \end{bmatrix}$$

- Calculate the Inverse matrix of B {or B^{-1}}

$$B = \begin{bmatrix} 3 & 0 & 2 \\ 2 & 0 & -2 \\ 0 & 1 & 1 \end{bmatrix}$$

Matrix - Example - Solution in MATLAB

```
>> A = [1 4 6 5 8; 2 3 7 9  0]

A =

     1     4     6     5     8
     2     3     7     9     0
```

```
>> A_Tr = transpose(A)

A_Tr =

     1     2
     4     3
     6     7
     5     9
     8     0
```

```
>> B = [3, 0, 2; 2, 0, -2; 0, 1, 1]

B =

     3     0     2
     2     0    -2
     0     1     1
```

```
>> B_Inv = inv(B)

B_Inv =

    0.2000    0.2000         0
   -0.2000    0.3000    1.0000
    0.2000   -0.3000         0
```

```
>> I = B * B_Inv

I =

    1.0000         0         0
   -0.0000    1.0000         0
         0         0    1.0000
```

Vector

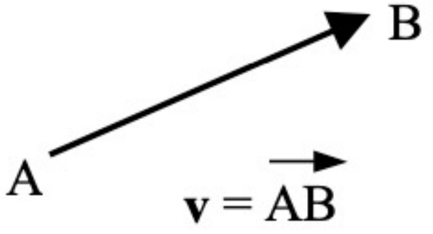
- Vector: a column vector in R^n is a nx1 matrix

a vector in R^3:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

vector *magnitude:*

$$|\mathbf{v}| = \sqrt{v_1^2 + v_2^2 + v_3^2}$$



<u>Magnitude</u>

<u>Direction</u>

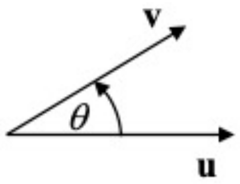
https://en.wikipedia.org/wiki/Vector_(mathematics_and_physics)

Vector - Operations

- Dot (Scalar) Product: $\mathbf{v} \cdot \mathbf{u} = v_1 u_1 + v_2 u_2 + v_3 u_3$

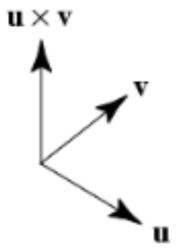
  or

  $\mathbf{v} \cdot \mathbf{u} = |\mathbf{v}| \cdot |\mathbf{u}| \cdot \cos\theta$



- **v, u**: non-zero vectors. When are they orthogonal?

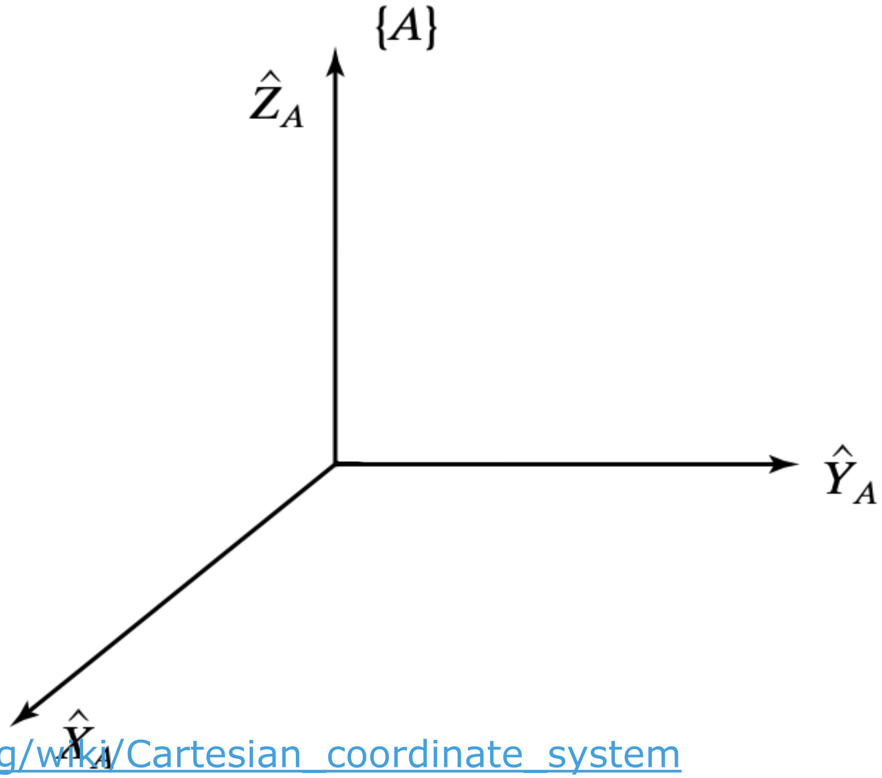
- Cross Product: $\mathbf{u} \times \mathbf{v} = |\mathbf{u}| \cdot |\mathbf{v}| \cdot \sin\theta$



https://www.mathworks.com/help/matlab/ref/dot.html
https://www.mathworks.com/help/matlab/ref/cross.html

Cartesian Coordinate System

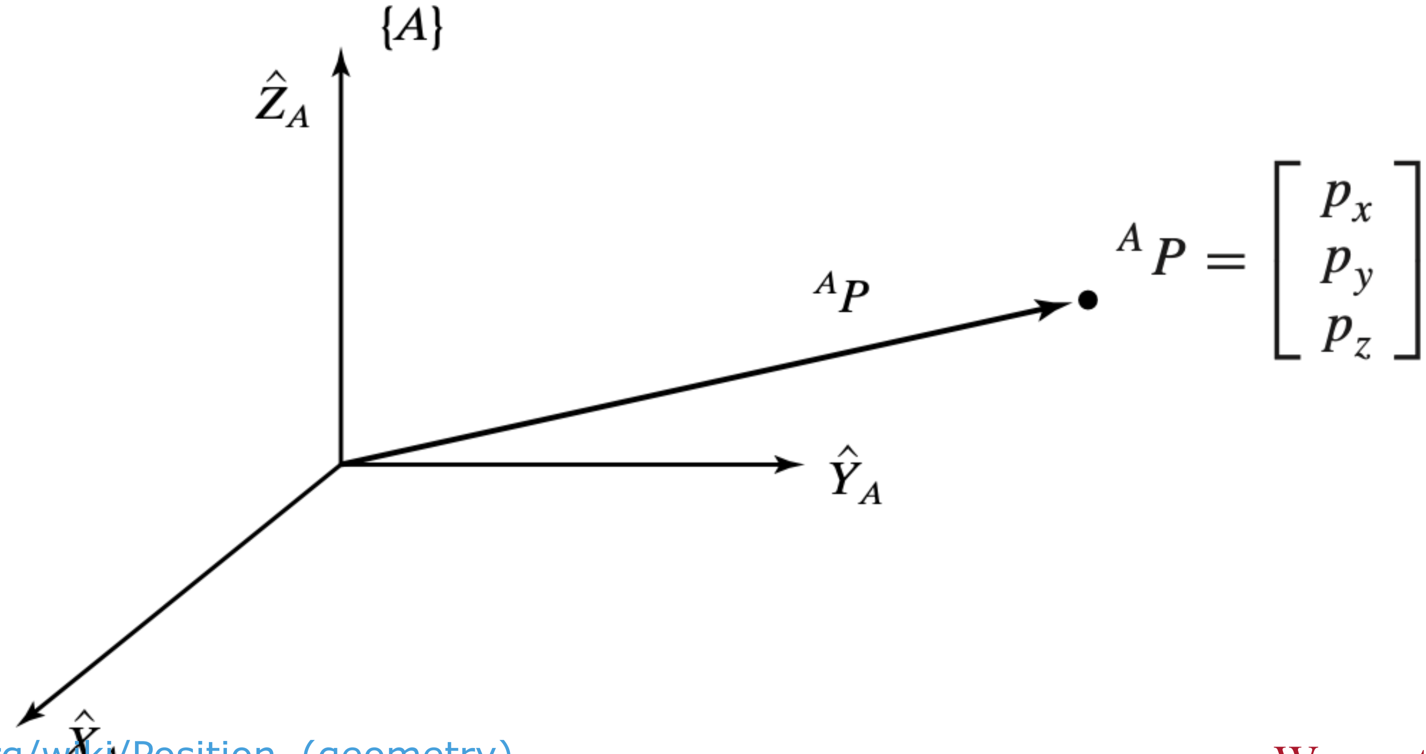
- Specifies each point uniquely by a pair of numerical coordinates
- Our philosophy: There is a universe coordinate system to which everything we discuss can be referenced.



https://en.wikipedia.org/wiki/Cartesian_coordinate_system

Description of a Position

- Locate any point in the universe with a *3 x 1* position vector
- Convention: vectors are written with a leading superscript indicating the coordinate system to which they are references (${}^{A}P$)



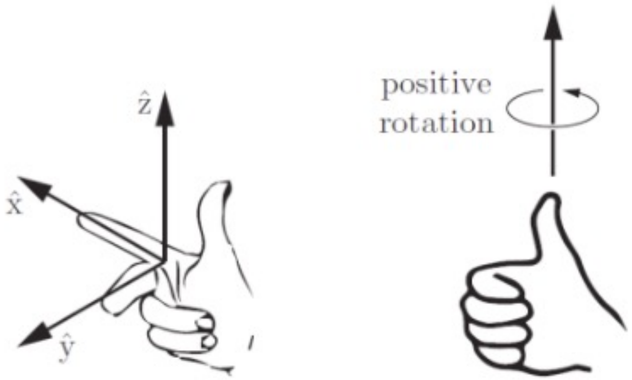
https://en.wikipedia.org/wiki/Position_(geometry)

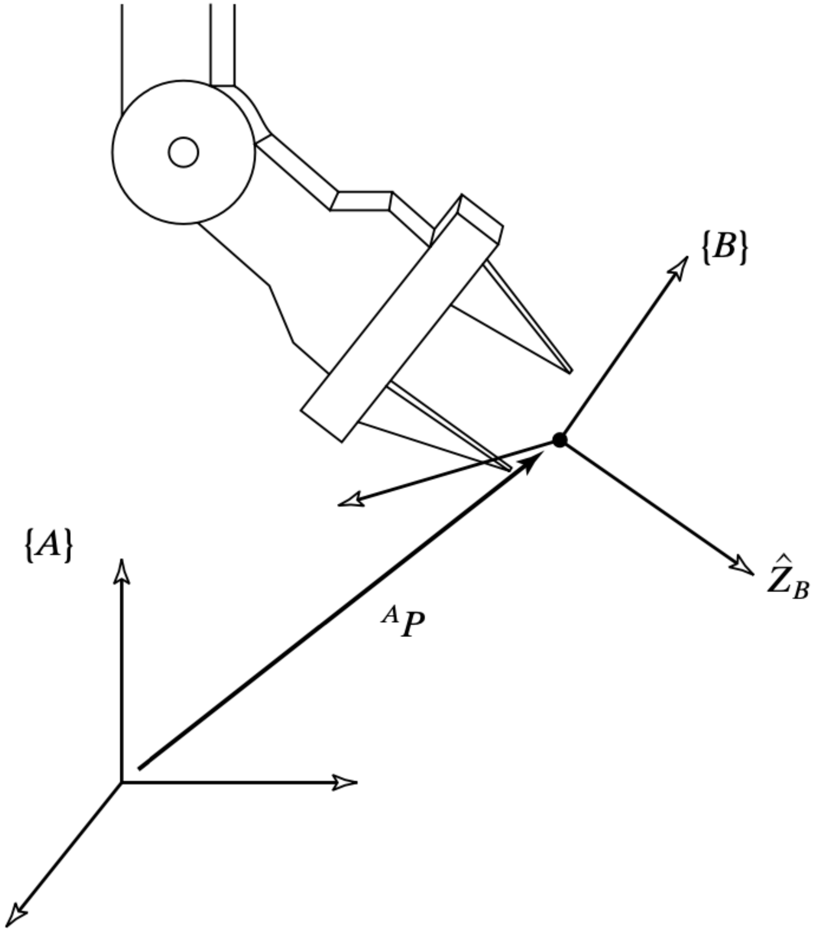
Description of an Orientation

- Rotation Matrix:

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

- {B} relative to {A}





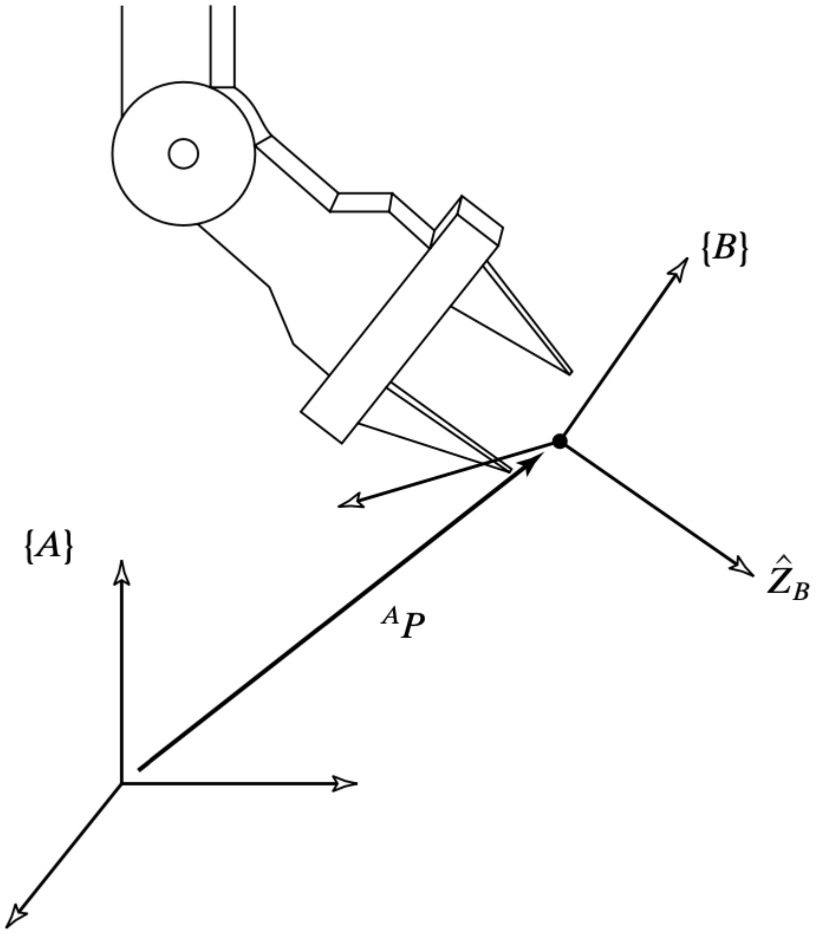
https://en.wikipedia.org/wiki/Orientation_(geometry)

Description of an Orientation

- The components of any vector are the projections of that vector onto the unit directions of its reference coordinate system.

$$
{}^{A}_{B}R = \left[\ {}^{A}\hat{X}_B \quad {}^{A}\hat{Y}_B \quad {}^{A}\hat{Z}_B\ \right] = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix}
$$

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



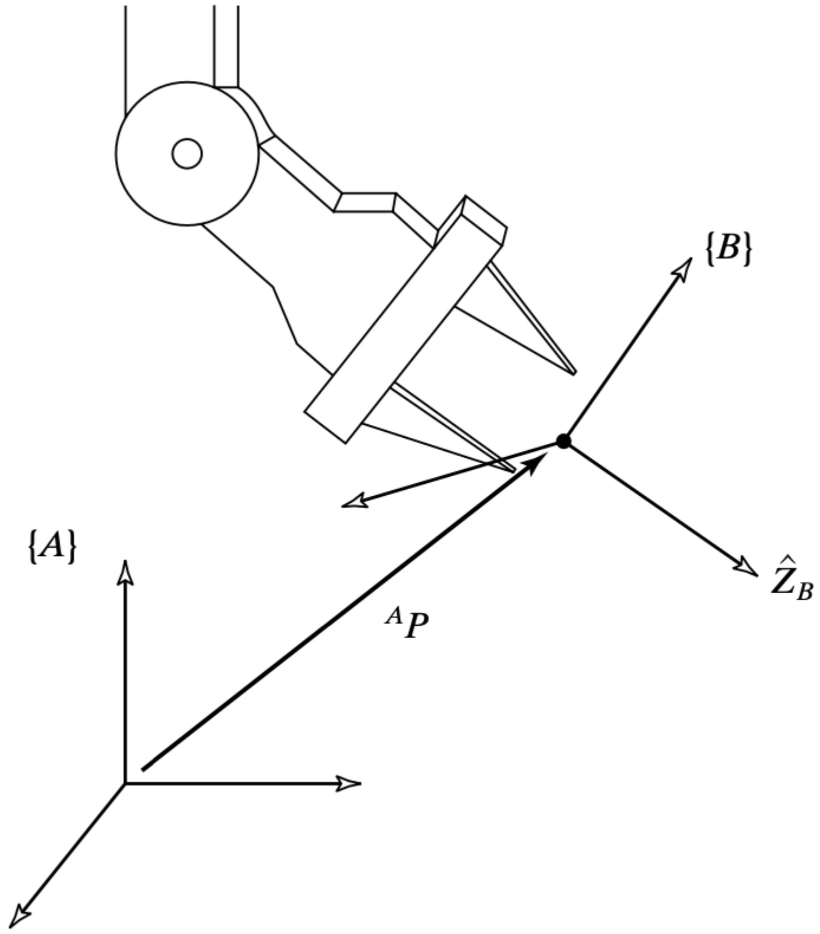
https://en.wikipedia.org/wiki/Orientation_(geometry)

Description of a Frame

- Frame: a set of four vectors giving position and orientation information.

- Frame $\{B\} = \{^A_B R, ^A P_{BORG}\}$

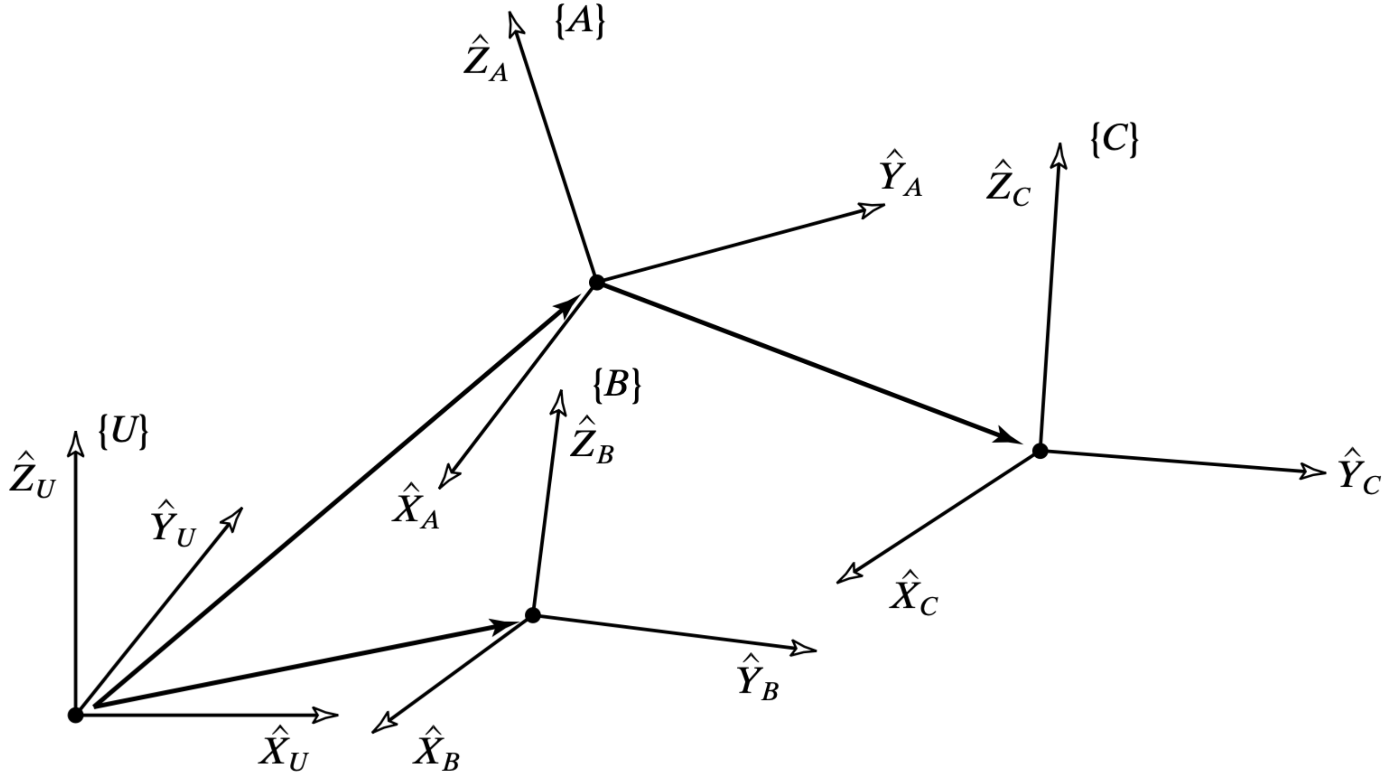
  where $^A_B R$ rotation matric of {B} relative to {A}, and $^A P_{BORG}$ the vector that locates the origin of the frame {B}



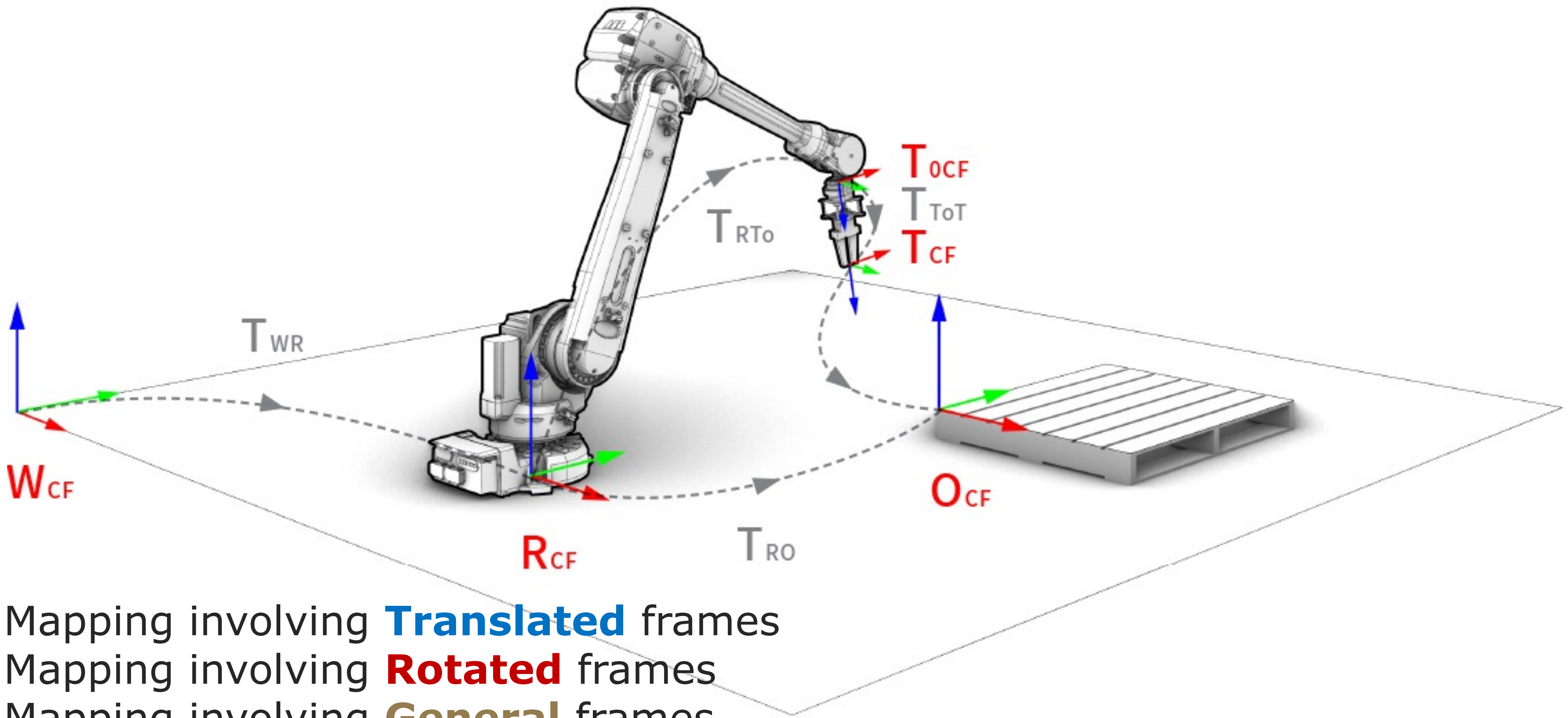
https://en.wikipedia.org/wiki/Frame_(linear_algebra)

Graphical representation of Frames

- {C} is known relative to {A} and not vice versa.
- A frame can be used as a description of one coordinate system relative to another.

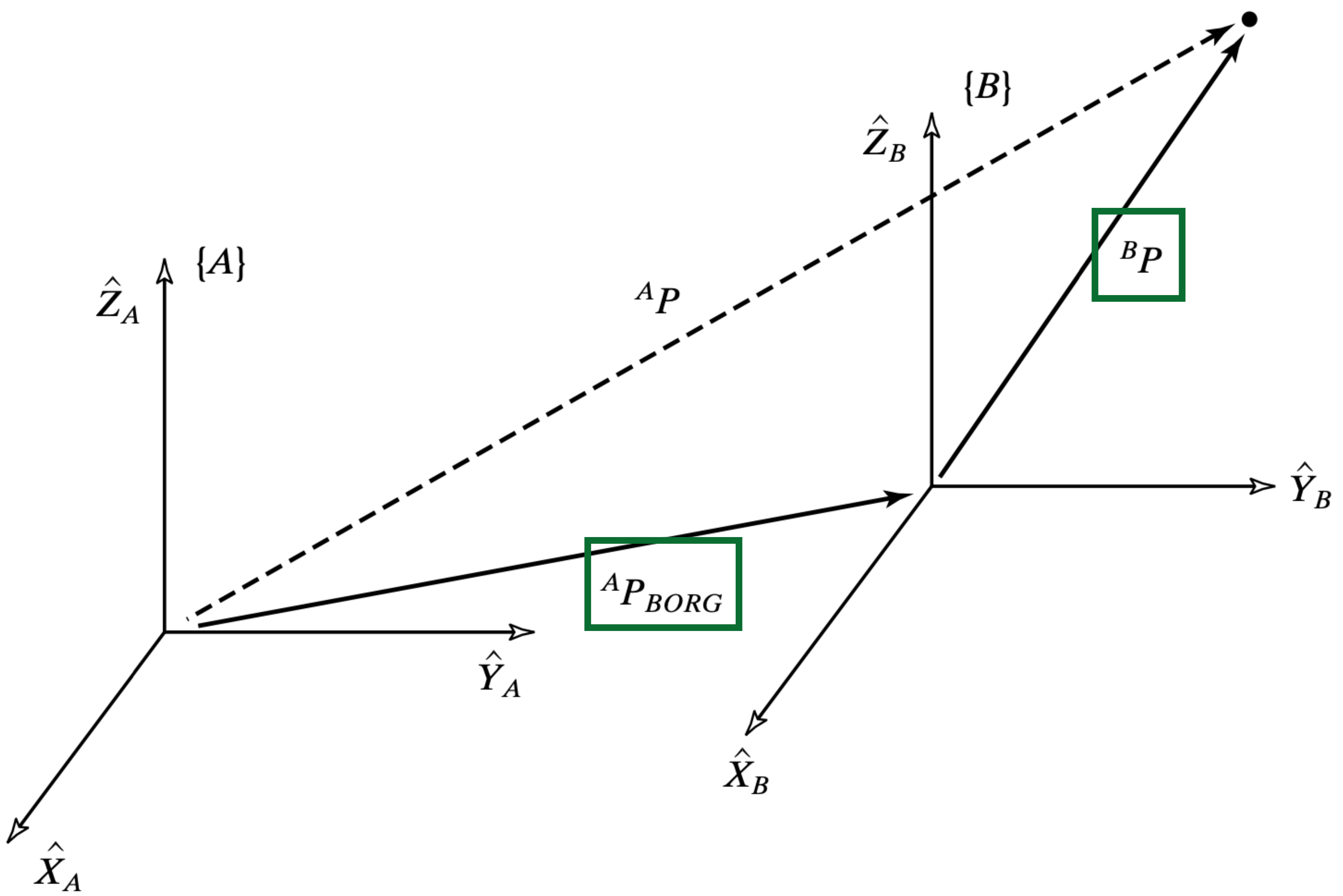


Mapping in Robotics

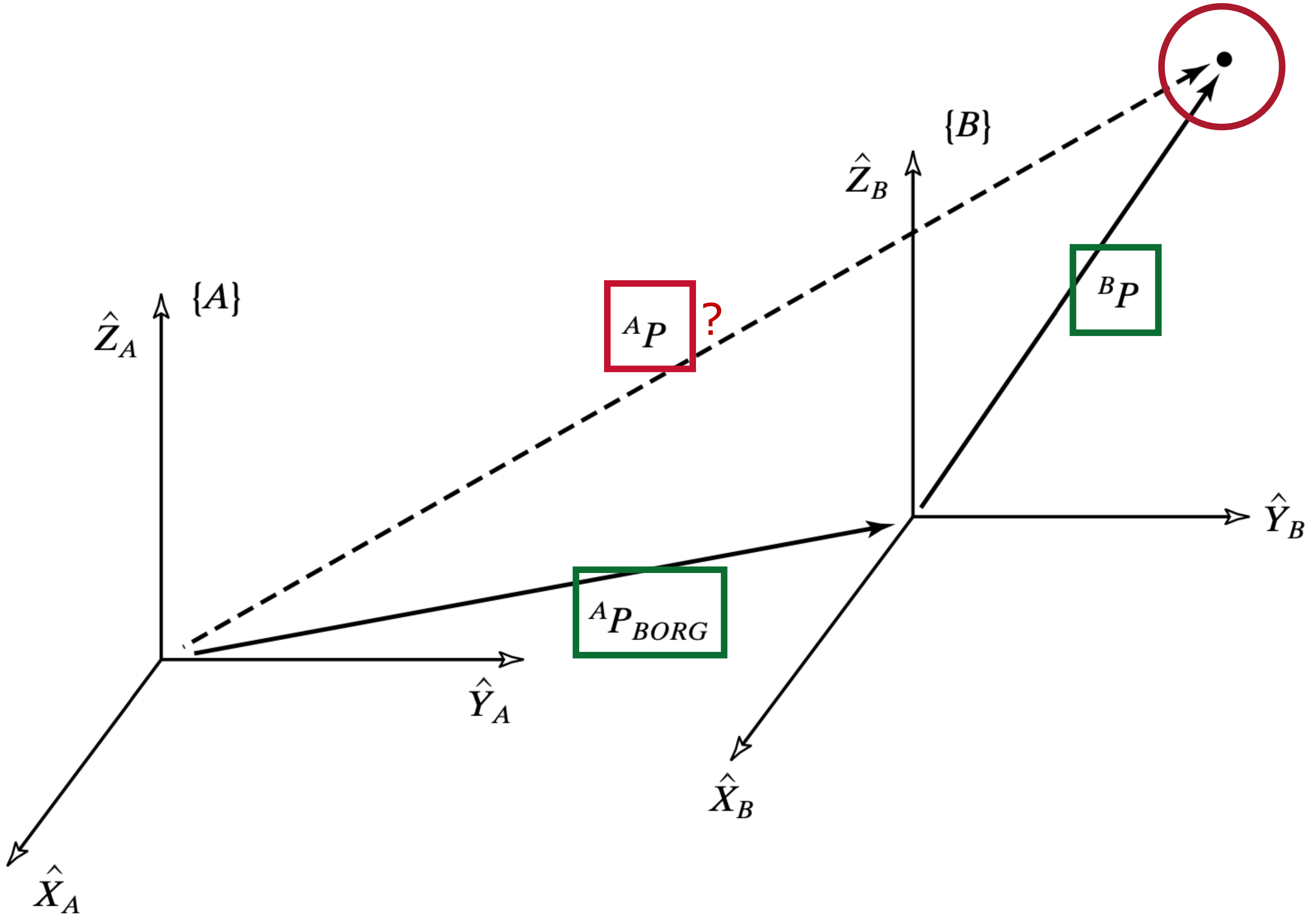


1. Mapping involving **Translated** frames
2. Mapping involving **Rotated** frames
3. Mapping involving **General** frames

Mapping involving Translated frames

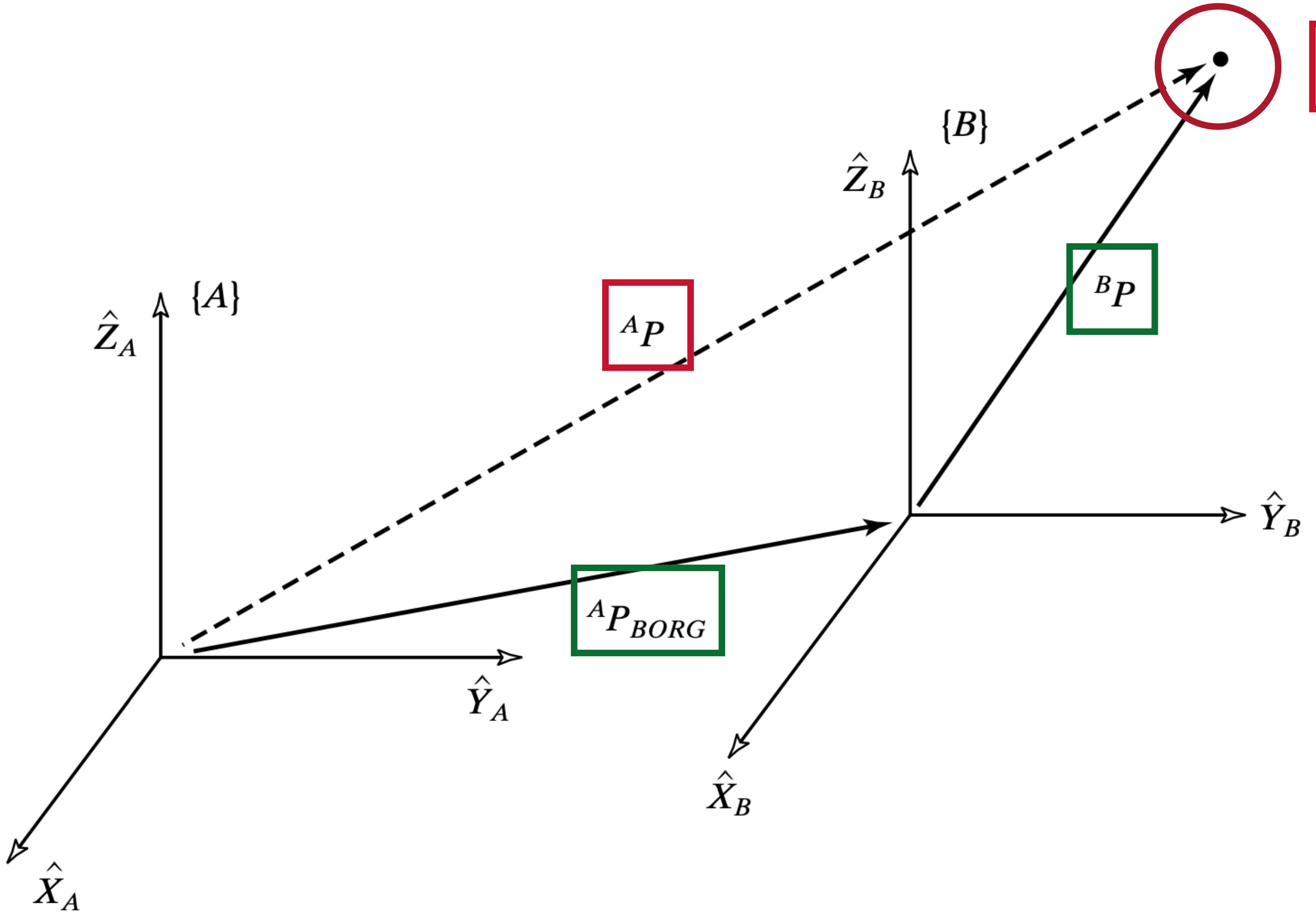


Mapping involving Translated frames

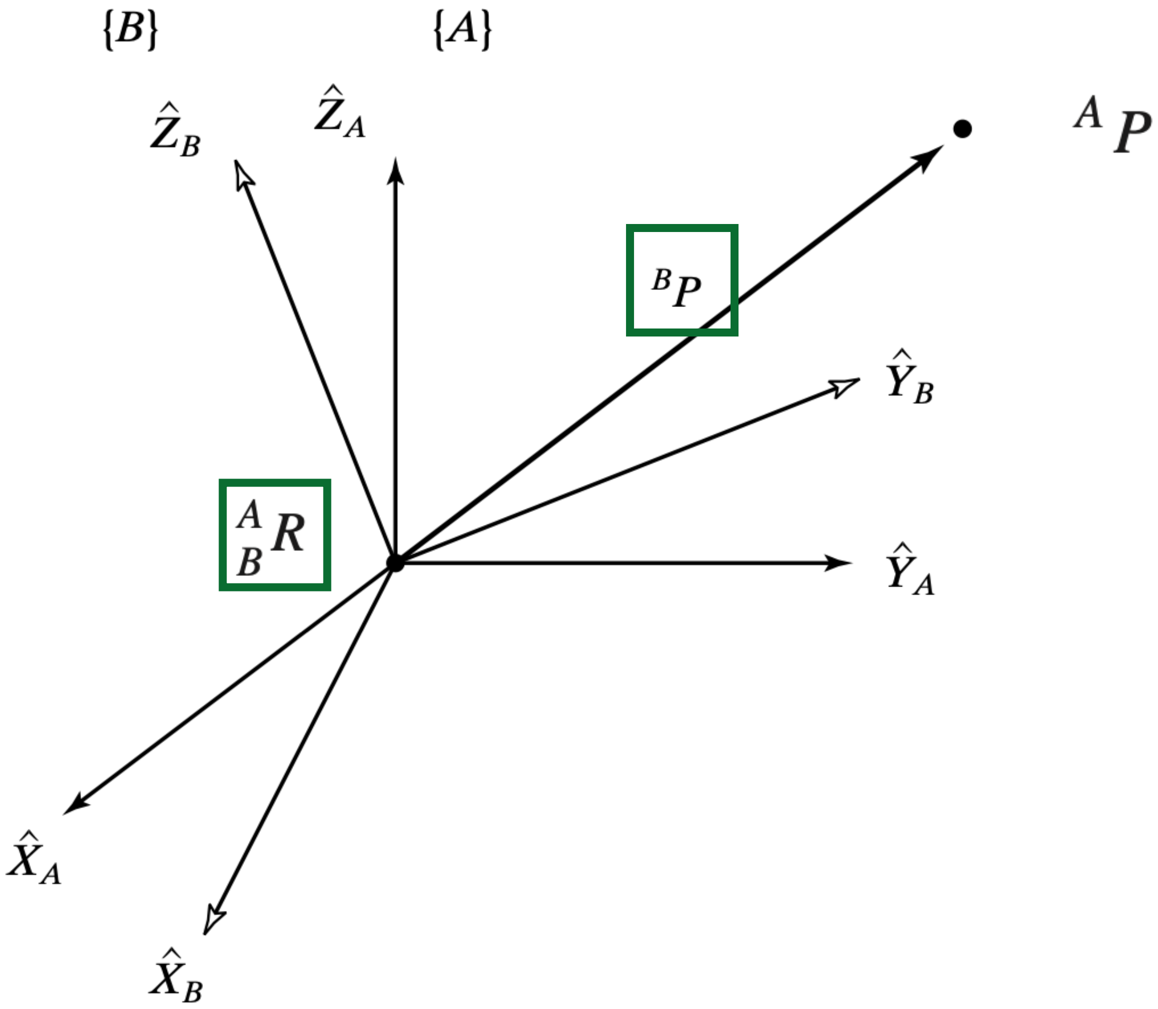


Mapping involving Translated frames

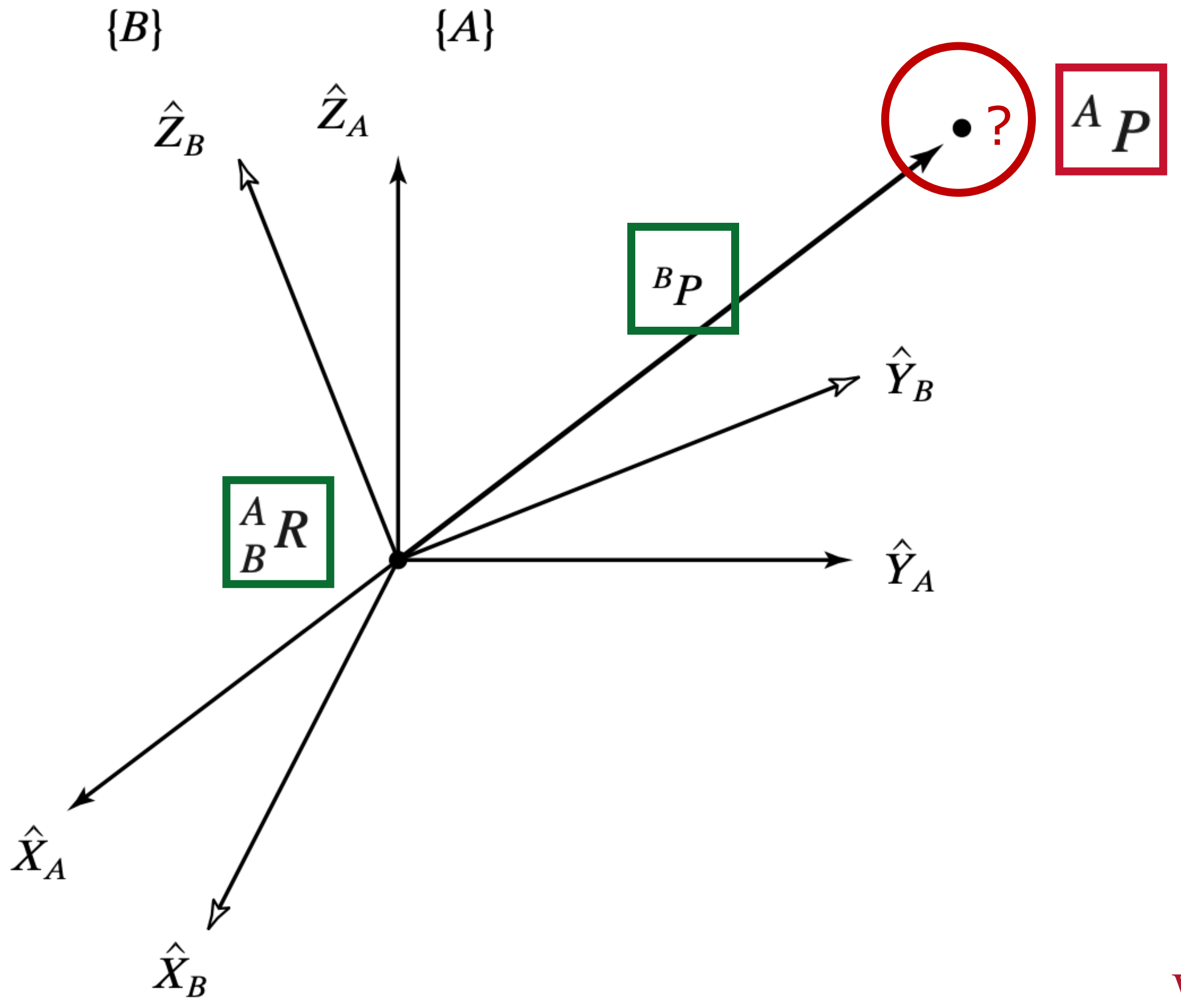
$$\boxed{{}^{A}P} = \boxed{{}^{B}P} + \boxed{{}^{A}P_{BORG}}$$



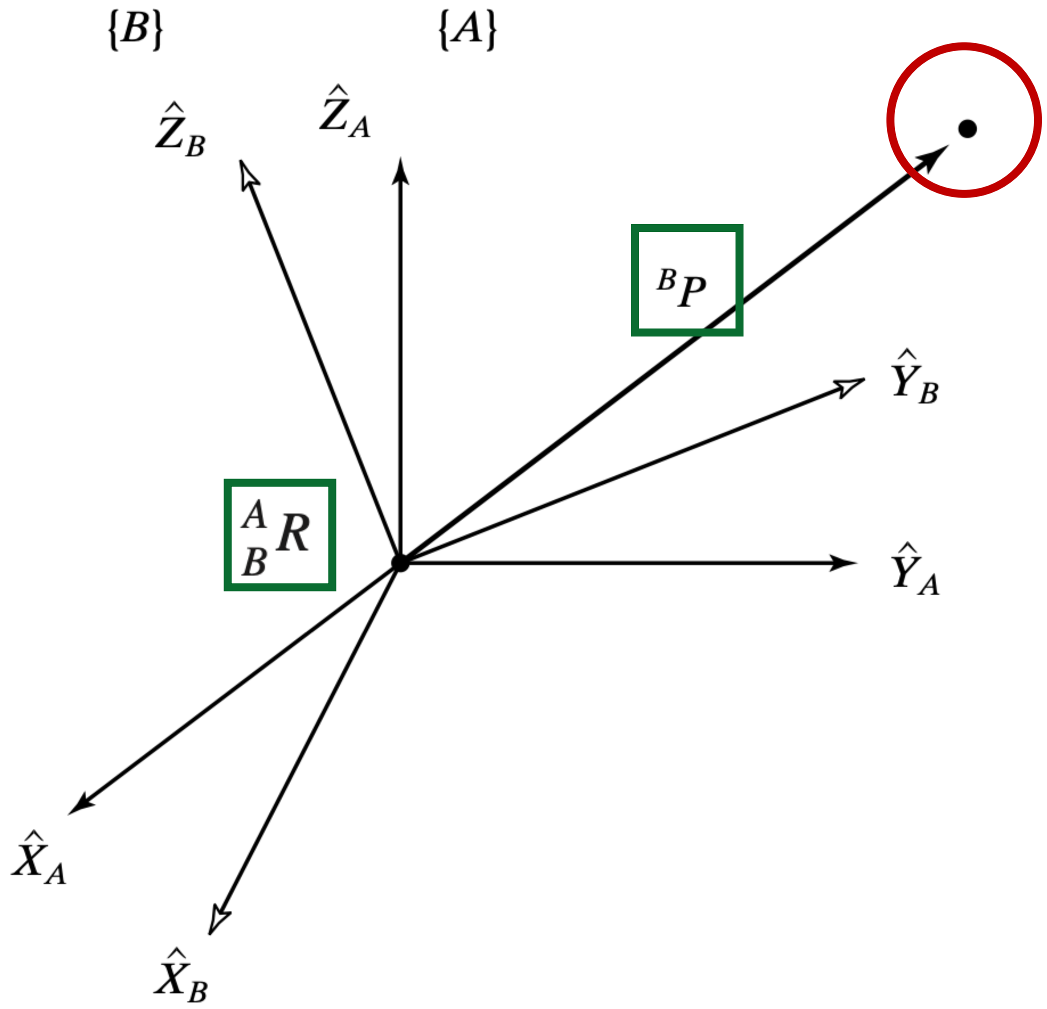
Mapping involving **Rotated** frames



Mapping involving **Rotated** frames



Mapping involving Rotated frames



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

Math Reminder: Basic Rotations and results

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

	30°	60°
sin	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$
cos	$\frac{\sqrt{3}}{2}$	$\frac{1}{2}$
tan	$\frac{1}{\sqrt{3}}$	$\sqrt{3}$

$\sin(90° + \theta) = \cos\theta$

$\cos(90° + \theta) = -\sin\theta$

$\sin(90° - \theta) = \cos\theta$

$\cos(90° - \theta) = \sin\theta$

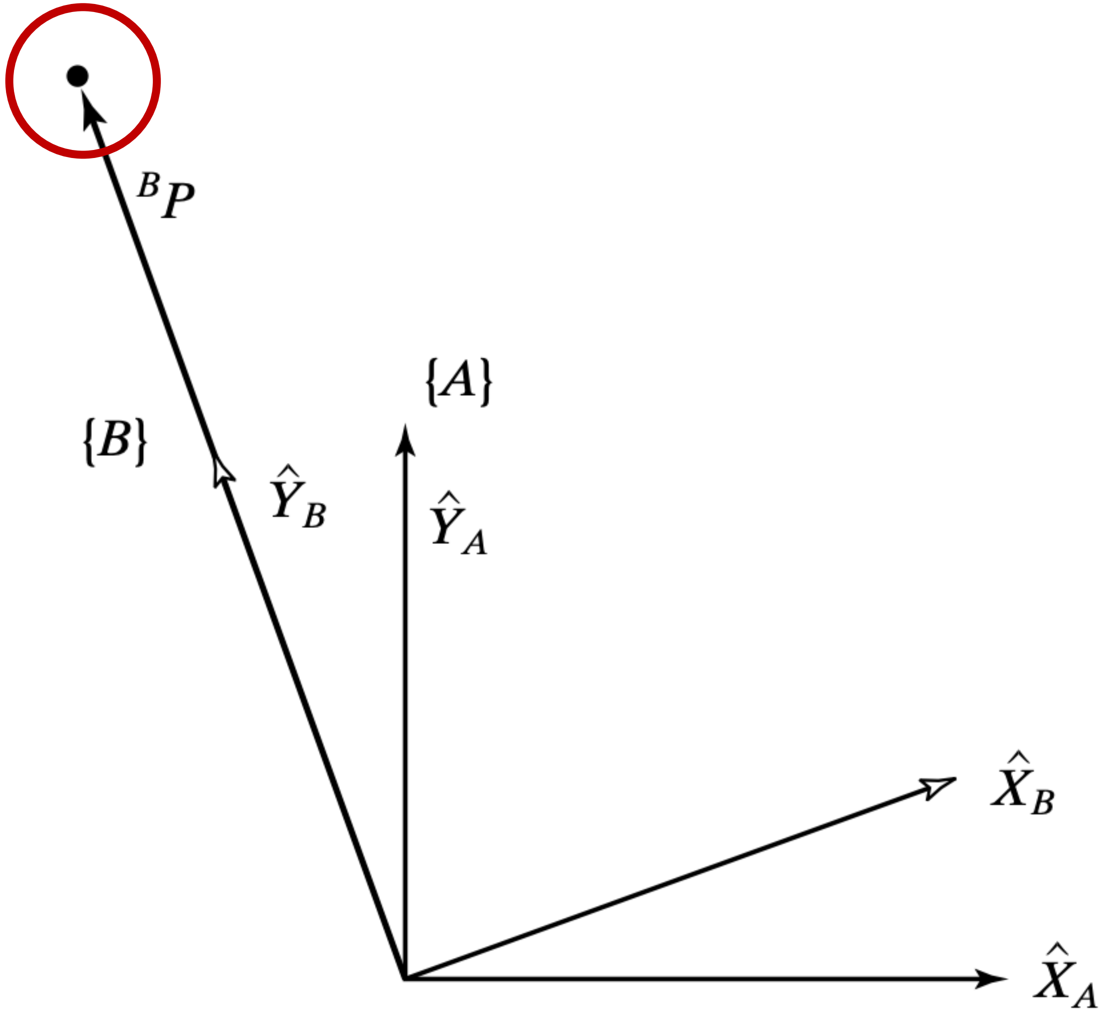
Math Reminder: Multiplication Example

- $$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} ax + by + cz \\ dx + ey + fz \\ gx + hy + iz \end{bmatrix}$$

- $$R_z(90^\circ)\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos 90^\circ & -\sin 90^\circ & 0 \\ \sin 90^\circ & \cos 90^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

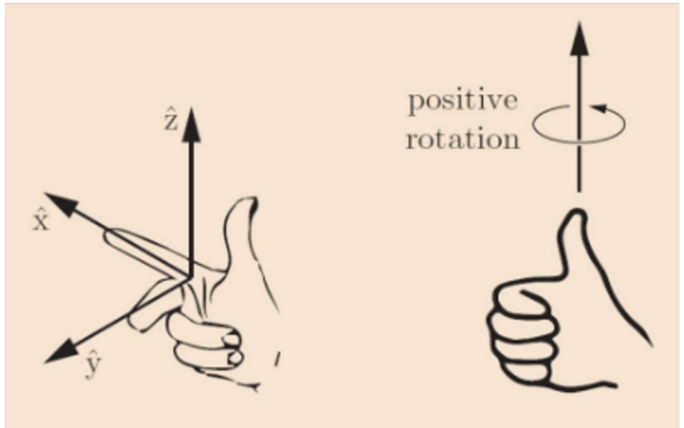
Mapping involving Rotated frames - Example 1

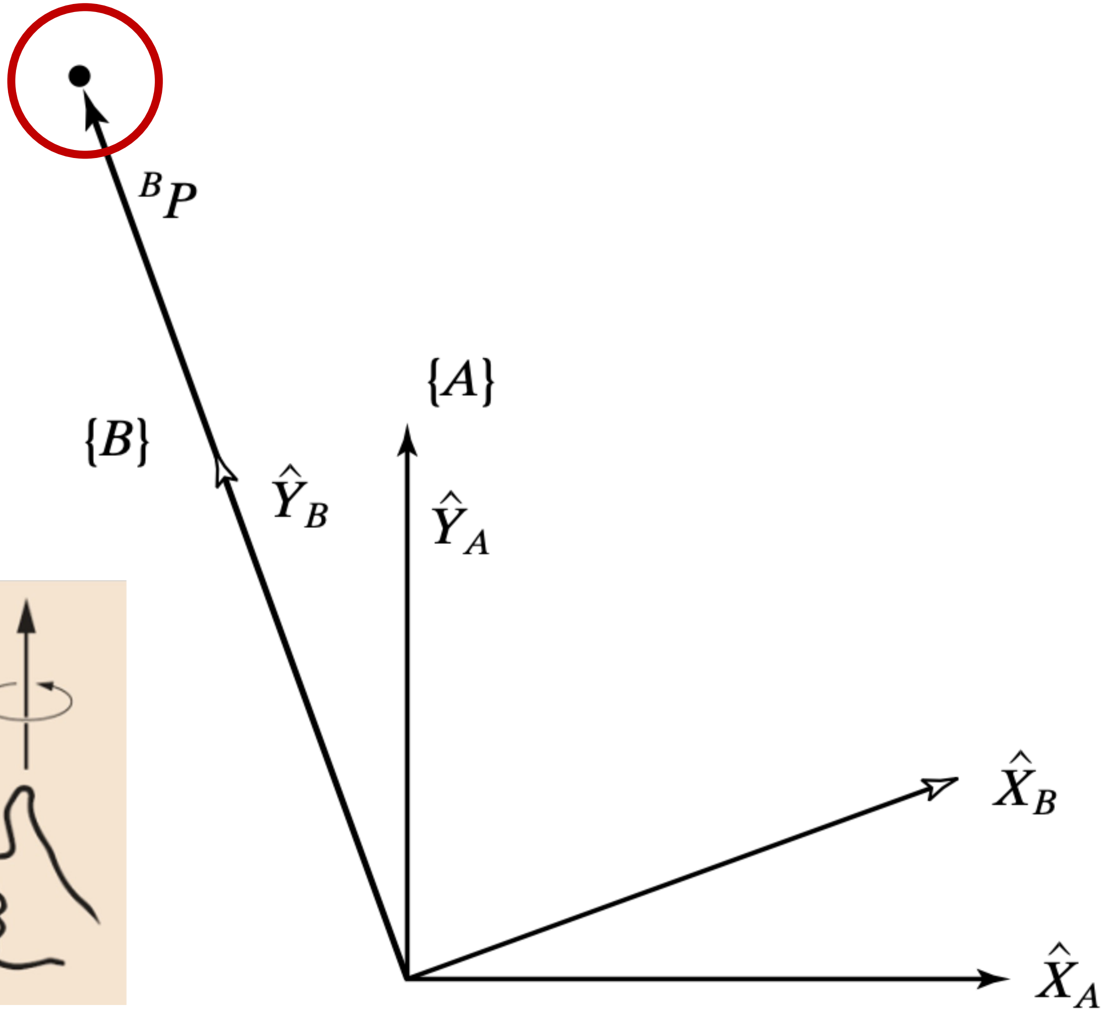
- A frame {B} is rotated relative to frame {A} about $\hat{Z}$ by 30 degrees.
- Given $^{B}P = \begin{bmatrix} 0.0 \\ 2.0 \\ 0.0 \end{bmatrix}$
- Calculate $\boxed{^{A}P}$?



Mapping involving Rotated frames - Example 1

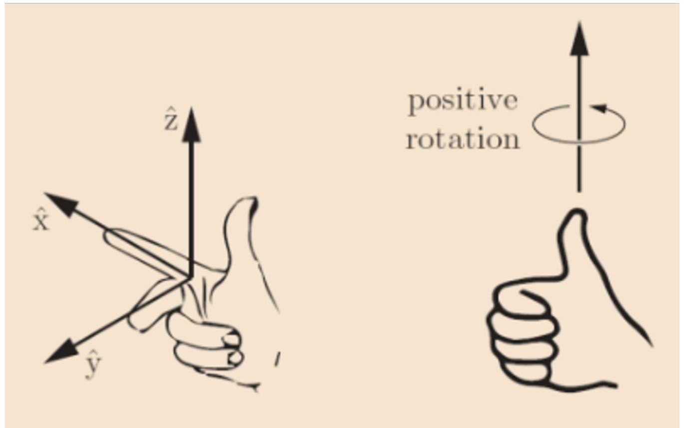
- A frame {B} is rotated relative to frame {A} about $\hat{Z}$ by 30 degrees.
- Given $^{B}P = \begin{bmatrix} 0.0 \\ 2.0 \\ 0.0 \end{bmatrix}$
- Calculate ^{A}P

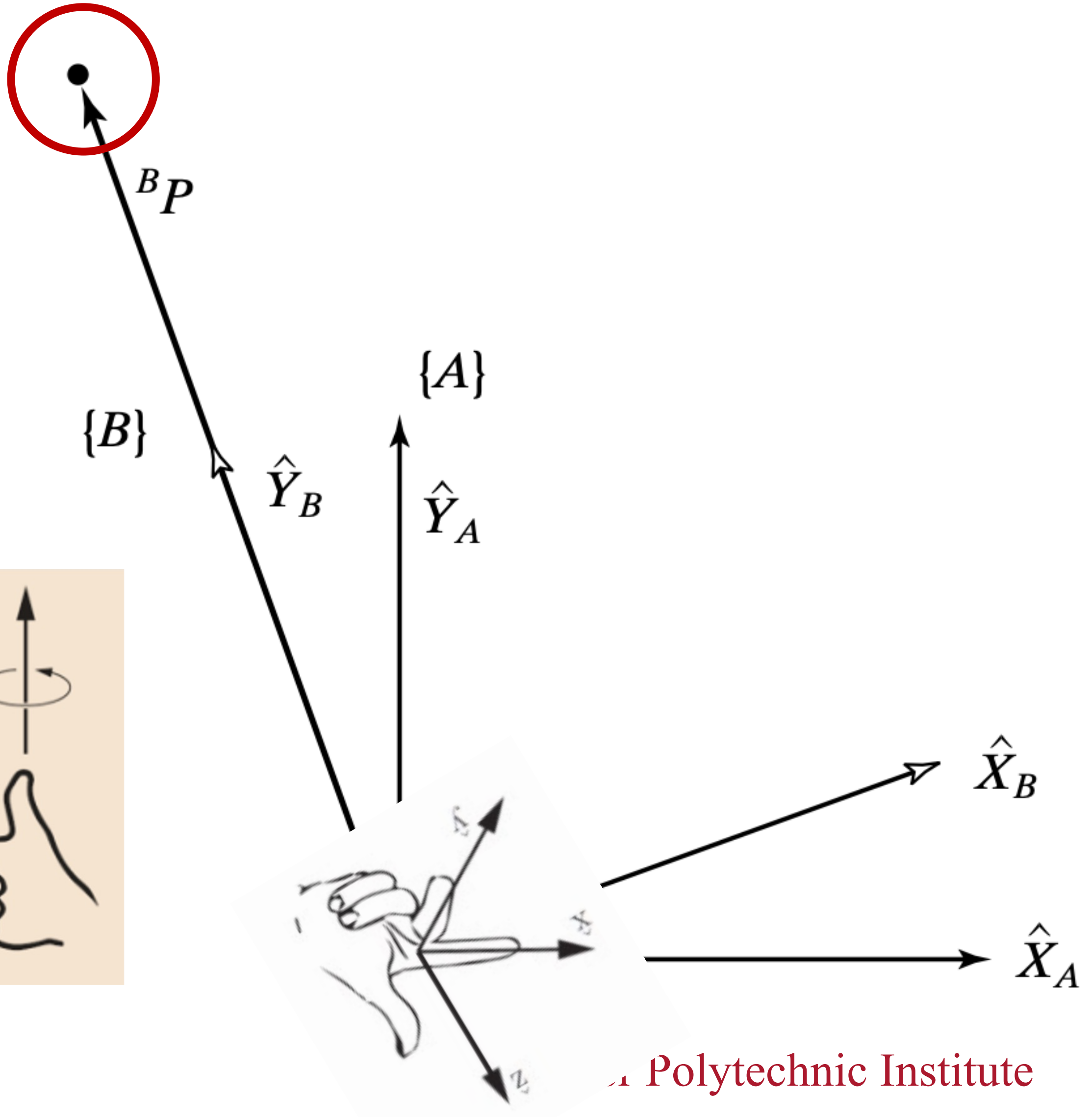




Mapping involving Rotated frames - Example 1

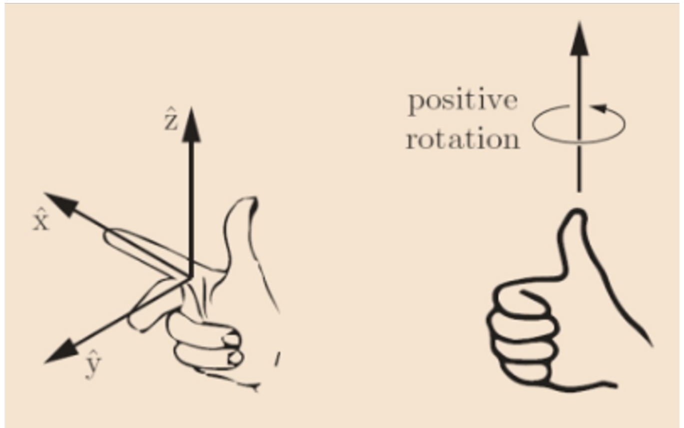
- A frame {B} is rotated relative to frame {A} about $\hat{Z}$ by 30 degrees.
- Given $^{B}P = \begin{bmatrix} 0.0 \\ 2.0 \\ 0.0 \end{bmatrix}$
- Calculate $\boxed{^{A}P}$?

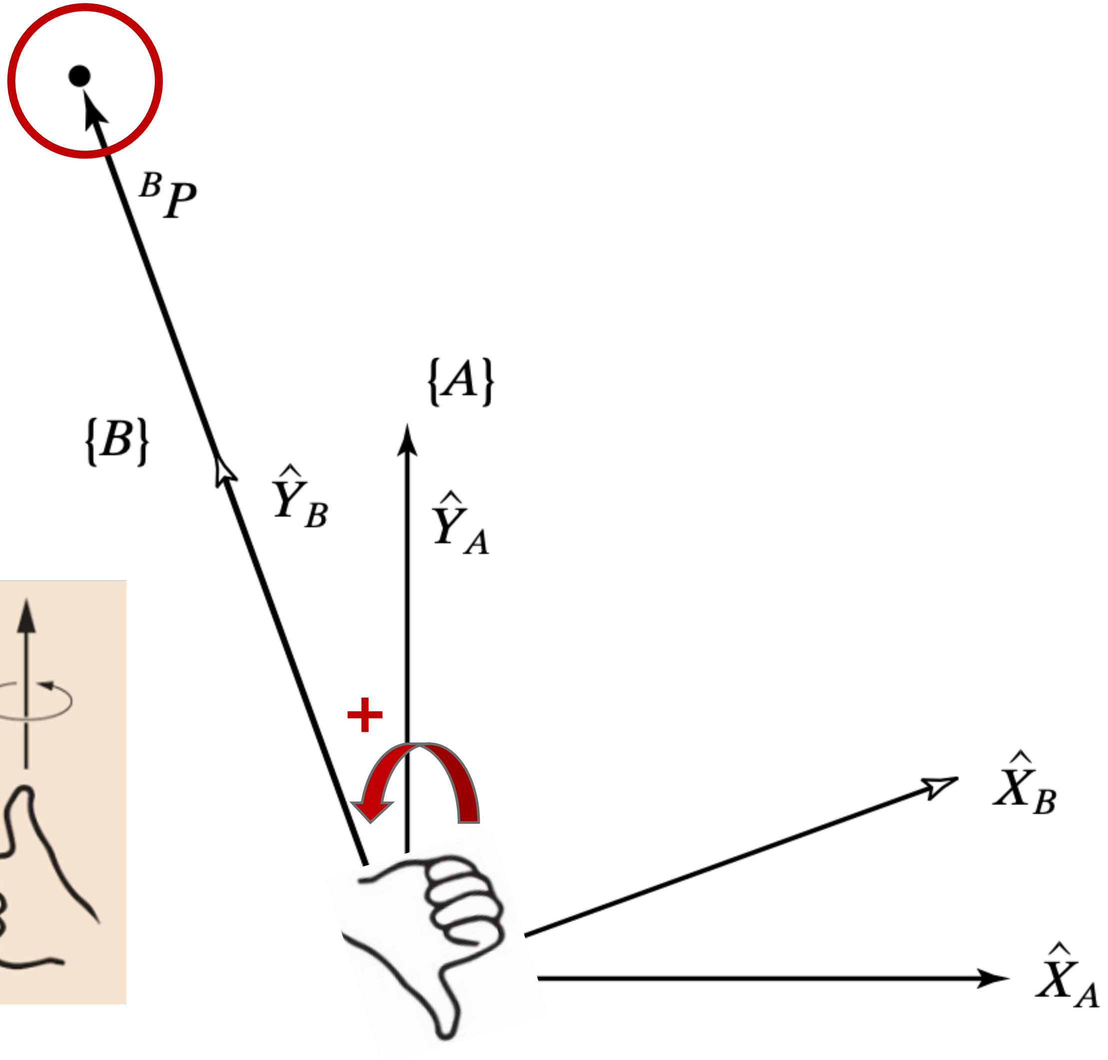




Mapping involving **Rotated** frames - **Example 1**

- A frame {B} is rotated relative to frame {A} about $\hat{Z}$ by 30 degrees.
- Given ${}^{B}P = \begin{bmatrix} 0.0 \\ 2.0 \\ 0.0 \end{bmatrix}$
- Calculate $\boxed{{}^{A}P}$





Mapping involving Rotated frames - Example 1

Solution in MATLAB

- $${}^{A}_{B}R = \begin{bmatrix} 0.866 & -0.500 & 0.000 \\ 0.500 & 0.866 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}$$

- $${}^{B}P = \begin{bmatrix} 0.0 \\ 2.0 \\ 0.0 \end{bmatrix}$$

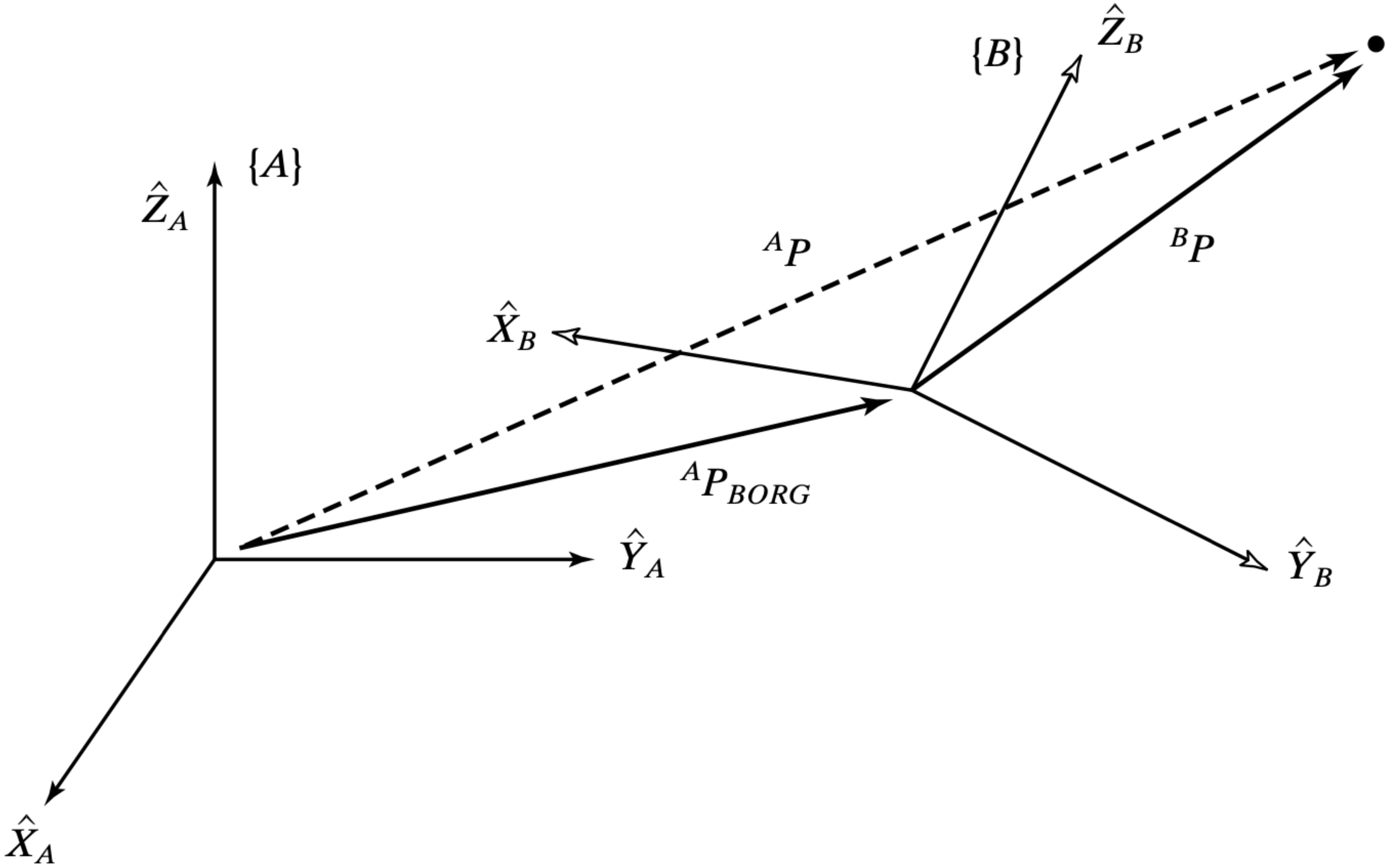
- $${}^{A}P = {}^{A}_{B}R\,{}^{B}P = \begin{bmatrix} -1.000 \\ 1.732 \\ 0.000 \end{bmatrix}$$

```
>> A_Rz30_B = rotz(30)

A_Rz30_B =

    0.8660   -0.5000         0
    0.5000    0.8660         0
         0         0    1.0000
```

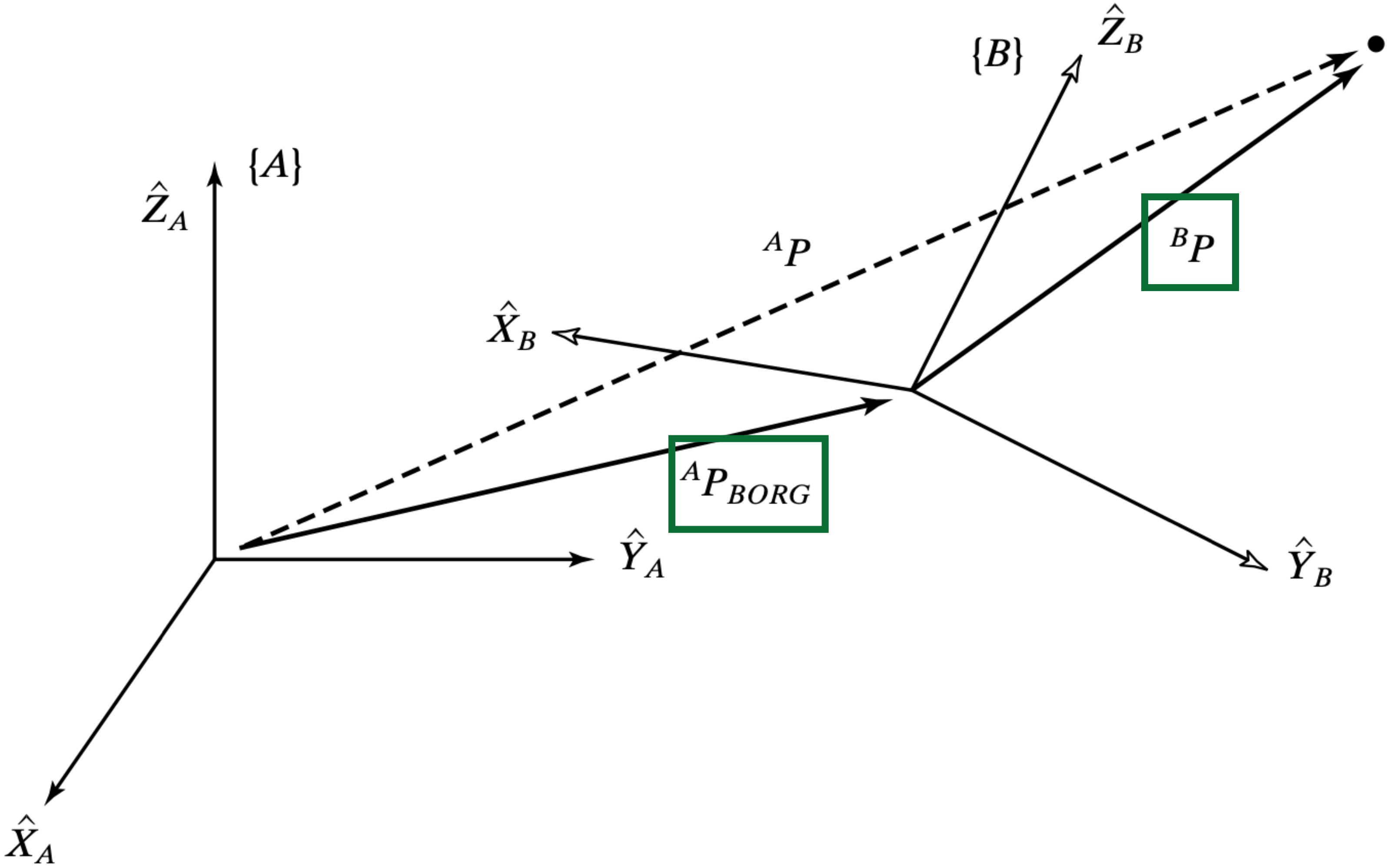
```
>> B_P = [0.0; 2.0; 0.0]

B_P =

     0
     2
     0
```

```
>> A_P = A_Rz30_B * B_P

A_P =

   -1.0000
    1.7321
         0
```

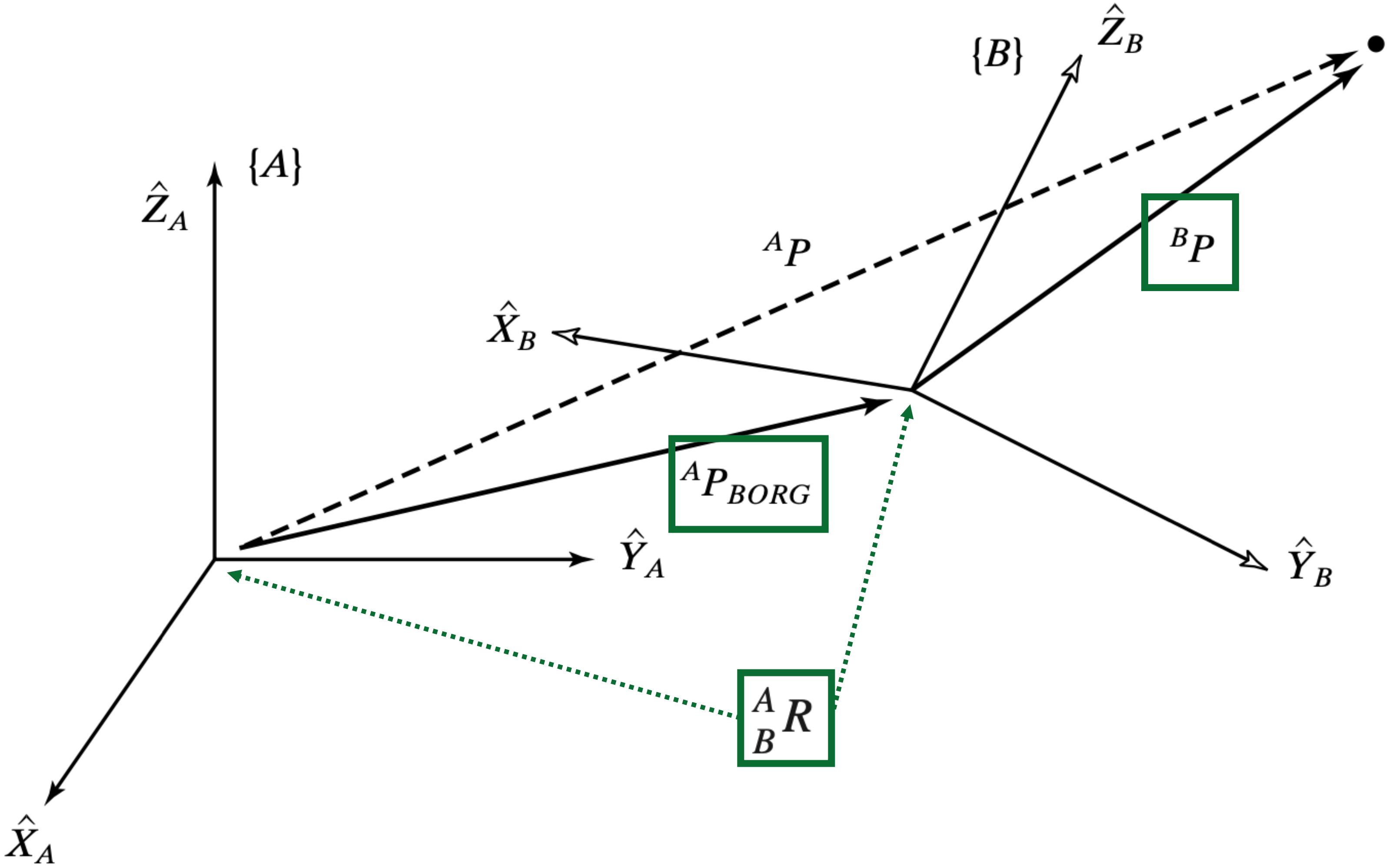
Mapping involving General frames



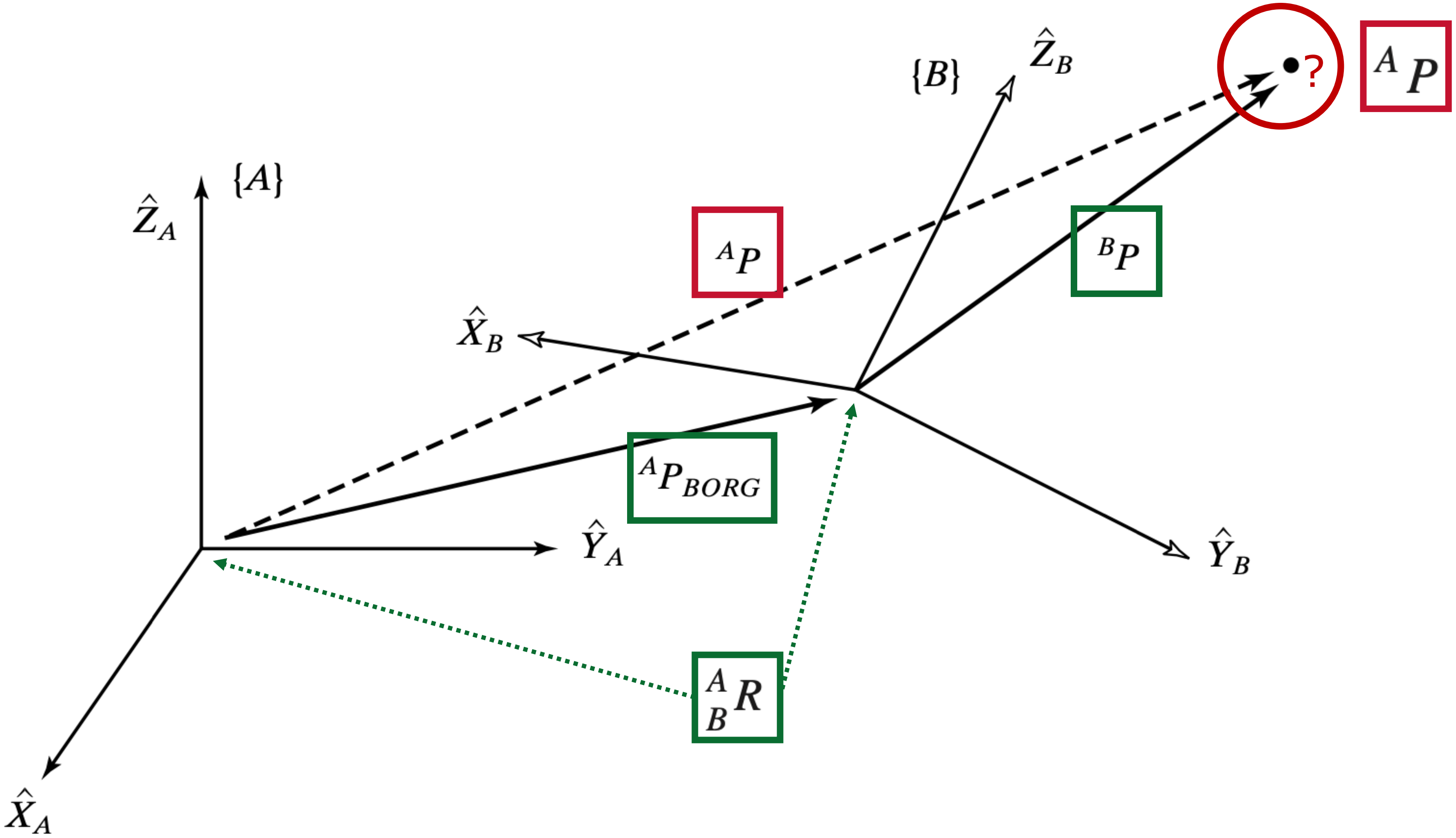
Mapping involving General frames



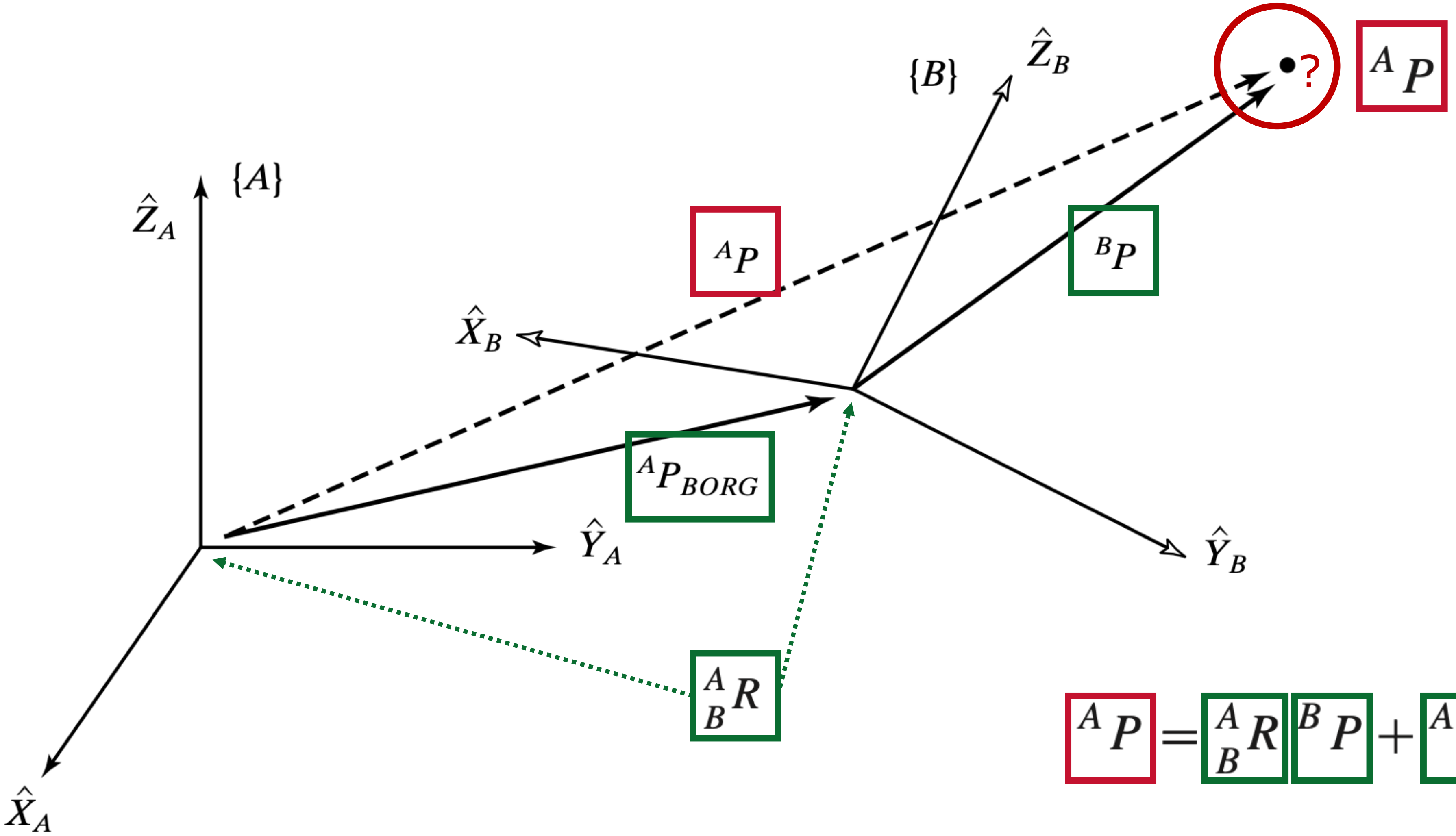
Mapping involving General frames



Mapping involving General frames



Mapping involving General frames



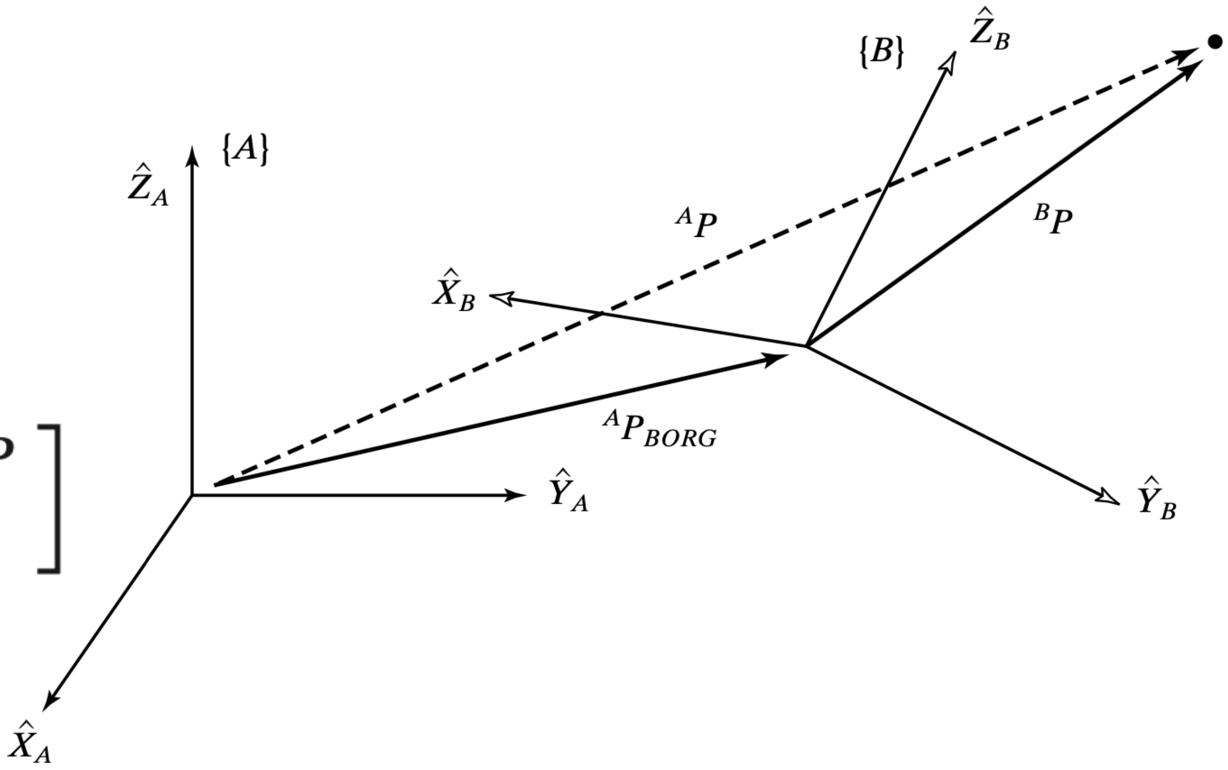
$${}^{A}P = {}^{A}_{B}R\,{}^{B}P + {}^{A}P_{BORG}$$

Mapping involving General frames

- $$^{A}P = {}^{A}_{B}R\,{}^{B}P + {}^{A}P_{BORG}$$
- Conceptual form: $^{A}P = {}^{A}_{B}T\,{}^{B}P$
- Homogeneous Transform:

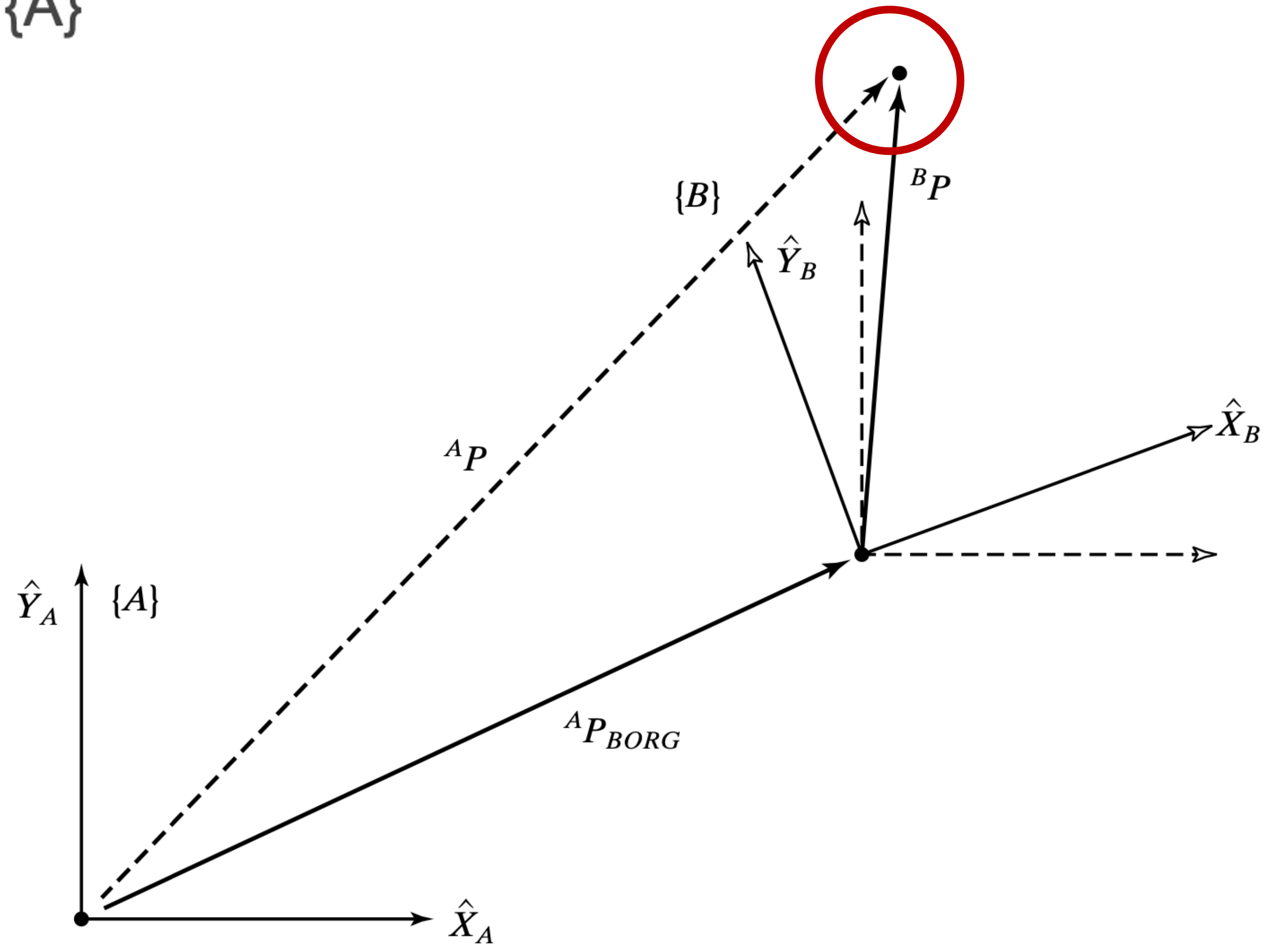
$$\begin{bmatrix} ^{A}P \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} & {}^{A}_{B}R & & {}^{A}P_{BORG} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} ^{B}P \\ 1 \end{bmatrix}$$

$$^{A}_{B}T = \left[\begin{array}{c|c} {}^{A}_{B}R & {}^{A}P_{BORG} \\ \hline 0_{1\times 3} & 1 \end{array}\right]$$



Mapping involving General frames - Example 2

- A frame {B} is rotated relative to frame {A} about $\widehat{Z_A}$ by 30 degrees, translated 10 units in $\widehat{X_A}$ and translated 5 units in $\widehat{Y_A}$.
- Given $^{B}P = \begin{bmatrix} 3.0 \\ 7.0 \\ 0.0 \end{bmatrix}$
- Calculate $\boxed{^{A}P}$?



Mapping involving General frames - Example 2

Solution in MATLAB

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

- $${}^{B}P = \begin{bmatrix} 3.0 \\ 7.0 \\ 0.0 \end{bmatrix}$$

- $${}^{A}P = {}^{A}_{B}T\ {}^{B}P = \begin{bmatrix} 9.098 \\ 12.562 \\ 0.000 \end{bmatrix}$$

$${}^{A}_{B}T = \left[\begin{array}{c|c} {}^{A}_{B}R & {}^{A}P_{BORG} \\ \hline 0_{1\times 3} & 1 \end{array}\right]$$

-
```
>> A_Rz30_B = rotz(30)

A_Rz30_B =

    0.8660   -0.5000         0
    0.5000    0.8660         0
         0         0    1.0000
```

-
```
>> A_P_BORG = [10; 5; 0]

A_P_BORG =

    10
     5
     0
```

-
```
>> A_T_B = [A_Rz30_B A_P_BORG; 0 0 0 1]

A_T_B =

    0.8660   -0.5000         0   10.0000
    0.5000    0.8660         0    5.0000
         0         0    1.0000         0
         0         0         0    1.0000
```

-
```
>> A_P = A_T_B * [B_P; 1]

A_P =

    9.0981
   12.5622
         0
    1.0000
```

Math Reminder: Homogeneous Transformations

- Homogeneous Transform:

$$ {}^{A}_{B}T = \left[\begin{array}{c|c} {}^{A}_{B}R & {}^{A}P_{BORG} \\ \hline 0_{1\times3} & 1 \end{array}\right] $$

- Pure Rotation:

$$ {}^{A}_{B}T = \begin{bmatrix} {}^{A}_{B}R & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} $$

- Pure Translation:

$$ {}^{A}_{B}T = \begin{bmatrix} I_3 & {}^{A}P_{BORG} \\ 0_{1\times3} & 1 \end{bmatrix} $$

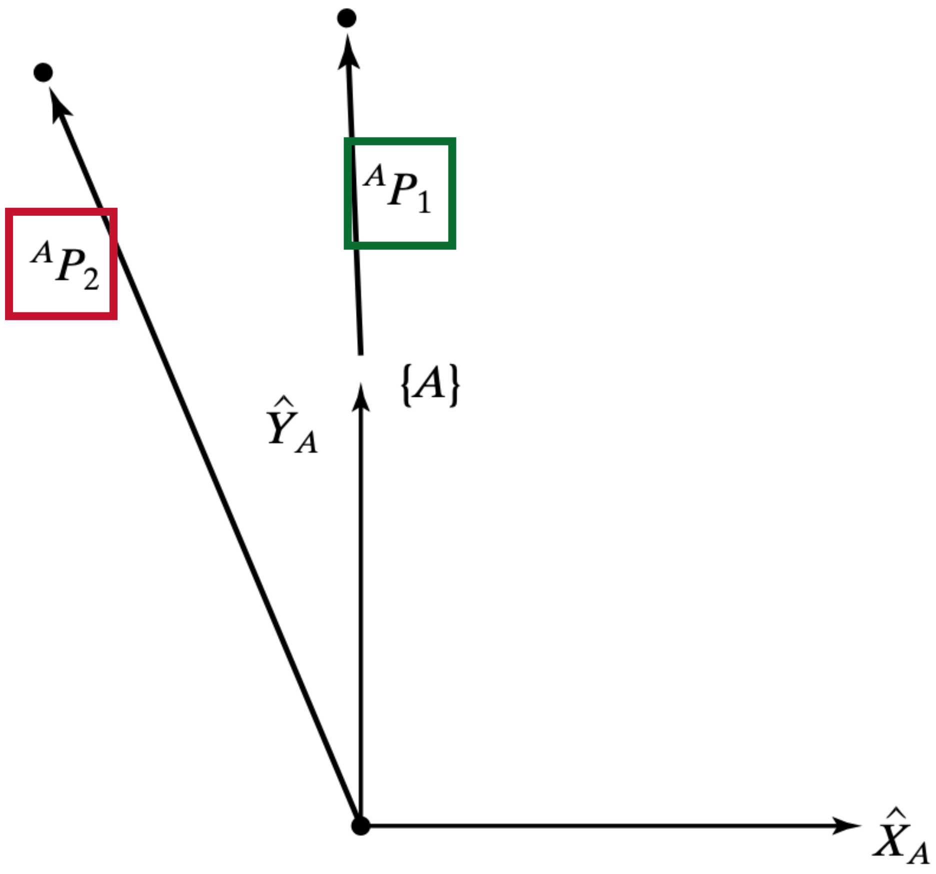
- Identity Matrix:

$$ I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

Transform Operator with Vectors - Example 3

- Example: We want to compute a new vector ${}^{A}P_2$ by rotating the vector ${}^{A}P_1$ about $\widehat{Z_A}$ by 30 degrees.

Given: $${}^{A}P_1 = \begin{bmatrix} 0.0 \\ 2.0 \\ 0.0 \end{bmatrix}$$



Transform Operator with Vectors - Example 3

Solution in MATLAB

- $R_z(30.0) = \begin{bmatrix} 0.866 & -0.500 & 0.000 \\ 0.500 & 0.866 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}$
- $^{A}P_1 = \begin{bmatrix} 0.0 \\ 2.0 \\ 0.0 \end{bmatrix}$
- $^{A}P_2 = R_z(30.0)\ {}^{A}P_1 = \begin{bmatrix} -1.000 \\ 1.732 \\ 0.000 \end{bmatrix}$

```
>> Rz30 = rotz(30)

Rz30 =

    0.8660   -0.5000         0
    0.5000    0.8660         0
         0         0    1.0000
```

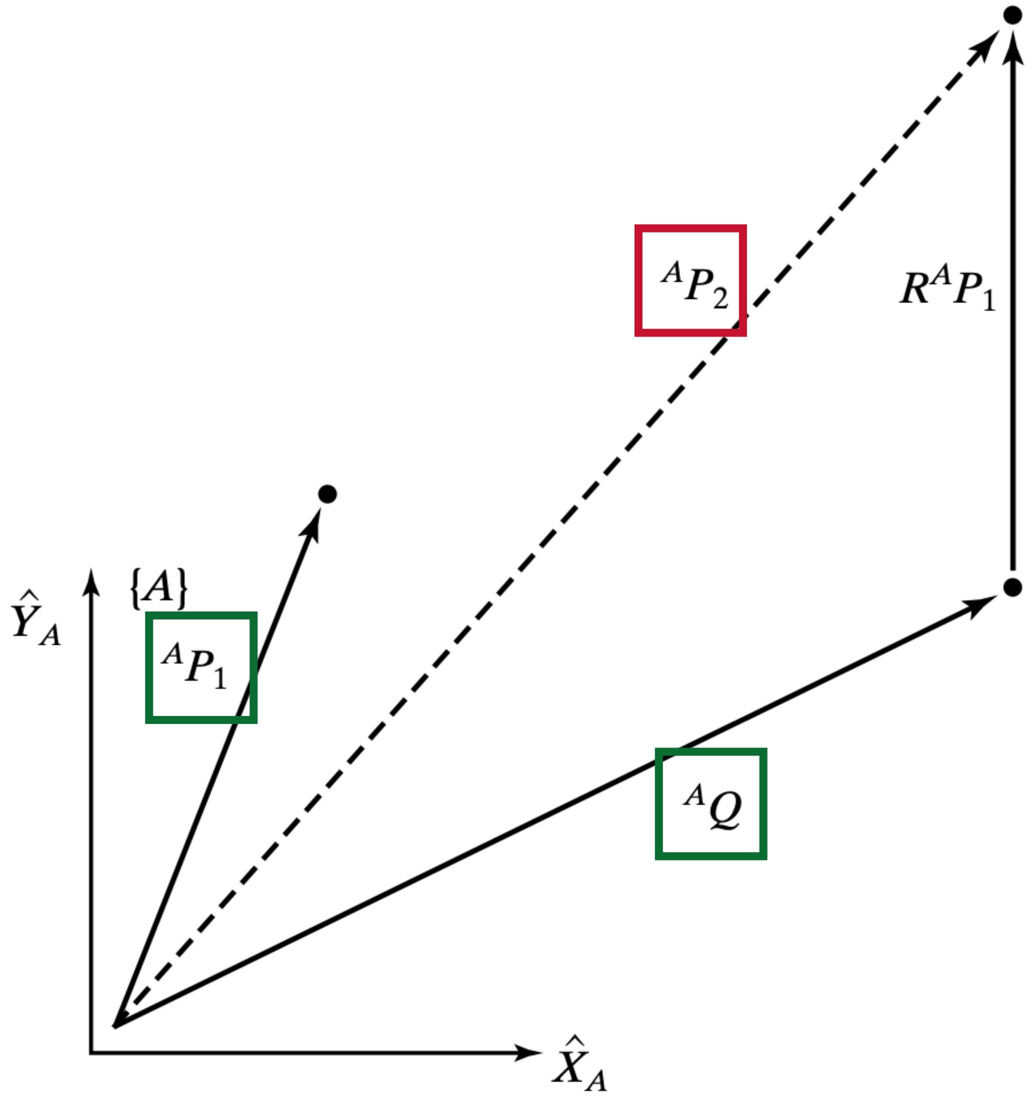
```
>> A_P1 = [0; 2; 0]

A_P1 =

     0
     2
     0
```

```
>> A_P2 = Rz30 * A_P1

A_P2 =

   -1.0000
    1.7321
         0
```

Transform Operator with Vectors - Example 4

- Example: We want to compute a new vector ${}^{A}P_2$ by rotating the vector ${}^{A}P_1$ about $\widehat{Z_A}$ by 30 degrees and translating it 10 units in $\widehat{X_A}$ and 5 units in $\widehat{Y_A}$.

Given: $${}^{A}P_1 = \begin{bmatrix} 3.0 \\ 7.0 \\ 0.0 \end{bmatrix}$$



Transform Operator with Vectors - Example 4

Solution and MATLAB code

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

- $${}^{A}P_1 = \begin{bmatrix} 3.0 \\ 7.0 \\ 0.0 \end{bmatrix}$$

- $${}^{A}P_2 = T\,{}^{A}P_1 = \begin{bmatrix} 9.098 \\ 12.562 \\ 0.000 \end{bmatrix}$$

-
```
>> A_Rz30_B = rotz(30)

A_Rz30_B =

    0.8660   -0.5000         0
    0.5000    0.8660         0
         0         0    1.0000
```

-
```
>> A_Q = [10; 5; 0]

A_Q =

    10
     5
     0
```

-
```
>> T = [A_Rz30_B A_Q; 0 0 0 1]

T =

    0.8660   -0.5000         0   10.0000
    0.5000    0.8660         0    5.0000
         0         0    1.0000         0
         0         0         0    1.0000
```

-
```
>> A_P1 = [3; 7; 0]

A_P1 =

     3
     7
     0
```

-
```
>> A_P2 = T * [A_P1; 1]

A_P2 =

    9.0981
   12.5622
         0
    1.0000
```

… end of Lecture 2