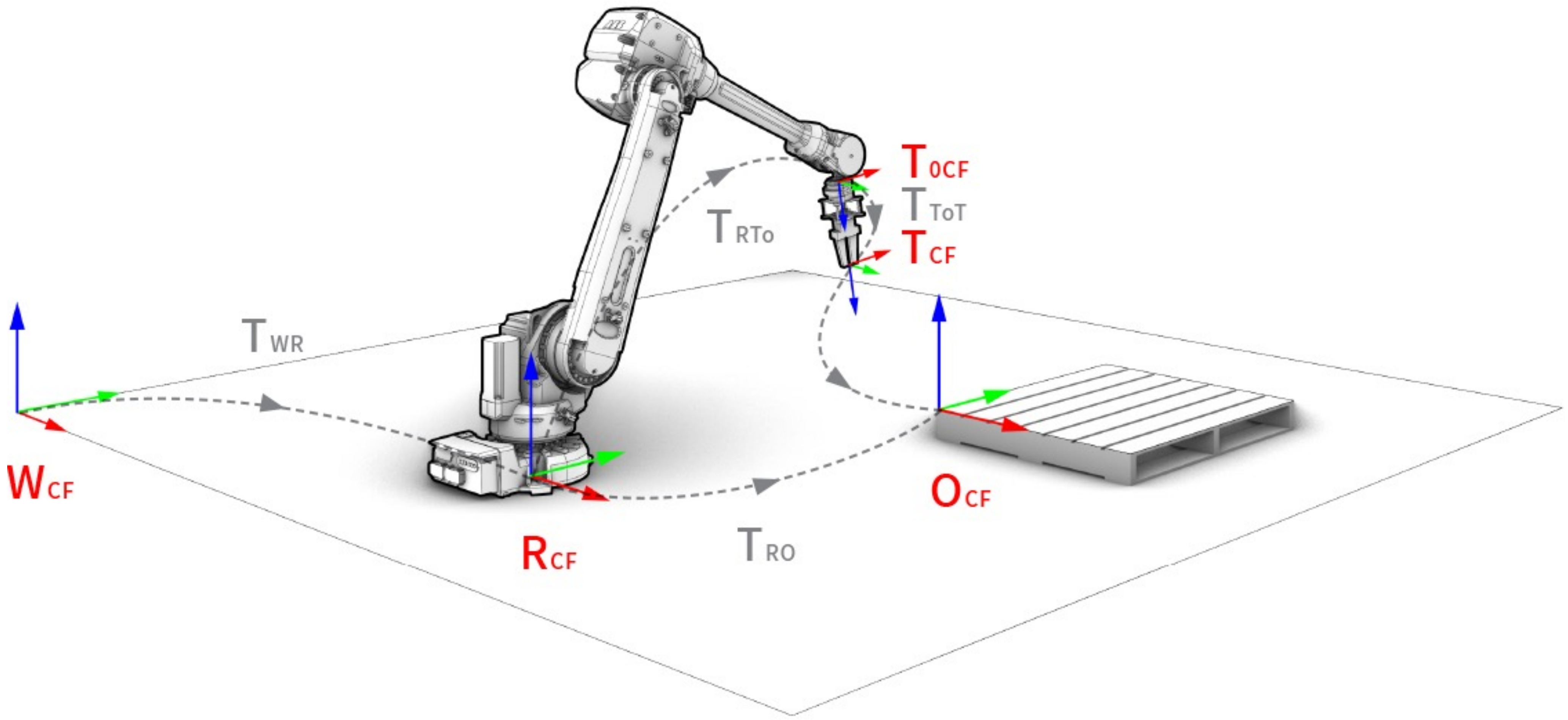
Lecture 3

Compound Transformations

Properties and representation of Orientation

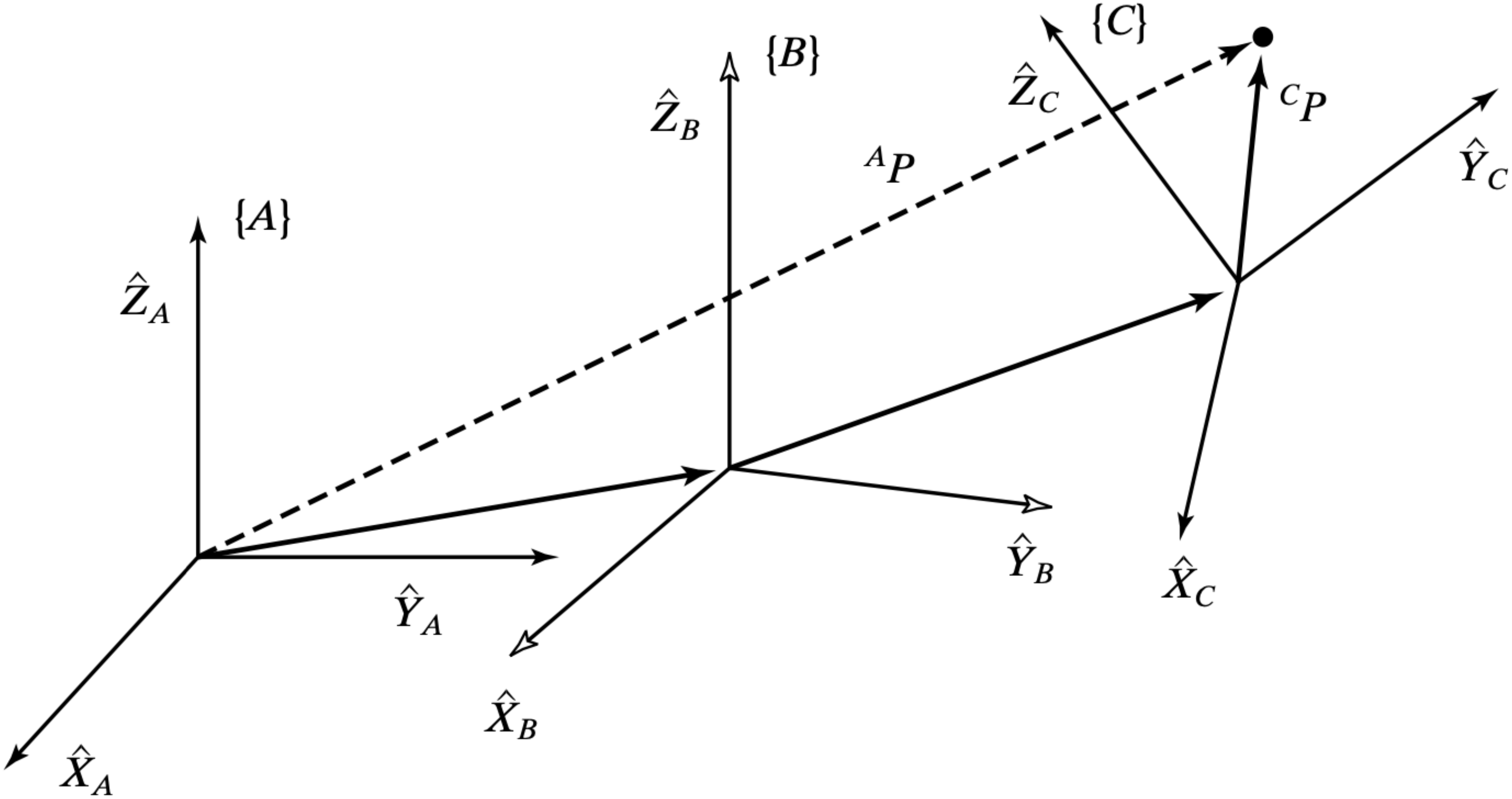
Alexandros Lioulemes, PhD

Compound Transformations

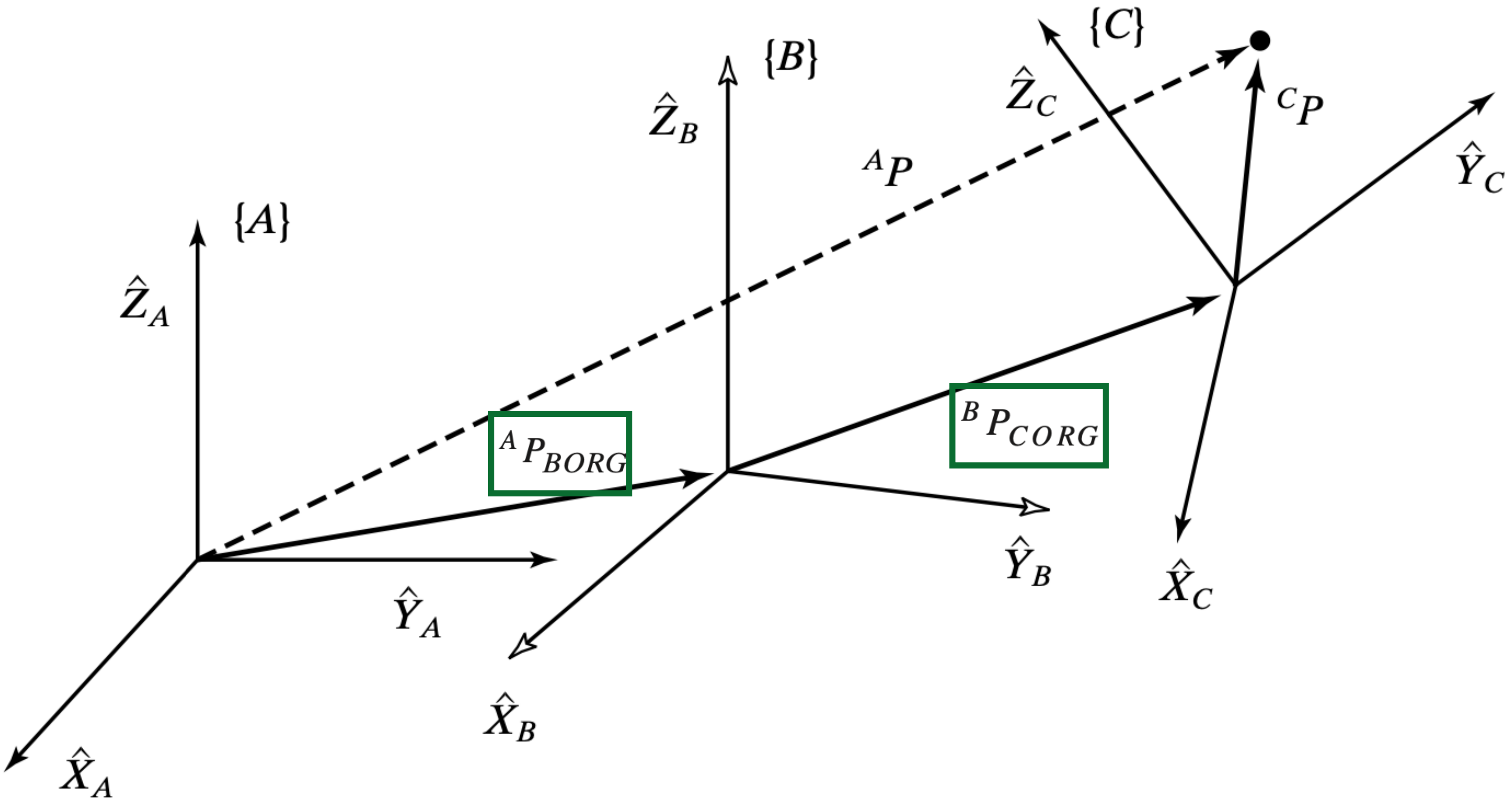


https://en.wikipedia.org/wiki/Compound_matrix

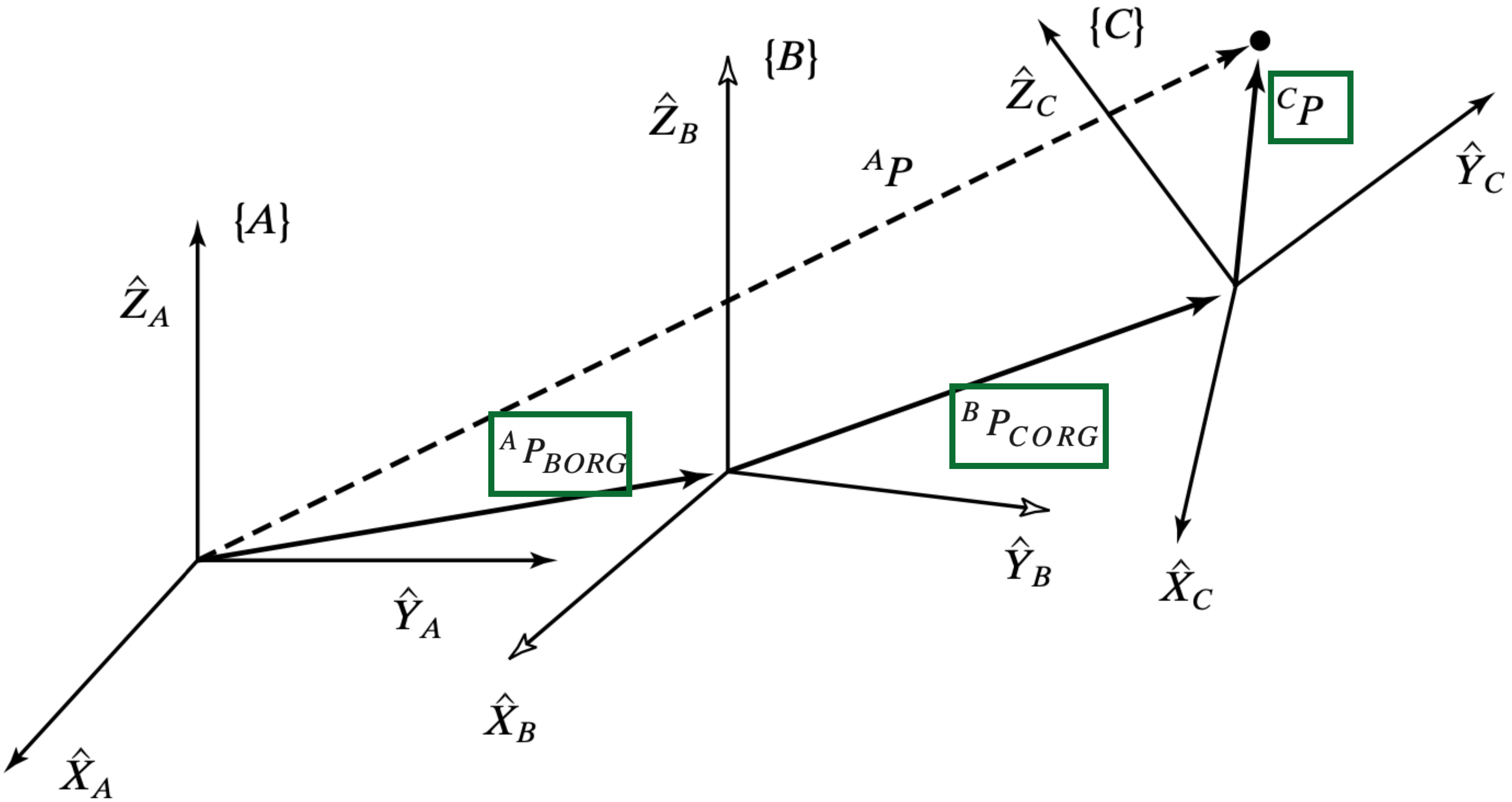
Compound Transformations



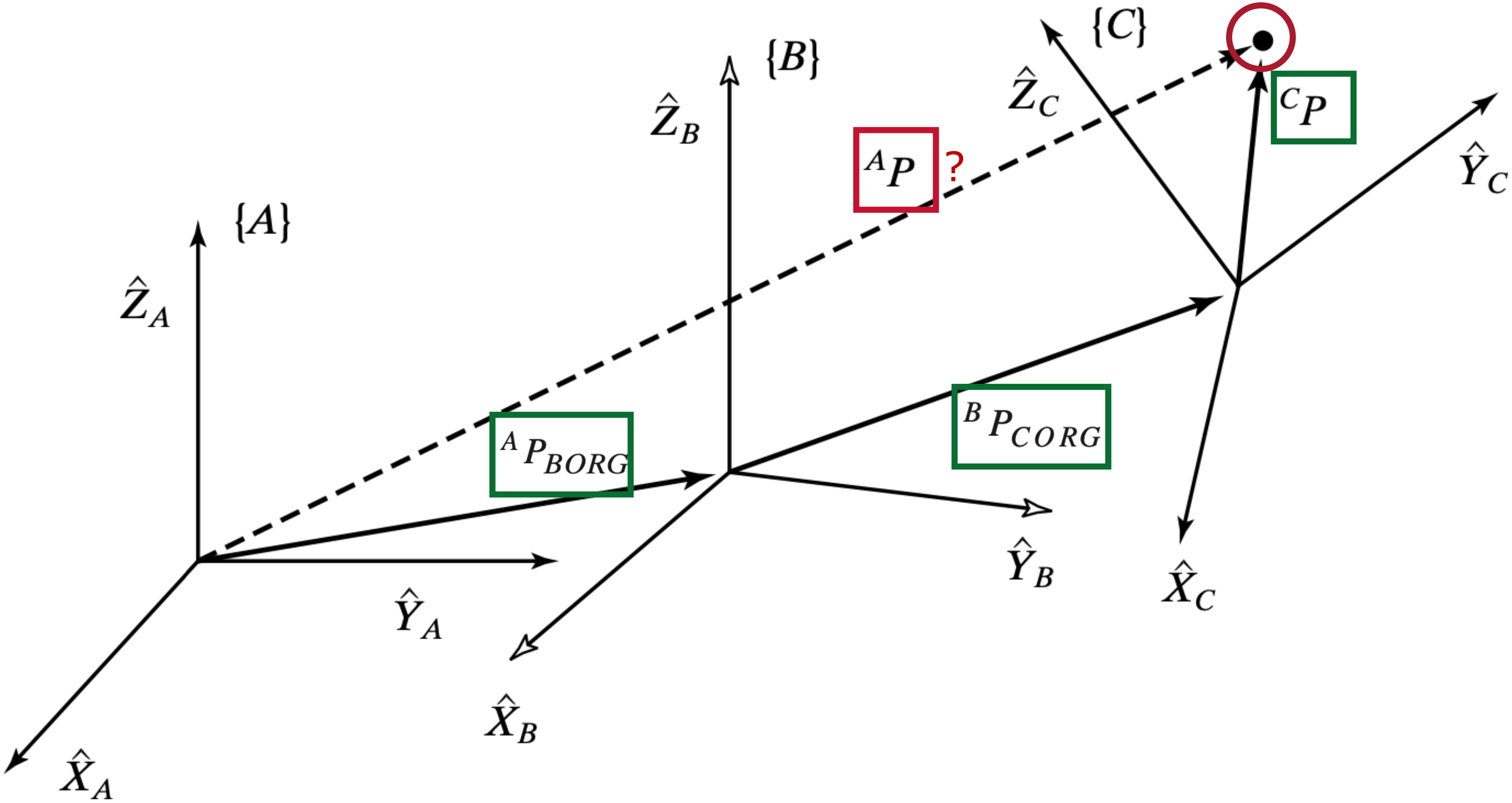
Compound Transformations



Compound Transformations



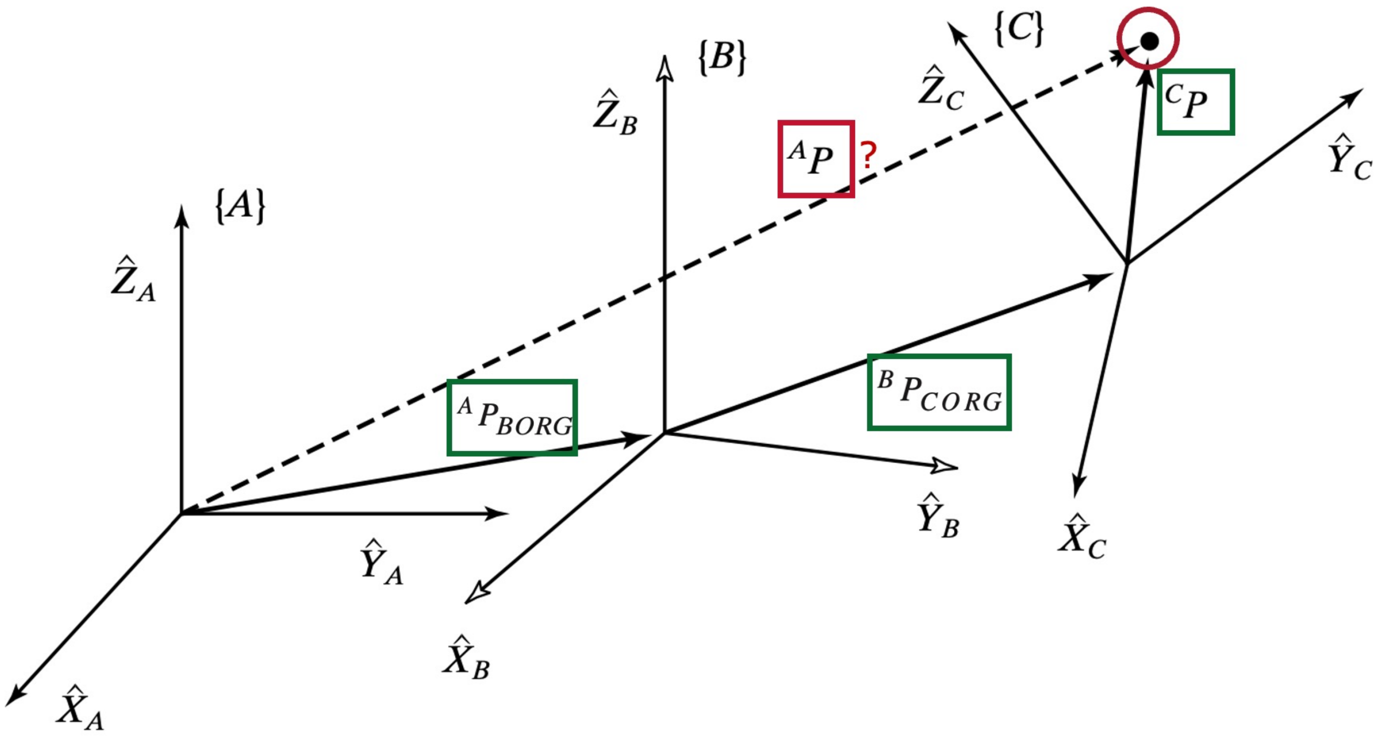
Compound Transformations



Compound Transformations

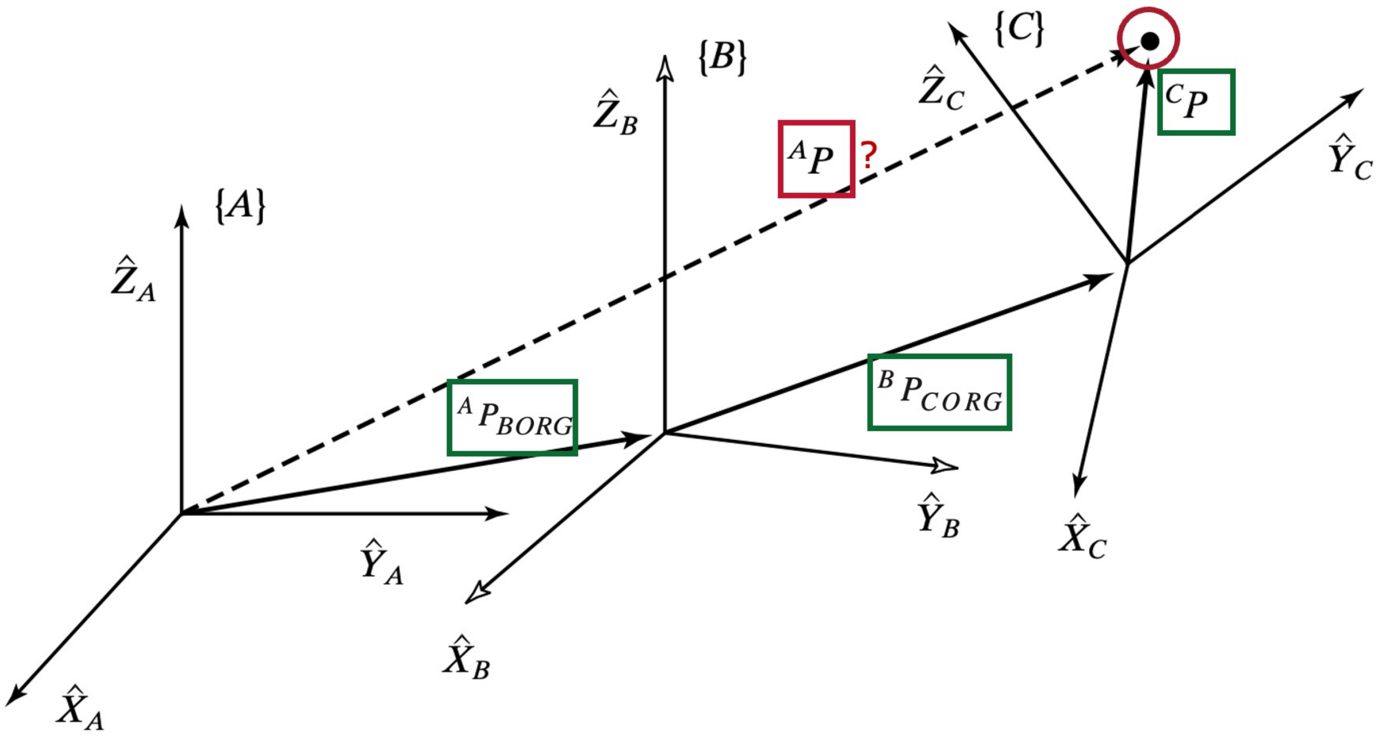
- ${}^{B}P = {}^{B}_{C}T\,{}^{C}P$
- ${}^{A}P = {}^{A}_{B}T\,{}^{B}P$

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



Compound Transformations

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



- Note: ${}^{A}P = {}^{A}_{B}T\,{}^{B}_{C}T\,{}^{C}P$

Compound Transformations

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

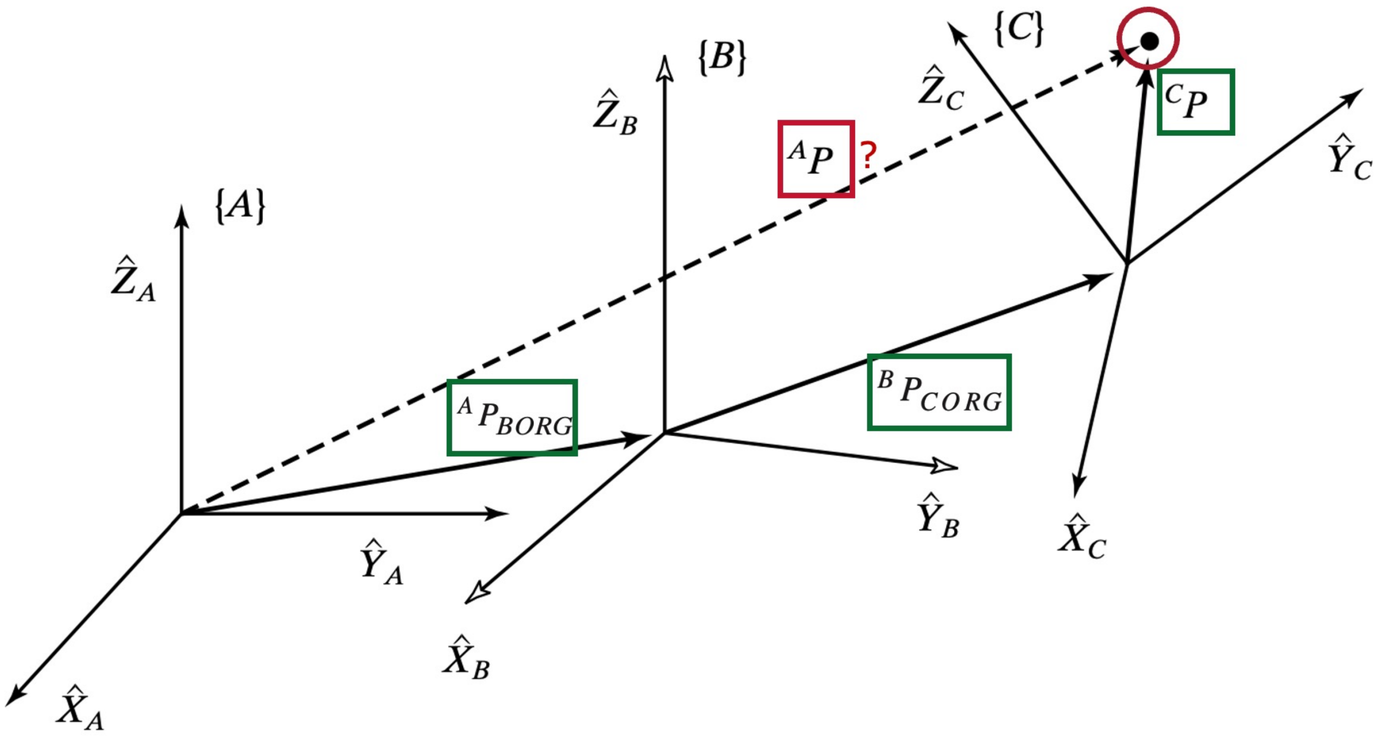
- ${}^{A}P = {}^{A}_{B}T\,{}^{B}_{C}T\,{}^{C}P$
- ${}^{A}_{C}T = {}^{A}_{B}T\,{}^{B}_{C}T$

- Note: ${}^{A}P = {}^{A}_{B}T\,{}^{B}_{C}T\,{}^{C}P$

- Reminder:

$${}^{A}_{B}T = \begin{bmatrix} {}^{A}_{B}R & {}^{A}P_{BORG} \\ 0_{1\times 3} & 1 \end{bmatrix}$$

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



Compound Transformations - Inverted transform

- What if we know ${}^{A}_{B}T$ but we need to find ${}^{B}_{A}T$?

Compound Transformations - Inverted transform

- What if we know ${}^{A}_{B}T$ but we need to find ${}^{B}_{A}T$?

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

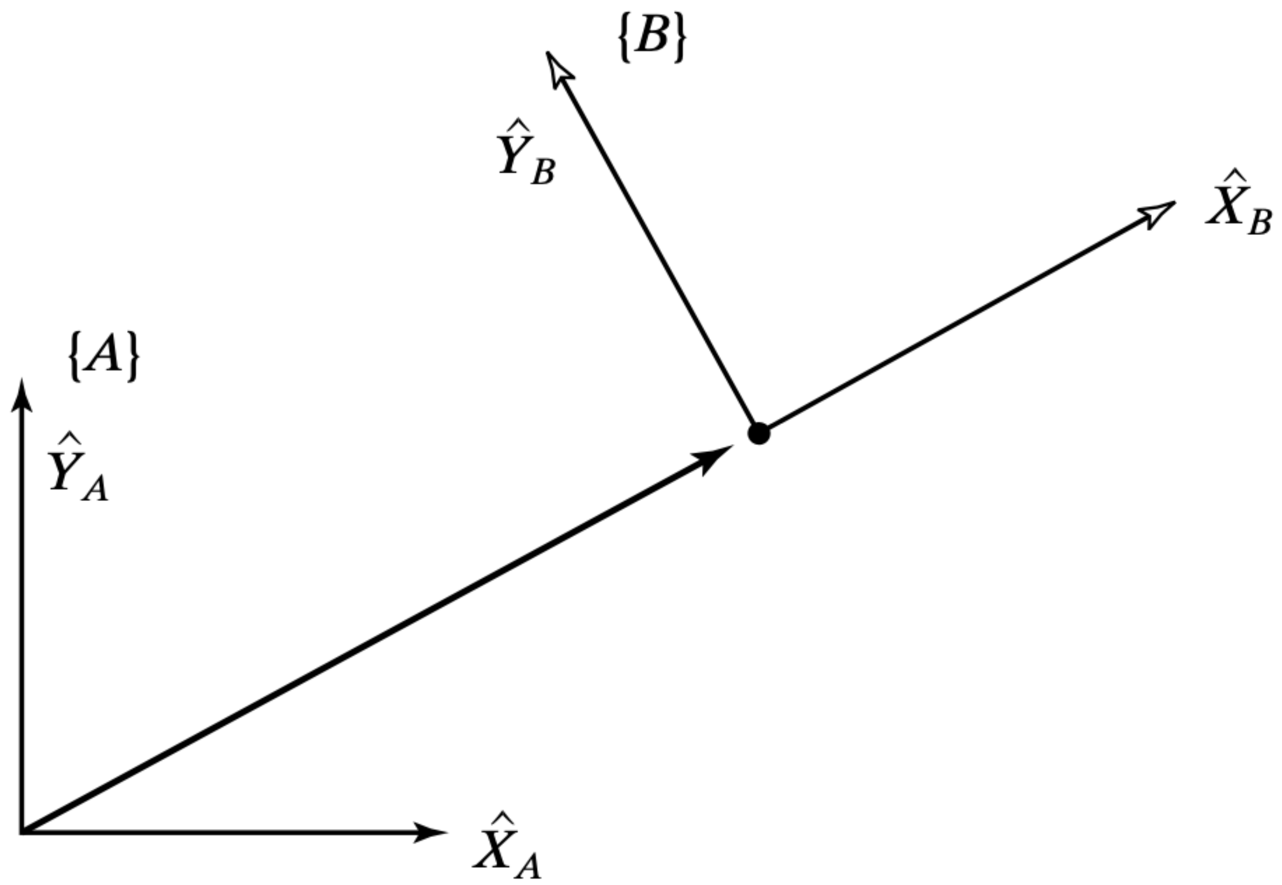
$$
\left(\begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline \mathbf{0}^{T} & 1 \end{array}\right)^{-1} = \left(\begin{array}{c|c} \mathbf{R}^{T} & -\mathbf{R}^{T}\mathbf{t} \\ \hline \mathbf{0}^{T} & 1 \end{array}\right)
$$

- Example 5

Compound Transformations - Inverted transform

- The figure shows a frame {B} that is rotated relative to frame {A} about $\widehat{Z_A}$ by 30 degrees and translated 4 units in $\widehat{X_A}$ and 3 units in $\widehat{Y_A}$.

- Find ${}^B_AT$



- Example 5
Compound Transformations - Inverted transform

- The figure shows a frame {B} that is rotated relative to frame {A} about $\widehat{Z_A}$ by 30 degrees and translated 4 units in $\widehat{X_A}$ and 3 units in $\widehat{Y_A}$.

- ```
  >> 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 = [4; 3; 0]

  A_P_BORG =

       4
       3
       0
  ```

- Example 5
Compound Transformations - Inverted transform

Solution in MATLAB

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

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

- ```
>> 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 = [4; 3; 0]

A_P_BORG =

     4
     3
     0
```

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

A_T_B =

    0.8660   -0.5000   0   4.0000
    0.5000   0.8660   0   3.0000
         0   0   1.0000   0
         0   0   0   1.0000
```
```

Compound Transformations - Inverted transform

Solution in MATLAB

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

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

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

- ```
>> 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 = [4; 3; 0]

A_P_BORG =

     4
     3
     0
```

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

A_T_B =

    0.8660   -0.5000   0   4.0000
    0.5000   0.8660   0   3.0000
         0   0   1.0000   0
         0   0   0   1.0000
```
```

- Example 5
Compound Transformations - Inverted transform

Solution in MATLAB

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

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

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

-
```
>> 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 = [4; 3; 0]

A_P_BORG =

     4
     3
     0
```

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

A_T_B =

    0.8660   -0.5000         0    4.0000
    0.5000    0.8660         0    3.0000
         0         0    1.0000         0
         0         0         0    1.0000
```

-
```
>> A_Rz30_B_Tr = transpose(A_Rz30_B)

A_Rz30_B_Tr =

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

- Example 5
Compound Transformations - Inverted transform

Solution in MATLAB

- $${}^A_BT = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 4.0 \\ 0.500 & 0.866 & 0.000 & 3.0 \\ 0.000 & 0.000 & 1.000 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^A_BT = \left[\begin{array}{c|c} {}^A_BR & {}^AP_{BORG} \\ \hline 0_{1\times 3} & 1 \end{array}\right]$$

- $${}^B_AT = \left[\begin{array}{ccc|c} & {}^A_BR^T & & -{}^A_BR^T\,{}^AP_{BORG} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

- $${}^B_AT = \begin{bmatrix} 0.866 & 0.500 & 0.000 & -4.964 \\ -0.500 & 0.866 & 0.000 & -0.598 \\ 0.000 & 0.000 & 1.000 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
>> 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 = [4; 3; 0]

A_P_BORG =

     4
     3
     0
```

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

A_T_B =

    0.8660   -0.5000         0    4.0000
    0.5000    0.8660         0    3.0000
         0         0    1.0000         0
         0         0         0    1.0000
```

```
>> A_Rz30_B_Tr = transpose(A_Rz30_B)

A_Rz30_B_Tr =

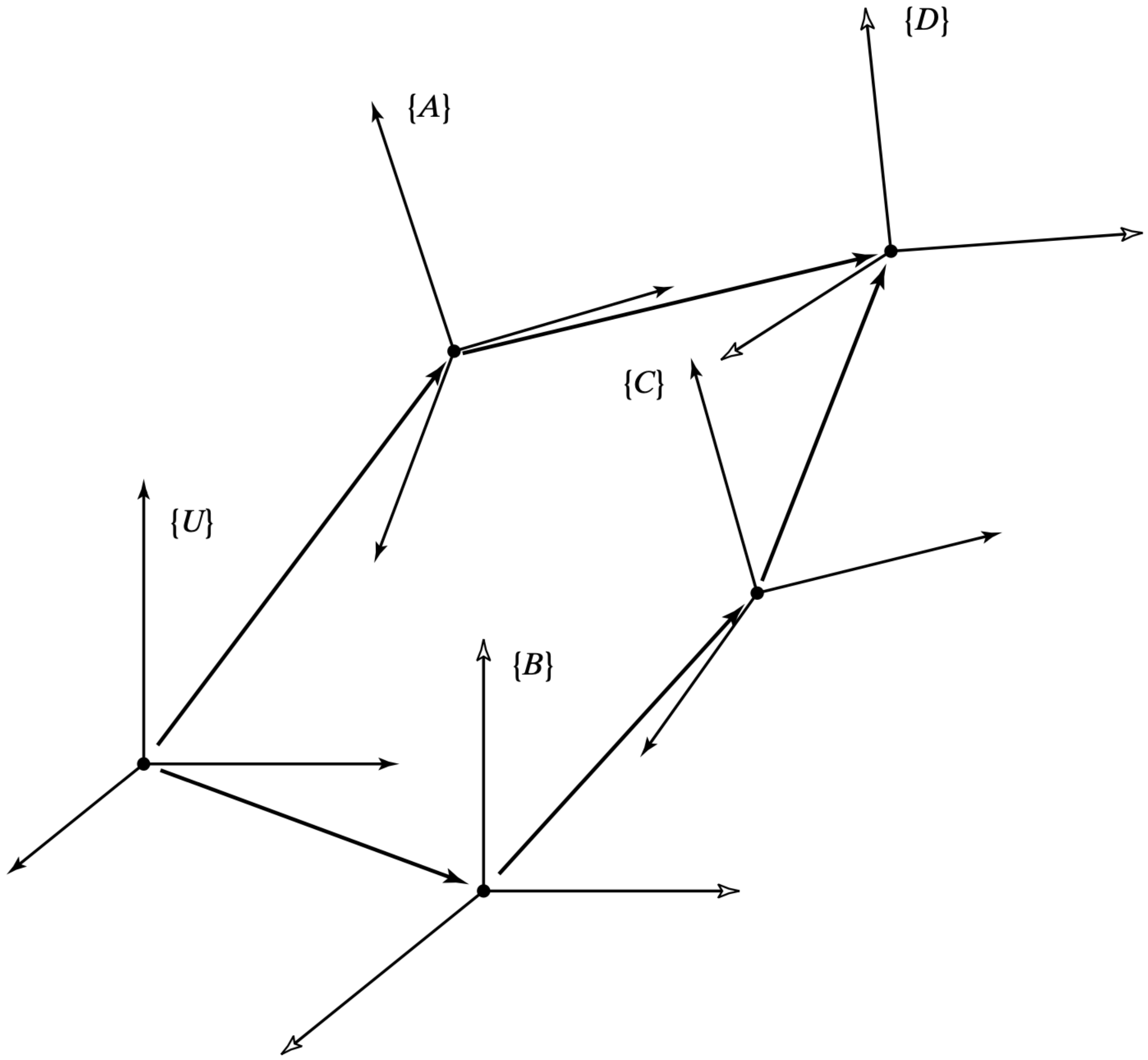
    0.8660    0.5000         0
   -0.5000    0.8660         0
         0         0    1.0000
```

```
>> B_T_A = [A_Rz30_B_Tr, -A_Rz30_B_Tr * A_P_BORG; 0, 0, 0, 1]

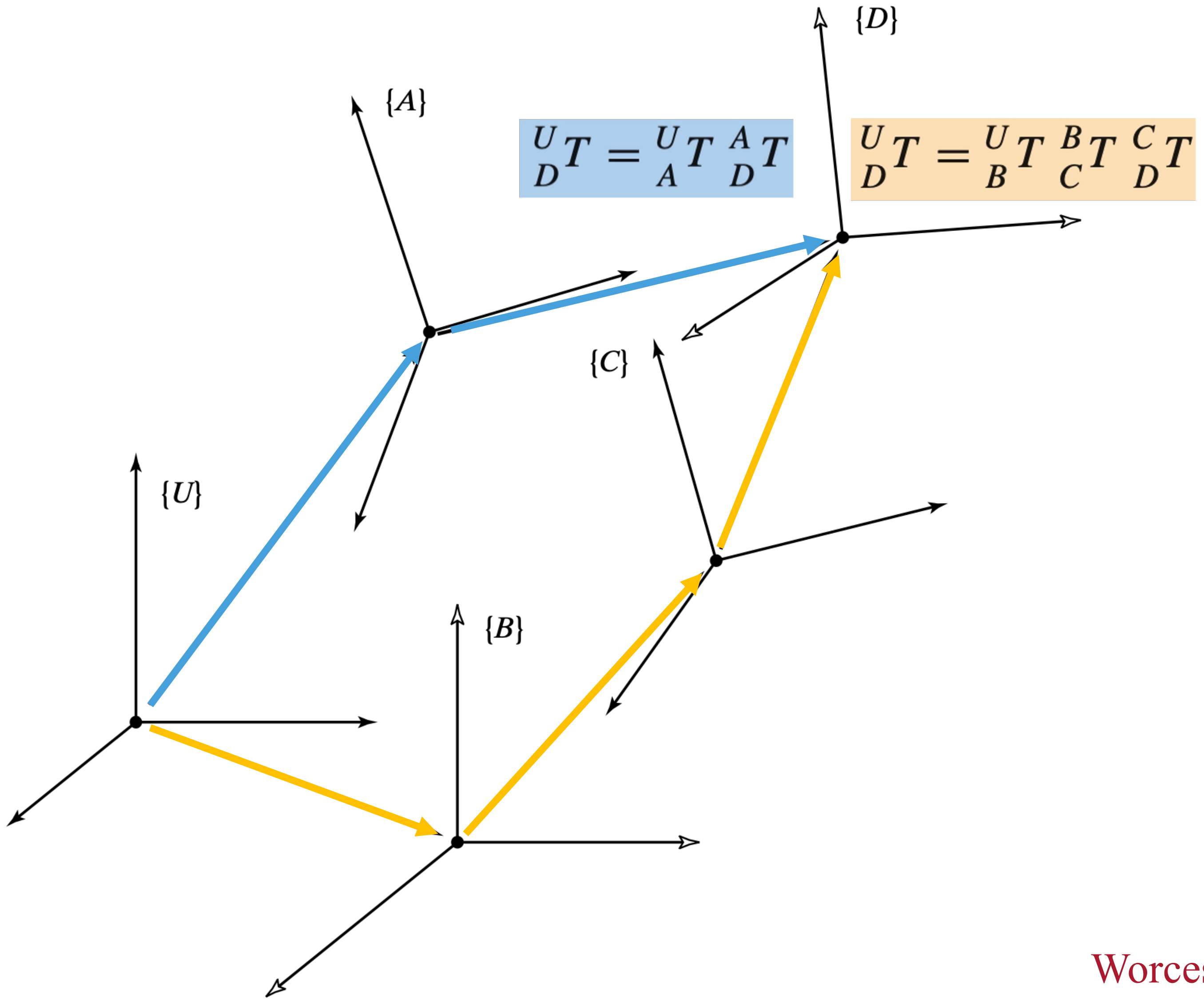
B_T_A =

    0.8660    0.5000         0   -4.9641
   -0.5000    0.8660         0   -0.5981
         0         0    1.0000         0
         0         0         0    1.0000
```

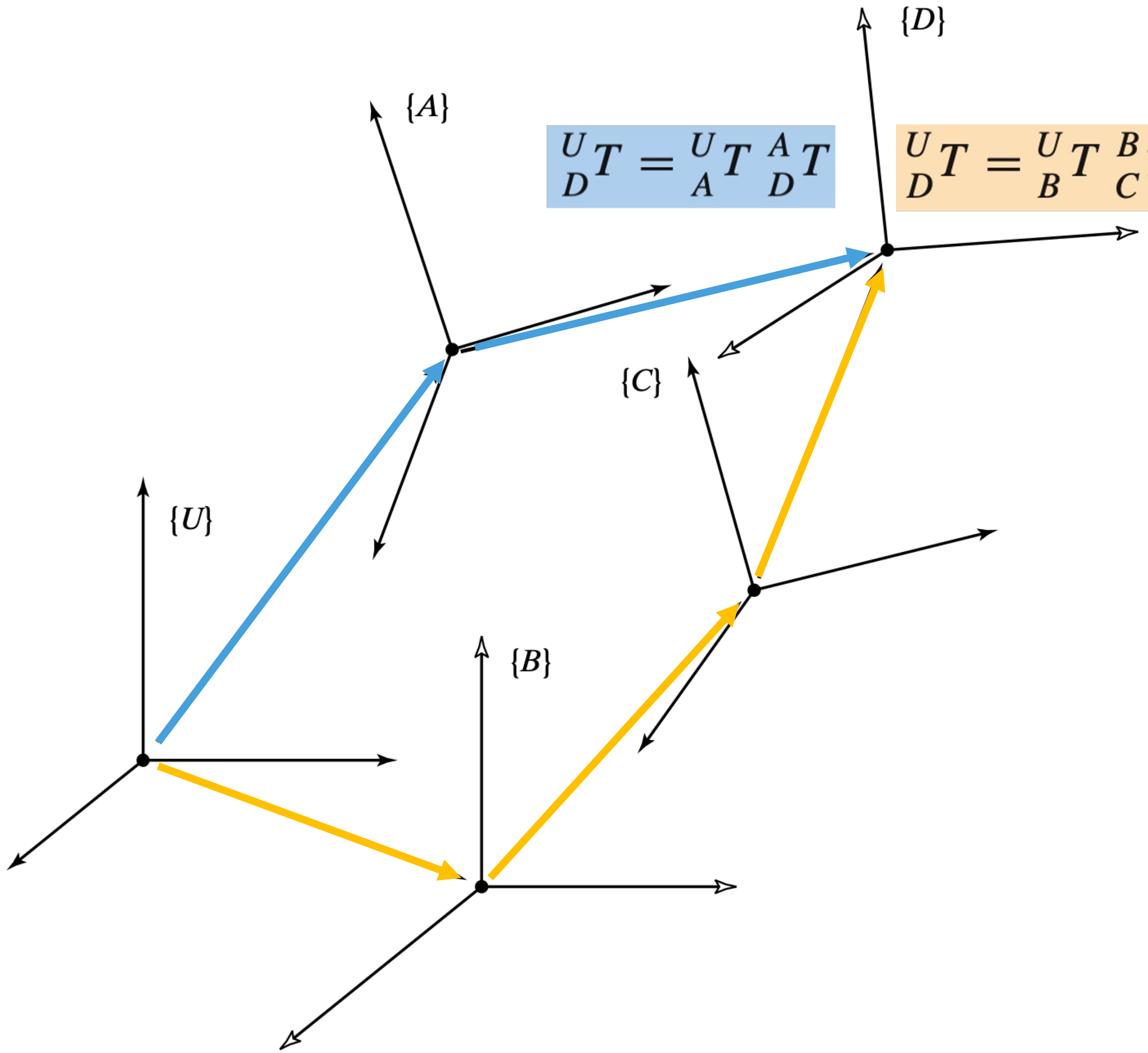
Transform Equations



Transform Equations



Transform Equations

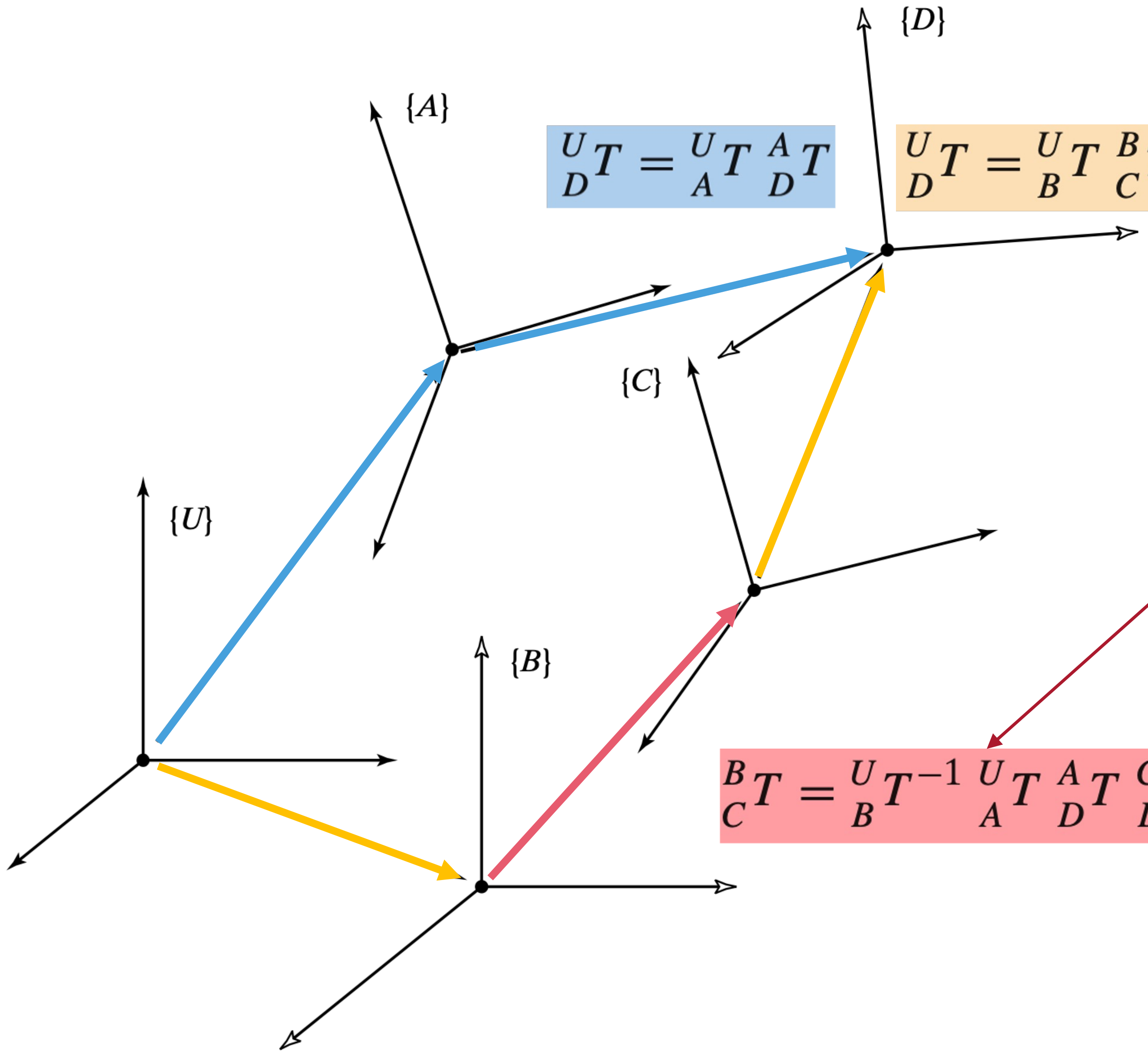


$${}^{U}_{D}T = {}^{U}_{A}T\,{}^{A}_{D}T$$

$${}^{U}_{D}T = {}^{U}_{B}T\,{}^{B}_{C}T\,{}^{C}_{D}T$$

$$\Rightarrow\quad {}^{U}_{A}T\,{}^{A}_{D}T = {}^{U}_{B}T\,{}^{B}_{C}T\,{}^{C}_{D}T$$

Transform Equations



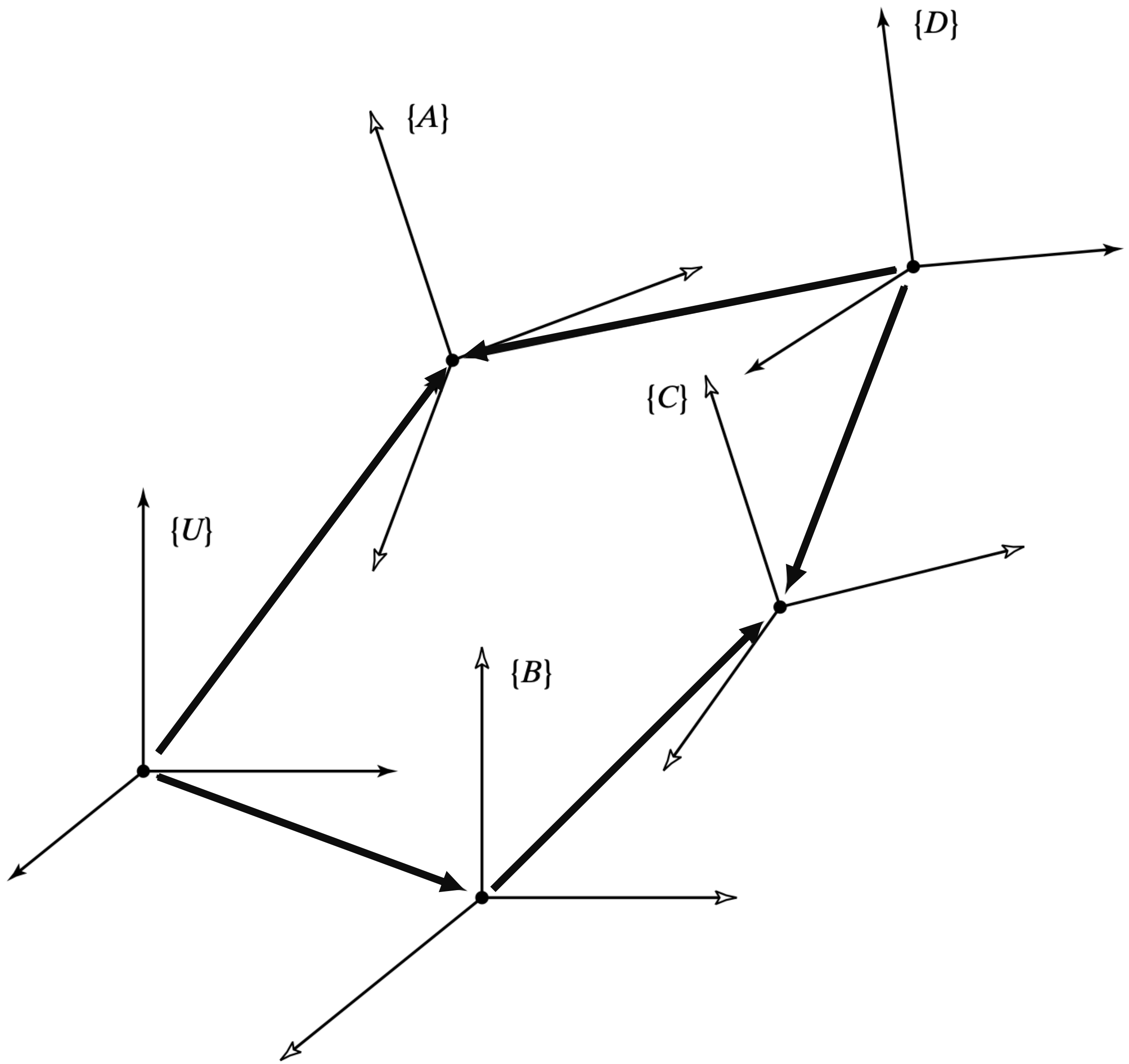
$${}^{U}_{D}T = {}^{U}_{A}T\,{}^{A}_{D}T$$

$${}^{U}_{D}T = {}^{U}_{B}T\,{}^{B}_{C}T\,{}^{C}_{D}T$$

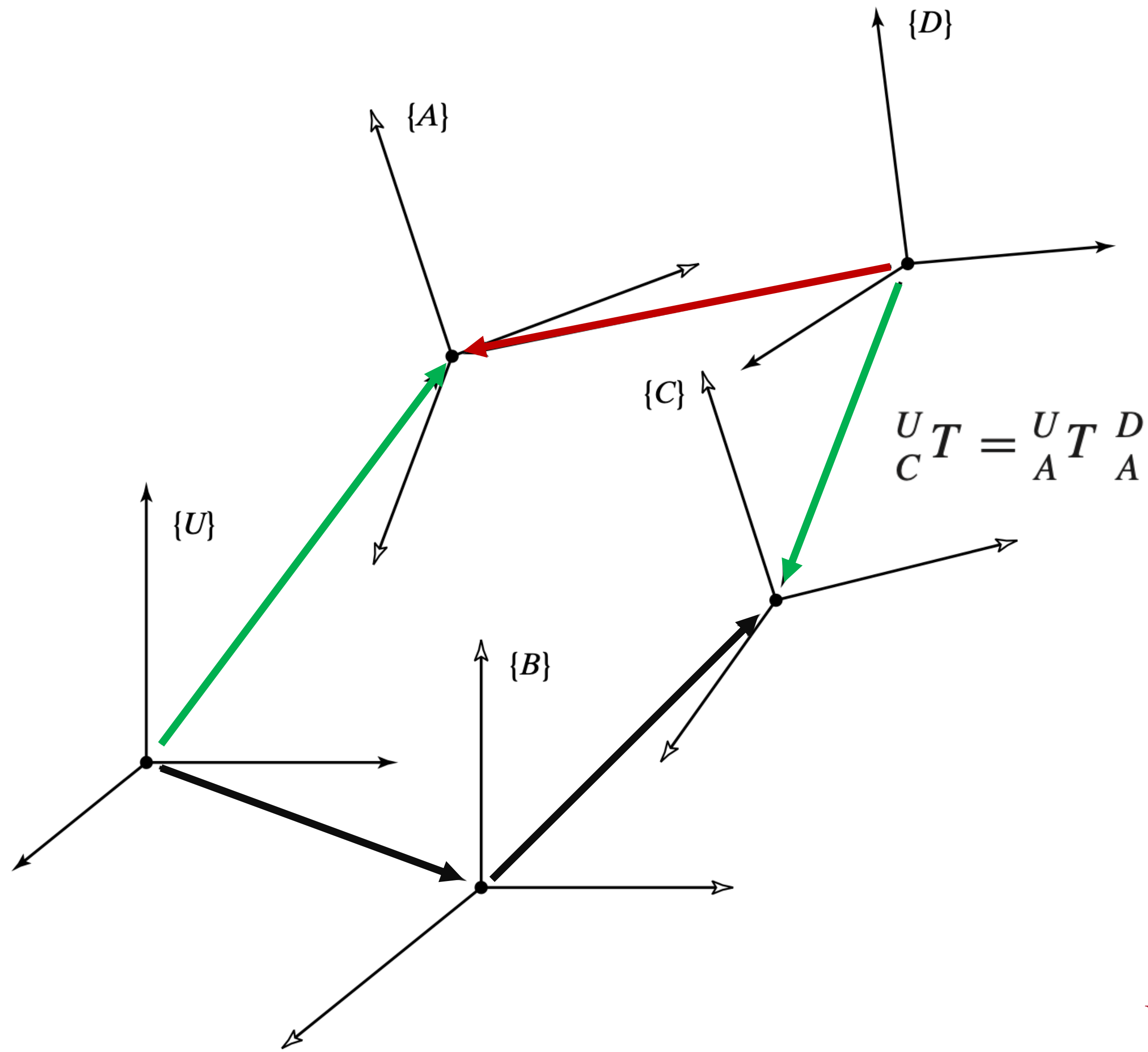
$$\Rightarrow \quad {}^{U}_{A}T\,{}^{A}_{D}T = {}^{U}_{B}T\,{}^{B}_{C}T\,{}^{C}_{D}T$$

$${}^{B}_{C}T = {}^{U}_{B}T^{-1}\,{}^{U}_{A}T\,{}^{A}_{D}T\,{}^{C}_{D}T^{-1}$$

Transform Equations

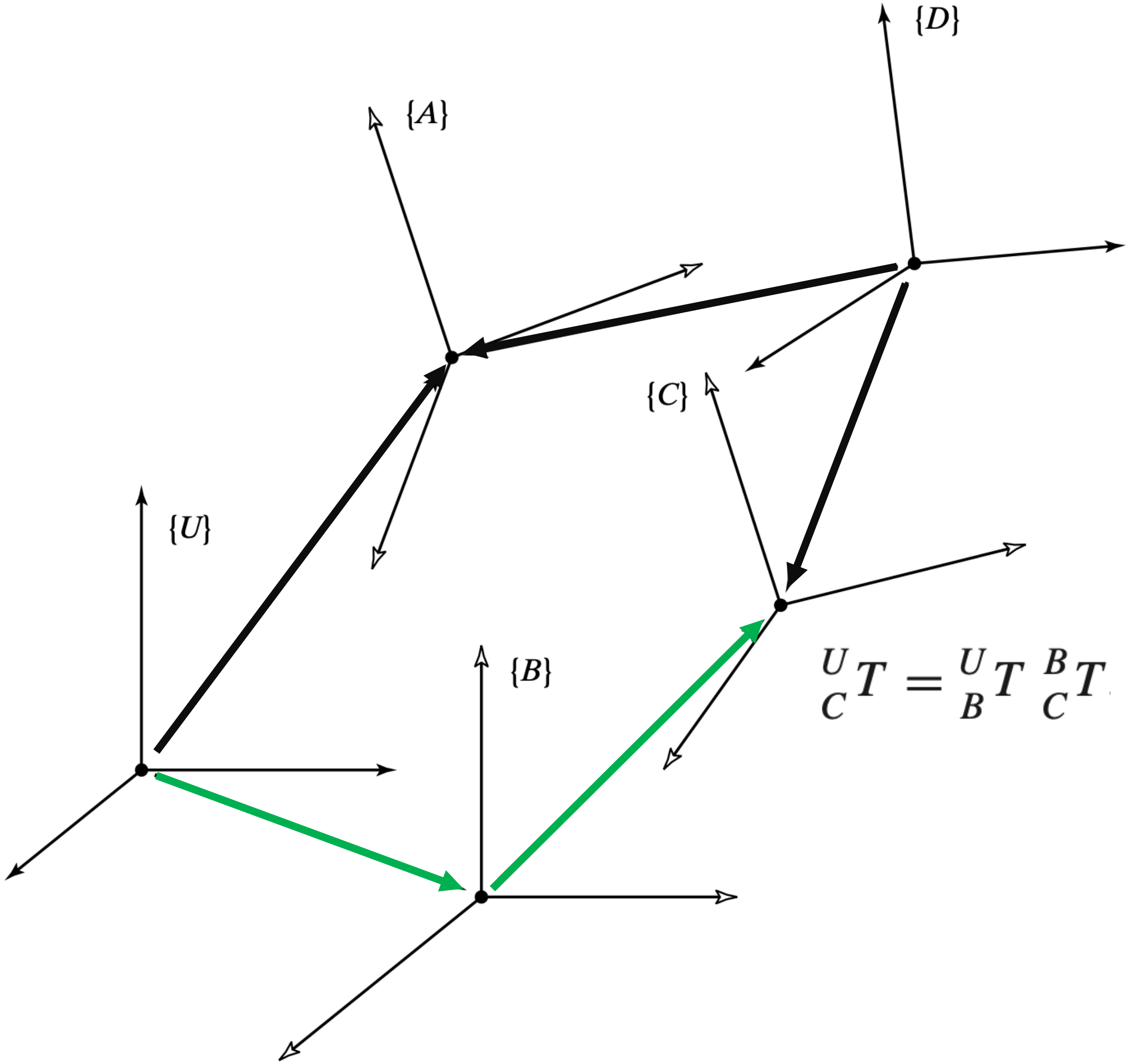


Transform Equations



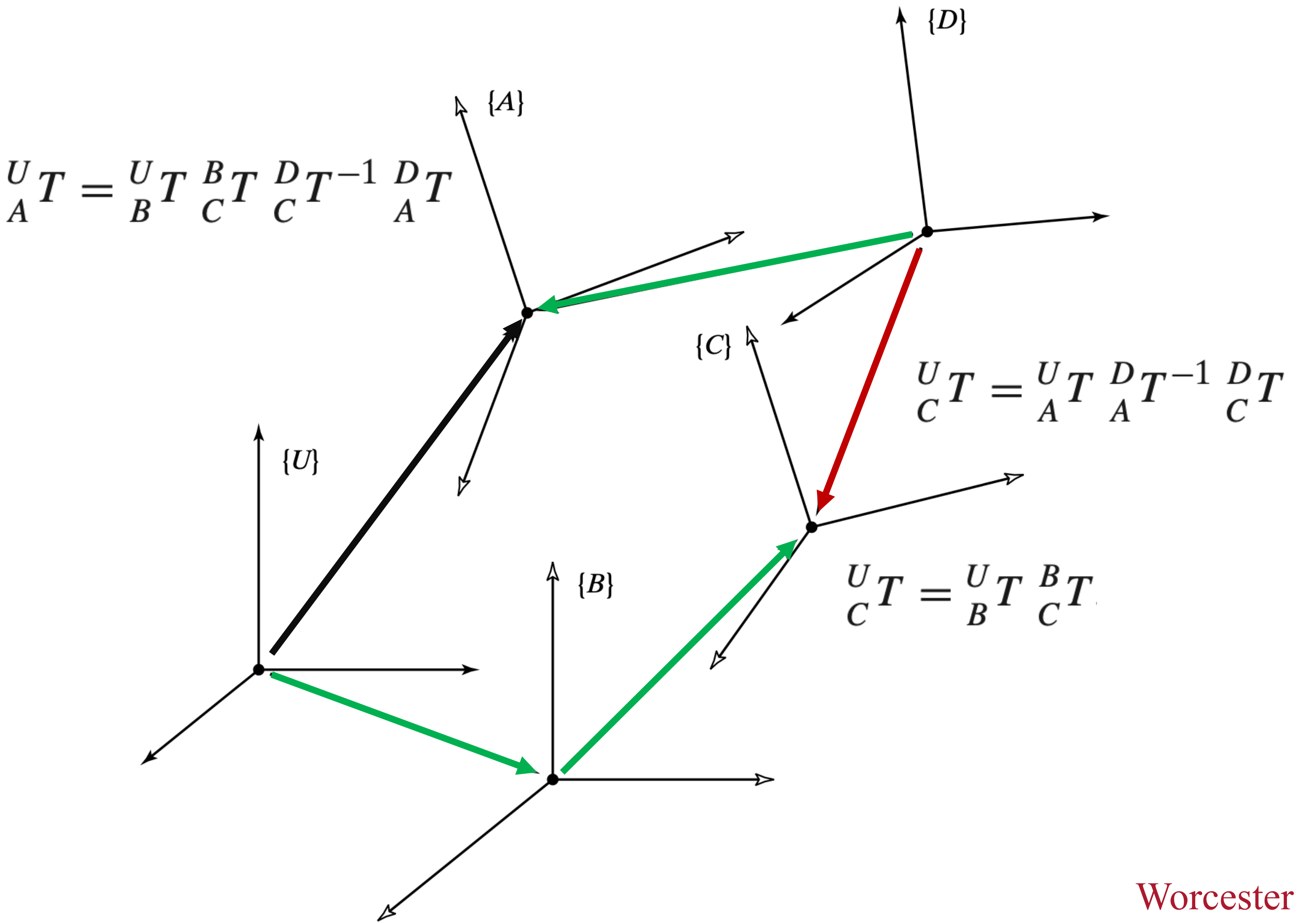
$$ {}^{U}_{C}T = {}^{U}_{A}T \; {}^{D}_{A}T^{-1} \; {}^{D}_{C}T $$

Transform Equations

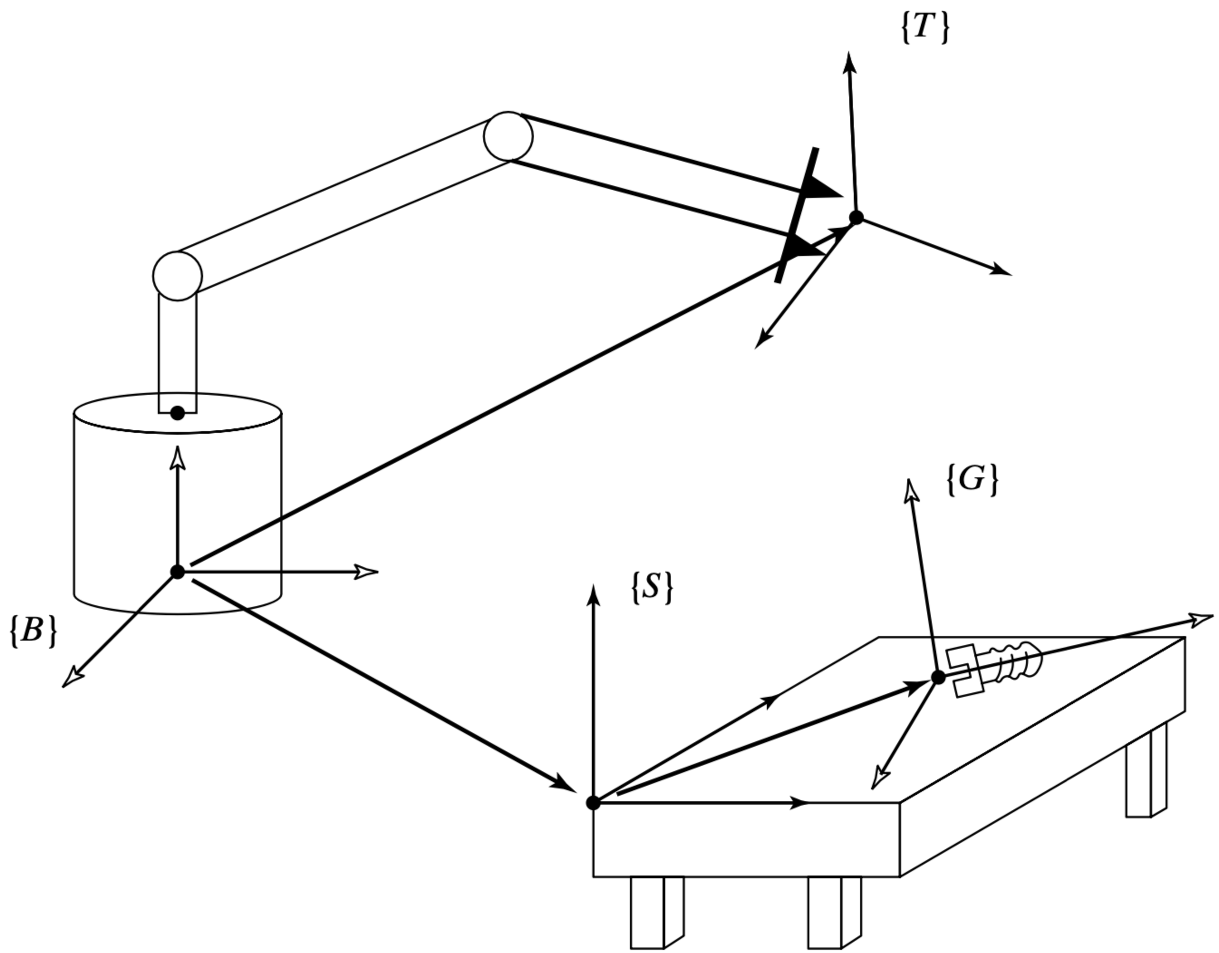


$$ {}^{U}_{C}T = {}^{U}_{B}T \; {}^{B}_{C}T $$

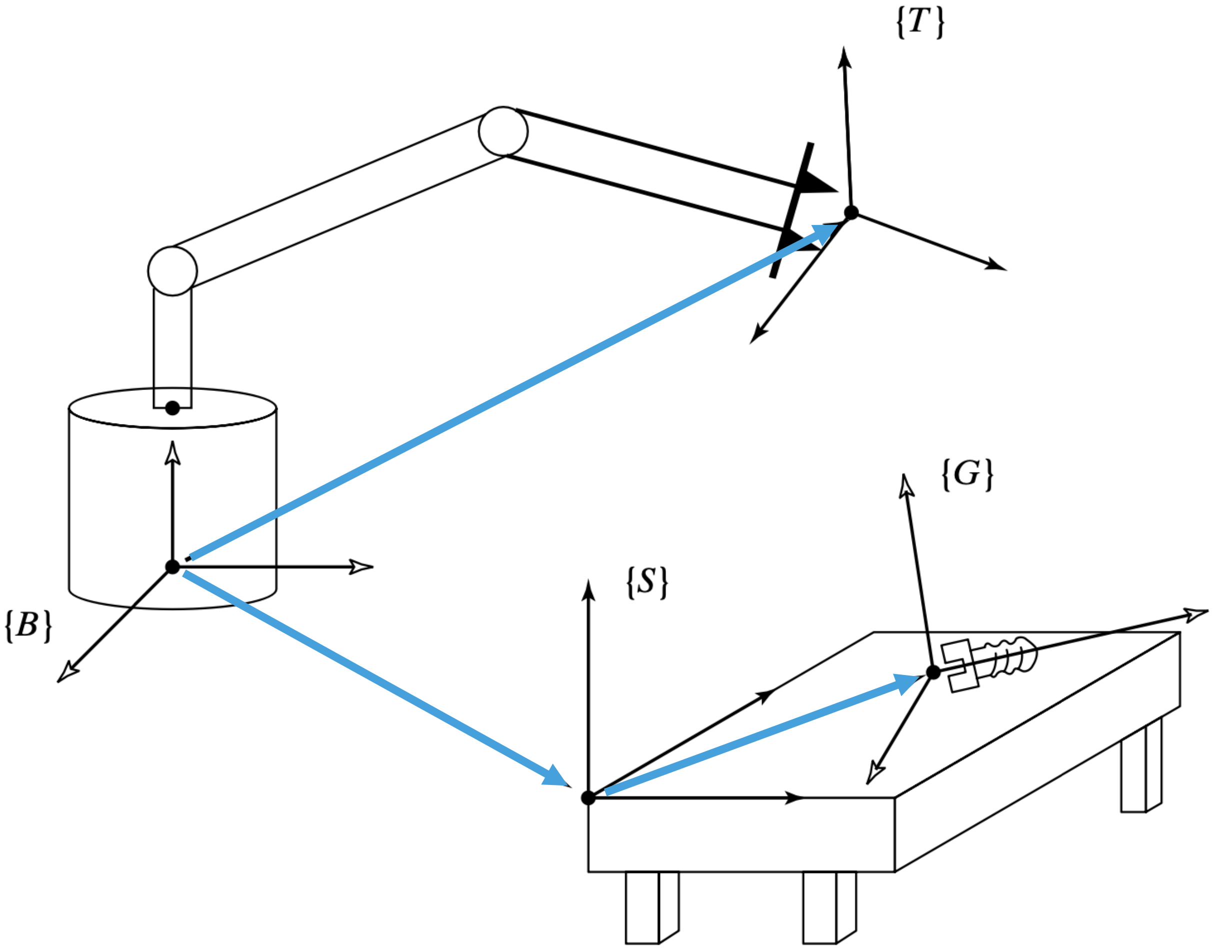
Transform Equations



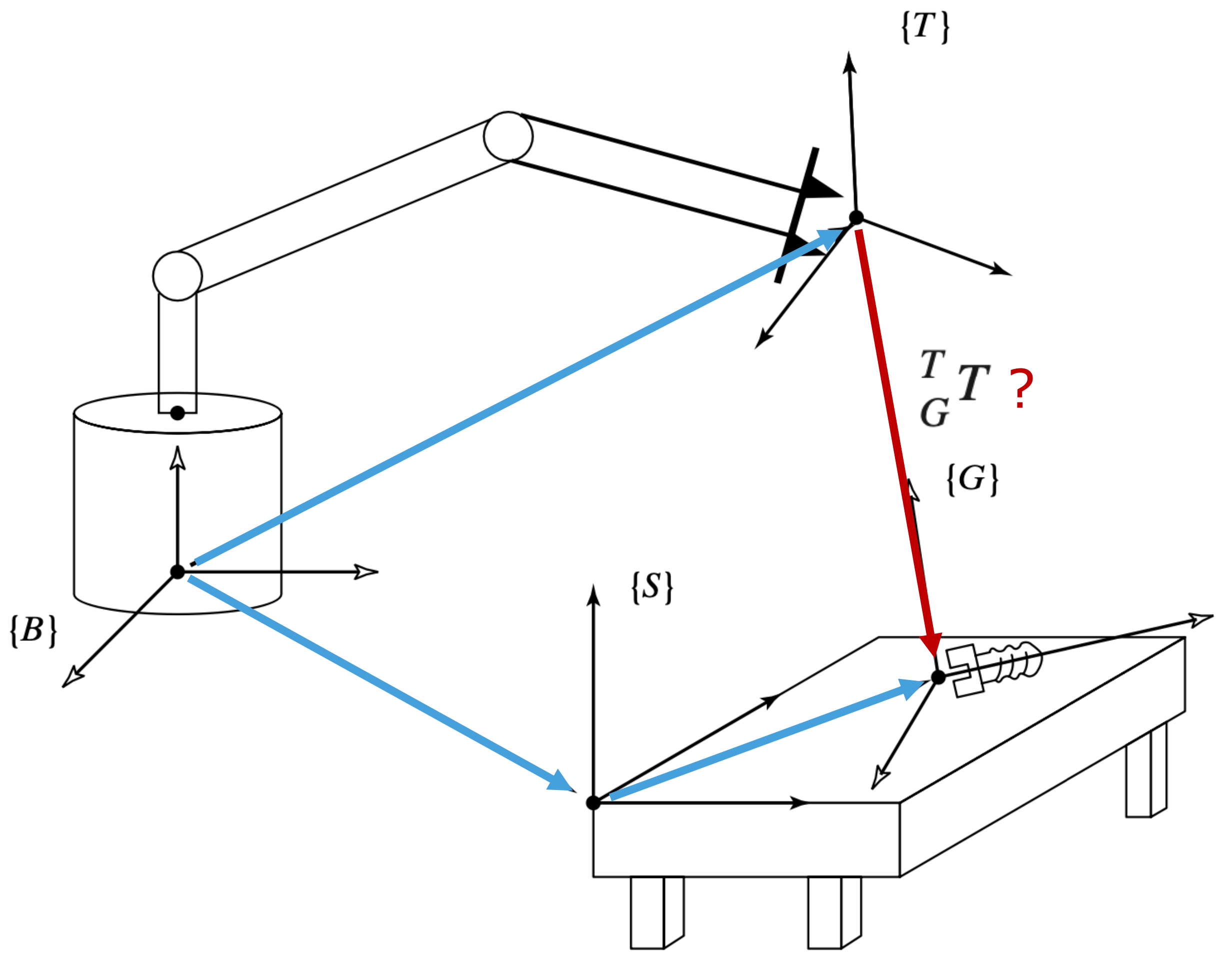
Transform Equations - Example 6



Transform Equations - Example 6

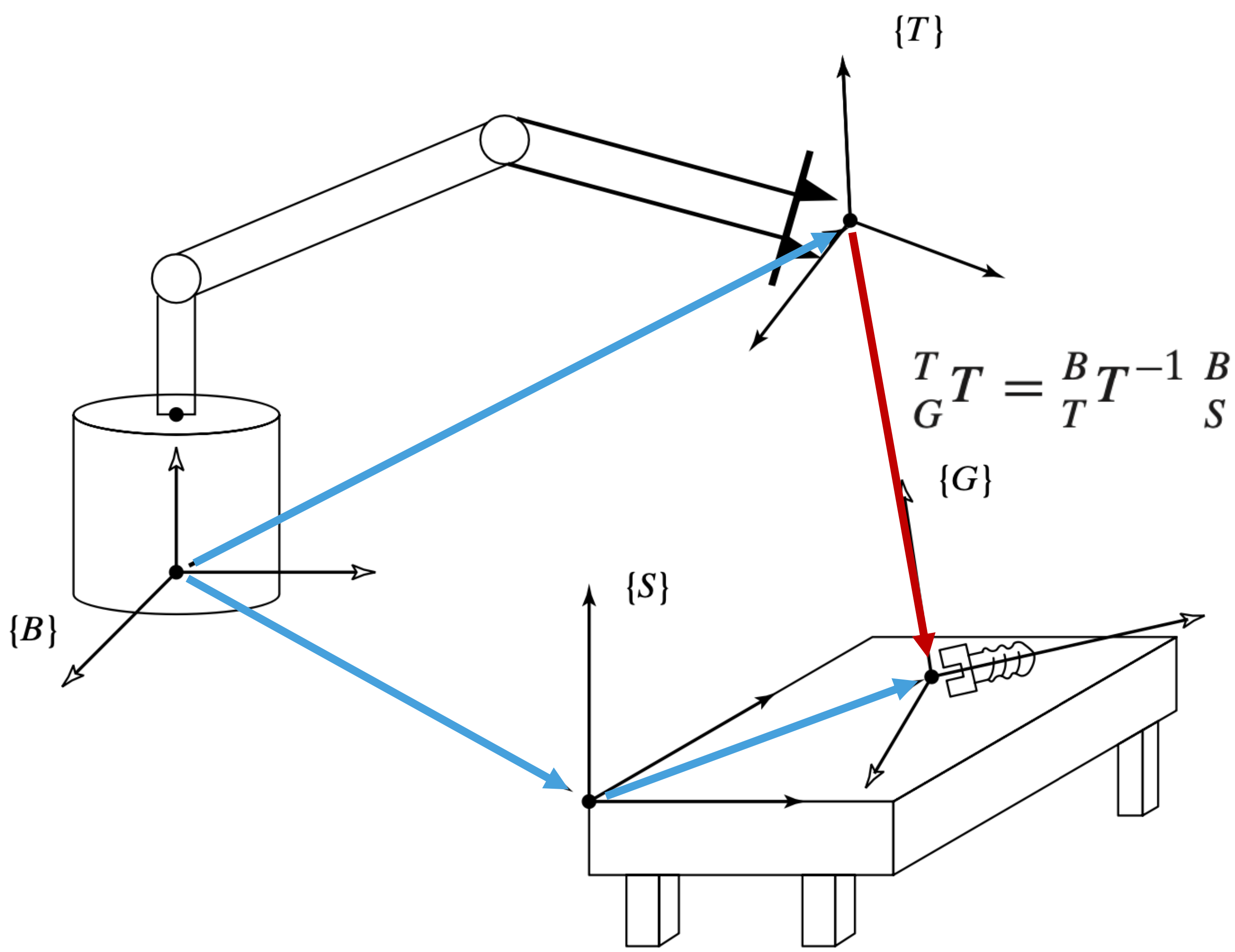


Transform Equations - Example 6



How do we represent the transformation of the bolt relative to the manipulator's hand, ${}^{T}_{G}T$?

Transform Equations - Example 6



$$ {}^{T}_{G}T = {}^{B}_{T}T^{-1}\,{}^{B}_{S}T\,{}^{S}_{G}T $$

Homogeneous Transformation (HT) matrix

- The Homogeneous Transformation (HT) matrices belong to the Special Euclidean group SE(3): [4x4]

$$H = \left[\begin{array}{c|c} R_{3\times3} & d_{3\times1} \\ \hline f_{1\times3} & s_{1\times1} \end{array}\right] = \left[\begin{array}{c|c} Rotation & Translation \\ \hline perspective & scale\ factor \end{array}\right]$$

- 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:

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

Properties of HT matrix

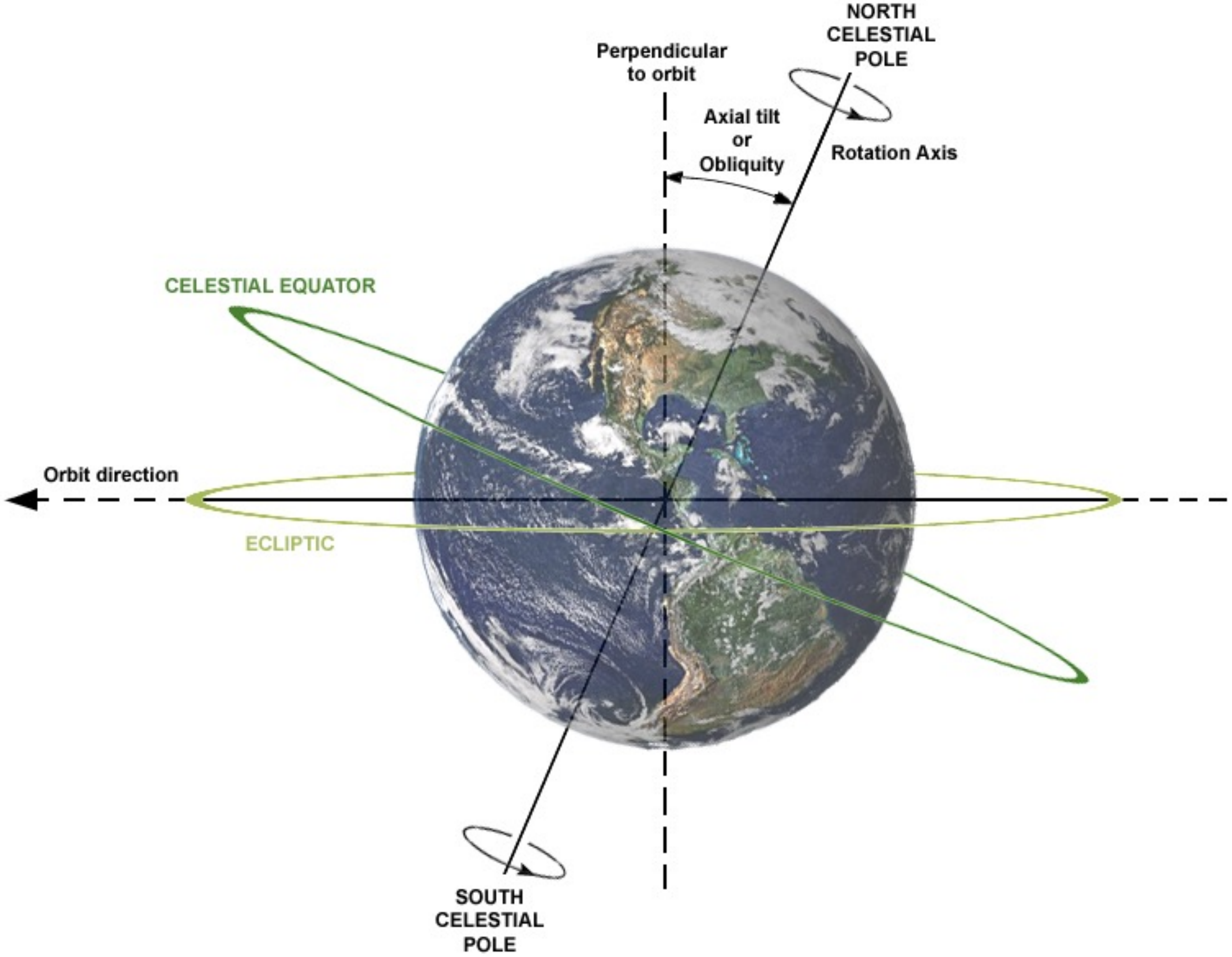
- The product of two HT matrices is a HT matrix: ${}^{A}_{C}T = {}^{A}_{B}T\,{}^{B}_{C}T.$
- The inverse of a HT is equal to:

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

- <u>In our course</u> we use the concept of **Transformation with respect to the current frame (post-multiplication)**

$$
{}^{0}_{3}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T
$$

Rotation and Orientation



Properties of Rotation Matrix

- An orientation is represented by 3 × 3 rotation matrix.
- The inverse of a rotation is equal to the transpose of it.

$$R^{\mathrm{T}} = R^{-1}$$

- The columns are mutually orthogonal and have unit magnitude.

$$R = [\hat{X}\ \hat{Y}\ \hat{Z}].$$

- The determinant of a rotation matrix is always equal to +1. [det(R) = 1;]

- Rotation matrices may also be called **proper orthonormal matrices**,
  - where "proper" refers to the fact that the determinant is +1
  - nonproper orthonormal matrices have the determinant −1

Properties of Rotation Matrix in MATLAB - Part 1

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

```
>> R = rotx(90)

R =

     1     0     0
     0     0    -1
     0     1     0
```

- $R^{\mathrm{T}} = R^{-1}$

```
>> R_tr = transpose(R)

R_tr =

     1     0     0
     0     0     1
     0    -1     0
```

```
>> R_inv = inv(R)

R_inv =

     1     0     0
     0     0     1
     0    -1     0
```

- $\det R = 1$

```
>> det(R)

ans =

     1
```

- `>> R * R_inv`

```
ans =

     1     0     0
     0     1     0
     0     0     1
```

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

Properties of Rotation Matrix in MATLAB - Part 2

- $R = [\hat{X}\ \hat{Y}\ \hat{Z}]$

- ```
  >> R = rotx(90)

  R =

       1     0     0
       0     0    -1
       0     1     0
  ```

- ```
  >> X_hat = R(:,1)

  X_hat =

       1
       0
       0
  ```
  ```
  >> Y_hat = R(:,2)

  Y_hat =

       0
       0
       1
  ```
  ```
  >> Z_hat = R(:,3)

  Z_hat =

       0
      -1
       0
  ```

```
>> R = [X_hat, Y_hat, Z_hat]

R =

     1     0     0
     0     0    -1
     0     1     0
```

Properties of Rotation Matrix in MATLAB - Part 3

- $R = [\hat{X}\ \hat{Y}\ \hat{Z}]$

- ```
  >> R = rotx(90)

  R =

       1     0     0
       0     0    -1
       0     1     0
  ```

- The columns have unit magnitude:

$$\left.\begin{aligned} |\hat{X}| &= 1, \\ |\hat{Y}| &= 1, \\ |\hat{Z}| &= 1, \end{aligned}\right\}$$

```
>> norm(X_hat)

ans =

     1

>> norm(Y_hat)

ans =

     1

>> norm(Z_hat)

ans =

     1
```

- *MATLAB norm function: n = norm(v) returns the Euclidean norm of vector v. This norm is also called the 2-norm, vector magnitude, or Euclidean length.*
```

Properties of Rotation Matrix in MATLAB - Part 4

- $R = [\hat{X}\ \hat{Y}\ \hat{Z}]$

- The columns are mutually orthogonal:

$$\left.\begin{aligned}\hat{X}\cdot\hat{Y}&=0,\\ \hat{X}\cdot\hat{Z}&=0,\\ \hat{Y}\cdot\hat{Z}&=0.\end{aligned}\right\}$$

```
>> dot(X_hat,Y_hat)

ans =

     0

>> dot(X_hat,Z_hat)

ans =

     0

>> dot(Y_hat,Z_hat)

ans =

     0
```

-
```
>> R = rotx(90)

R =

     1     0     0
     0     0    -1
     0     1     0
```

Representation of Orientation

- We can describe an orientation with fewer than nine numbers. How?
  - Linear algebra: for any proper orthonormal matrix R, there exists a skew-symmetric matrix S such that

$$R = (I_3 - S)^{-1}(I_3 + S)$$

  - A skew-symmetric matrix $S = -S^T$ of dimension 3 is specified by three parameters (s_x, s_y, s_z) as:

$$S = \begin{bmatrix} 0 & -s_x & s_y \\ s_x & 0 & -s_x \\ -s_y & s_x & 0 \end{bmatrix}$$

  - Thus, any 3 × 3 rotation matrix can be specified by just three parameters.

Representation of Orientation

- Translations are quite easy to visualize, whereas rotations seem less intuitive
- Hard to describe and specify the orientations in 3D space
- Rotations don't commute

$${}^{A}_{B}R\,{}^{B}_{C}R \quad \text{not the same} \quad {}^{B}_{C}R\,{}^{A}_{B}R$$

Representation of Orientation - Example 1

- Consider two rotations, one about Z-axis by 30 degrees and one about X-axis by 30 degrees.
- $Rot_z(30) \cdot Rot_x(30)$?
- $Rot_x(30) \cdot Rot_z(30)$?
- What do you observe ?

Representation of Orientation - Example 1

- Consider two rotations, one about Z-axis by 30 degrees and one about X-axis by 30 degrees.
- $Rot_z(30) \cdot Rot_x(30)$
- $Rot_x(30) \cdot Rot_z(30)$

$$R_z(30) = \begin{bmatrix} 0.866 & -0.500 & 0.000 \\ 0.500 & 0.866 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}$$

$$R_x(30) = \begin{bmatrix} 1.000 & 0.000 & 0.000 \\ 0.000 & 0.866 & -0.500 \\ 0.000 & 0.500 & 0.866 \end{bmatrix}$$

$$\boxed{\begin{aligned} R_z(30)R_x(30) &= \begin{bmatrix} 0.87 & -0.43 & 0.25 \\ 0.50 & 0.75 & -0.43 \\ 0.00 & 0.50 & 0.87 \end{bmatrix} \\ \neq R_x(30)R_z(30) &= \begin{bmatrix} 0.87 & -0.50 & 0.00 \\ 0.43 & 0.75 & -0.50 \\ 0.25 & 0.43 & 0.87 \end{bmatrix} \end{aligned}}$$

Representation of Orientation - Example 1

MATLAB code

```
>> rotz(30)

ans =

    0.8660   -0.5000         0
    0.5000    0.8660         0
         0         0    1.0000

>> rotx(30)

ans =

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

$$R_z(30) = \begin{bmatrix} 0.866 & -0.500 & 0.000 \\ 0.500 & 0.866 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}$$

$$R_x(30) = \begin{bmatrix} 1.000 & 0.000 & 0.000 \\ 0.000 & 0.866 & -0.500 \\ 0.000 & 0.500 & 0.866 \end{bmatrix}$$

```
>> rotz(30) * rotx(30)

ans =

    0.8660   -0.4330    0.2500
    0.5000    0.7500   -0.4330
         0    0.5000    0.8660

>> rotx(30) * rotz(30)

ans =

    0.8660   -0.5000         0
    0.4330    0.7500   -0.5000
    0.2500    0.4330    0.8660
```

$$R_z(30)R_x(30) = \begin{bmatrix} 0.87 & -0.43 & 0.25 \\ 0.50 & 0.75 & -0.43 \\ 0.00 & 0.50 & 0.87 \end{bmatrix}$$

$$\neq R_x(30)R_z(30) = \begin{bmatrix} 0.87 & -0.50 & 0.00 \\ 0.43 & 0.75 & -0.50 \\ 0.25 & 0.43 & 0.87 \end{bmatrix}$$

Successive Rotations

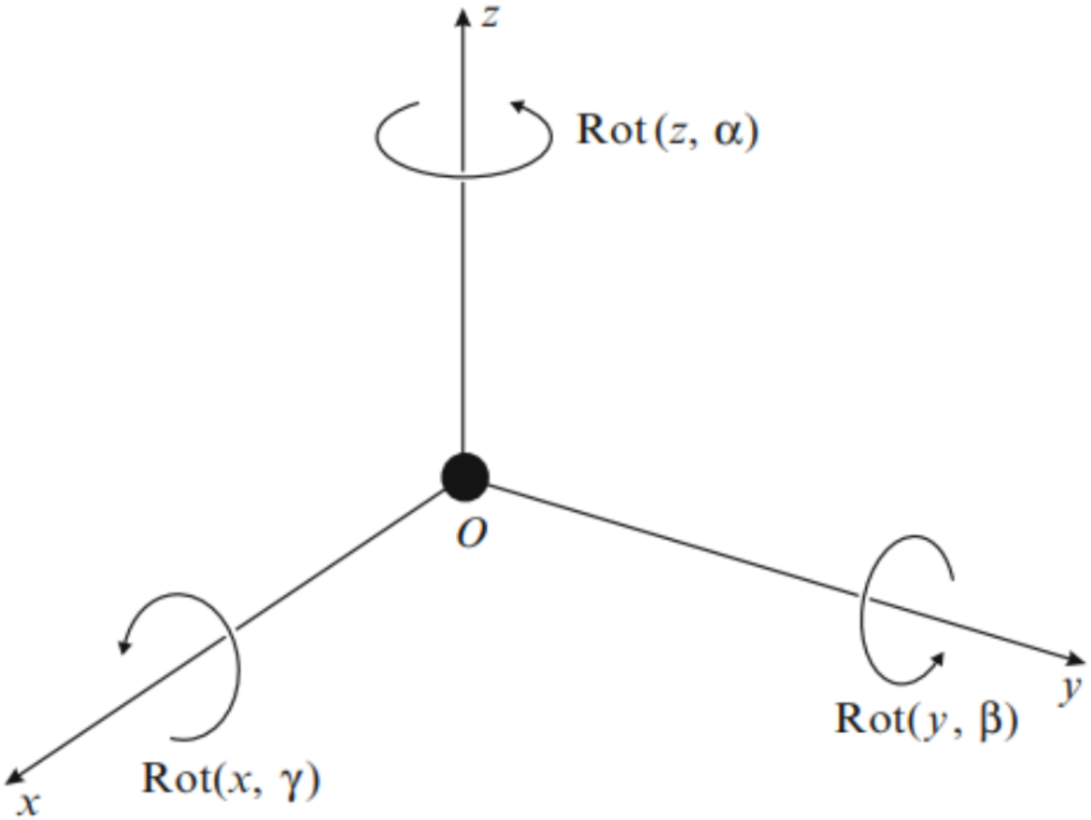
- Approaches:
  - Rotations around a fixed reference frame (**fixed**)
    - <u>Pre-multiply the rotation matrices</u>
  - Rotations around a moving reference frame (**current**)
    - <u>Post-multiply the rotation matrices</u>
- **Any 3D rotation can be decomposed into successive combination of basic rotations one way or another.**

Fundamental Rotations

- $$R_Z(\alpha) = \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- $$R_Y(\beta) = \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix}$$

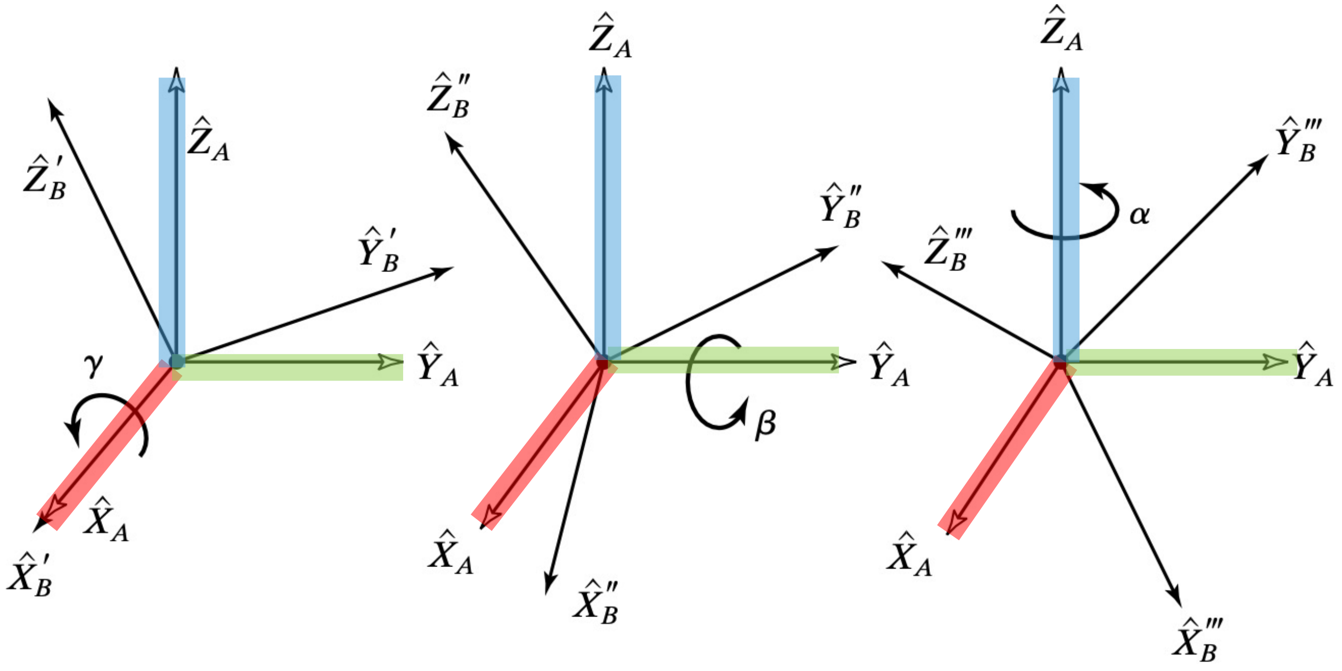
- $$R_X(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix}$$

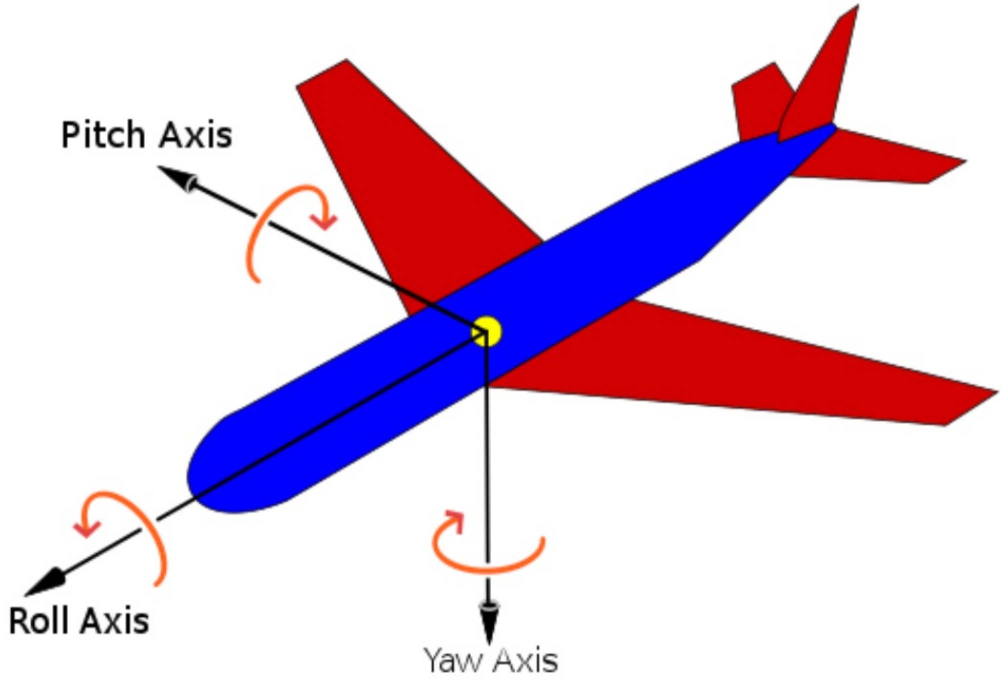


https://en.wikipedia.org/wiki/Rotation_matrix#Basic_3D_rotations

X-Y-Z fixed angles (Roll, pitch, Yaw)

- Start with the frame coincident with a known reference frame $\{A\}$. Rotate $\{B\}$ first about $\hat{X}_A$ by an angle γ, then about $\hat{Y}_A$ by an angle β, and, finally, about $\hat{Z}_A$ by an angle α.





X-Y-Z fixed angles

- $${}^{A}_{B}R_{XYZ}(\gamma,\beta,\alpha) = R_Z(\alpha)R_Y(\beta)R_X(\gamma)$$
  $$= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix}$$

- **Apply pre-multiplications of rotations (from the *right to left*) of $R_X(\gamma)$, then $R_Y(\beta)$, and then $R_Z(\alpha)$.**

- $${}^{A}_{B}R_{XYZ}(\gamma,\beta,\alpha) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}$$

X-Y-Z fixed angles - Analysis in MATLAB

```
>> syms a b g
>> R_Z_a = [cos(a) -sin(a) 0; sin(a) cos(a) 0; 0 0 1]

R_Z_a =

[cos(a), -sin(a), 0]
[sin(a),  cos(a), 0]
[     0,       0, 1]

>> R_Y_b = [cos(b) 0 sin(b); 0 1 0; -sin(b) 0 cos(b)]

R_Y_b =

[ cos(b), 0, sin(b)]
[      0, 1,      0]
[-sin(b), 0, cos(b)]

>> R_X_g = [1 0 0; 0 cos(g) -sin(g); 0 sin(g) cos(g)]

R_X_g =

[1,      0,       0]
[0, cos(g), -sin(g)]
[0, sin(g),  cos(g)]

>> temp = R_Y_b * R_X_g

temp =

[ cos(b), sin(b)*sin(g), cos(g)*sin(b)]
[      0,        cos(g),       -sin(g)]
[-sin(b), cos(b)*sin(g), cos(b)*cos(g)]

>> R_Z_a * temp

ans =

[cos(a)*cos(b), cos(a)*sin(b)*sin(g) - cos(g)*sin(a), sin(a)*sin(g) + cos(a)*cos(g)*sin(b)]
[cos(b)*sin(a), cos(a)*cos(g) + sin(a)*sin(b)*sin(g), cos(g)*sin(a)*sin(b) - cos(a)*sin(g)]
[      -sin(b),                        cos(b)*sin(g),                        cos(b)*cos(g)]
```

$${}^{A}_{B}R_{XYZ}(\gamma,\beta,\alpha) = R_Z(\alpha)R_Y(\beta)R_X(\gamma)$$

$$= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix}$$

$${}^{A}_{B}R_{XYZ}(\gamma,\beta,\alpha) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}$$

X-Y-Z fixed angles - Example 2

- *For α = 30°, β = 30° and γ = 30°.*

  What is the ${}^{A}_{B}R_{XYZ}(\gamma, \beta, \alpha)$ *?*

$$
{}^{A}_{B}R_{XYZ}(\gamma, \beta, \alpha) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}
$$

```
[cos(a)*cos(b), cos(a)*sin(b)*sin(g) - cos(g)*sin(a), sin(a)*sin(g) + cos(a)*cos(g)*sin(b)]
[cos(b)*sin(a), cos(a)*cos(g) + sin(a)*sin(b)*sin(g), cos(g)*sin(a)*sin(b) - cos(a)*sin(g)]
[      -sin(b),                        cos(b)*sin(g),                        cos(b)*cos(g)]
```

X-Y-Z fixed angles - Example 2

Solution and MATLAB code

```
>> a = deg2rad(30)

a =

    0.5236

>> b = deg2rad(30)

b =

    0.5236

>> g = deg2rad(30)

g =

    0.5236
```

```
R_Z_a =

    0.8660   -0.5000         0
    0.5000    0.8660         0
         0         0    1.0000

>> R_Y_b = [cos(b) 0 sin(b); 0 1 0; -sin(b) 0 cos(b)]

R_Y_b =

    0.8660         0    0.5000
         0    1.0000         0
   -0.5000         0    0.8660

>> R_X_g = [1 0 0; 0 cos(g) -sin(g); 0 sin(g) cos(g)]

R_X_g =

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

```
>> temp = R_Y_b * R_X_g

temp =

    0.8660    0.2500    0.4330
         0    0.8660   -0.5000
   -0.5000    0.4330    0.7500

>> A_R_XYZ_fixed_B = R_Z_a * temp

A_R_XYZ_fixed_B =

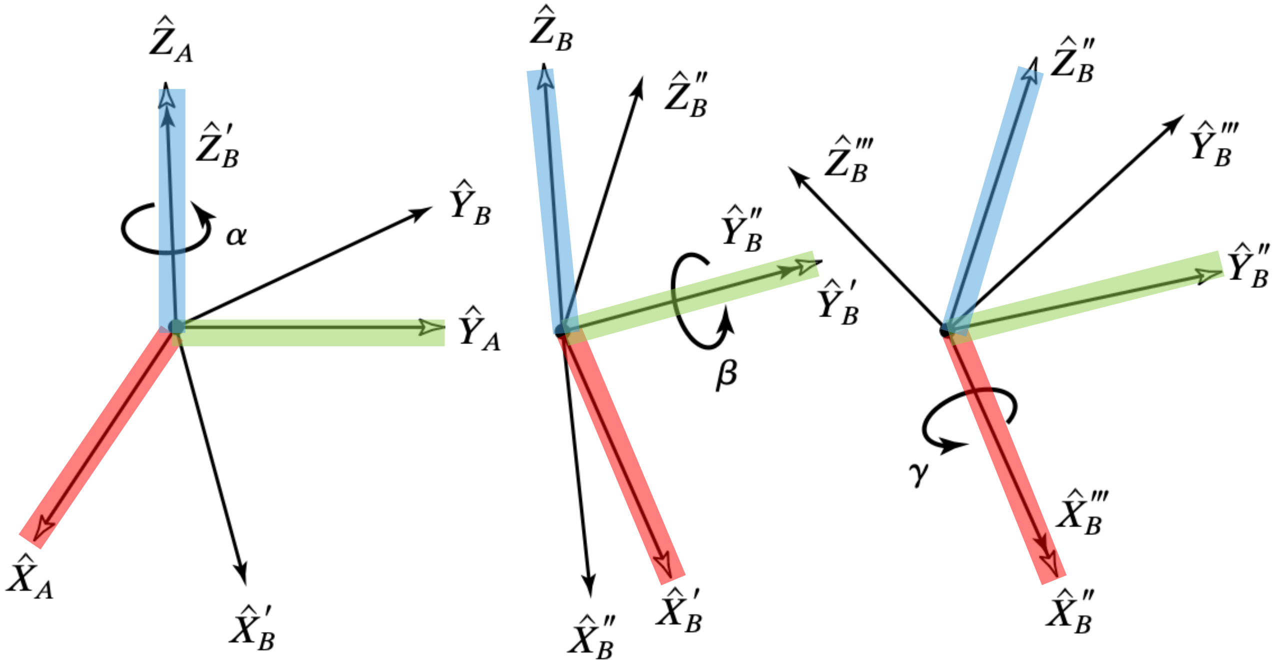
    0.7500   -0.2165    0.6250
    0.4330    0.8750   -0.2165
   -0.5000    0.4330    0.7500
```

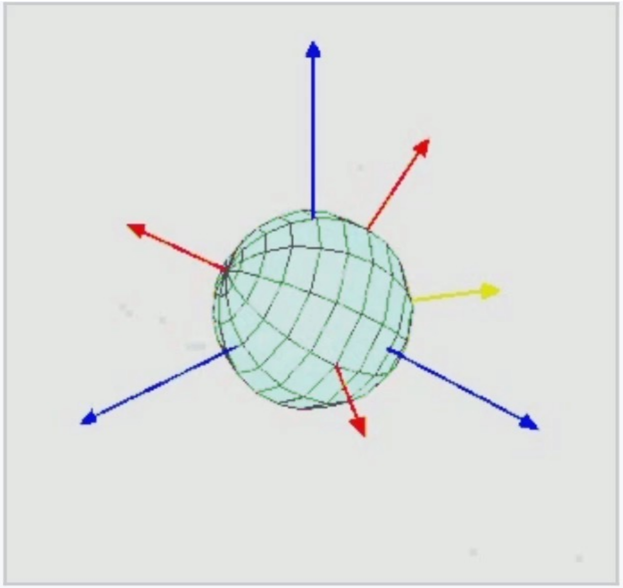
X-Y-Z fixed angles - Find α, β, γ

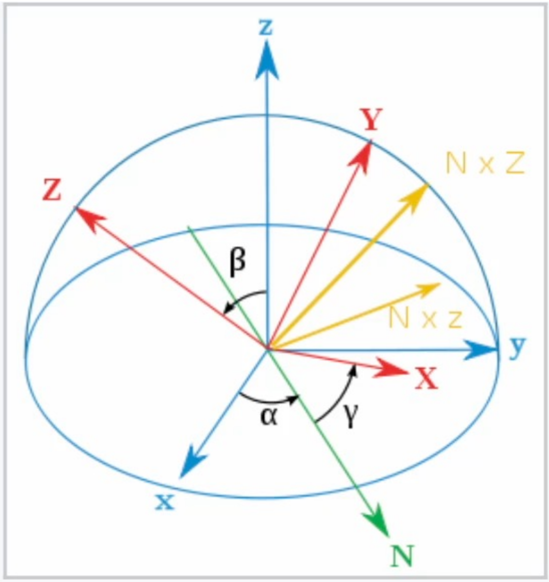
- $${}^{A}_{B}R_{XYZ}(\gamma,\beta,\alpha) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}$$
  - Solve a set of transcendental equations
    - Nine equations and three unknowns.
      - among the nine equations are six dependencies
    - We have **three equations and three unknowns**.
- $$\beta = \mathrm{Atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}),$$
- $$\alpha = \mathrm{Atan2}(r_{21}/c\beta, r_{11}/c\beta),$$
- $$\gamma = \mathrm{Atan2}(r_{32}/c\beta, r_{33}/c\beta),$$

Z-Y-X Euler angles

- Start with the frame coincident with a known frame $\{A\}$. Rotate $\{B\}$ first about $\hat{Z}_B$ by an angle α, then about $\hat{Y}_B$ by an angle β, and, finally, about $\hat{X}_B$ by an angle γ.







Z-Y-X Euler angles

- $^A_B R = {}^A_{B'}R \; {}^{B'}_{B''}R \; {}^{B''}_{B}R$
- $^A P = {}^A_B T \; {}^B P$

- $$^A_B R_{Z'Y'X'} = R_Z(\alpha) R_Y(\beta) R_X(\gamma)$$
$$= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix}$$

- **Apply post-multiplications of rotations (from the *<u>left to right</u>*) of $R_Z(\alpha)$, then $R_Y(\beta)$, and then $R_X(\gamma)$.**

- $$^A_B R_{Z'Y'X'}(\alpha, \beta, \gamma) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}$$

Z-Y-X Euler angles - Analysis in MATLAB

```
>> syms a b g
>> R_Z_a = [cos(a) -sin(a) 0; sin(a) cos(a) 0; 0 0 1]

R_Z_a =

[cos(a), -sin(a), 0]
[sin(a),  cos(a), 0]
[     0,       0, 1]

>> R_Y_b = [cos(b) 0 sin(b); 0 1 0; -sin(b) 0 cos(b)]

R_Y_b =

[ cos(b), 0, sin(b)]
[      0, 1,      0]
[-sin(b), 0, cos(b)]

>> R_X_g = [1 0 0; 0 cos(g) -sin(g); 0 sin(g) cos(g)]

R_X_g =

[1,      0,       0]
[0, cos(g), -sin(g)]
[0, sin(g),  cos(g)]

>> R_Z_a * R_Y_b * R_X_g

ans =

[cos(a)*cos(b), cos(a)*sin(b)*sin(g) - cos(g)*sin(a), sin(a)*sin(g) + cos(a)*cos(g)*sin(b)]
[cos(b)*sin(a), cos(a)*cos(g) + sin(a)*sin(b)*sin(g), cos(g)*sin(a)*sin(b) - cos(a)*sin(g)]
[      -sin(b),                        cos(b)*sin(g),                        cos(b)*cos(g)]
```

$${}^{A}_{B}R_{Z'Y'X'} = R_Z(\alpha)R_Y(\beta)R_X(\gamma)$$

$$= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix}$$

$${}^{A}_{B}R_{Z'Y'X'}(\alpha,\beta,\gamma) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}$$

Z-Y-X Euler angles - Example 3

- *For α = 30°, β = 30° and γ = 30°.*

  What is the ${}^{A}_{B}R_{Z'Y'X'}$ *?*

$$
{}^{A}_{B}R_{Z'Y'X'}(\alpha,\beta,\gamma) = \boxed{\begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}}
$$

```
[cos(a)*cos(b), cos(a)*sin(b)*sin(g) - cos(g)*sin(a), sin(a)*sin(g) + cos(a)*cos(g)*sin(b)]
[cos(b)*sin(a), cos(a)*cos(g) + sin(a)*sin(b)*sin(g), cos(g)*sin(a)*sin(b) - cos(a)*sin(g)]
[      -sin(b),                        cos(b)*sin(g),                        cos(b)*cos(g)]
```

Z-Y-X Euler angles - Example 3

Solution in MATLAB

```
>> a = deg2rad(30)

a =

    0.5236

>> b = deg2rad(30)

b =

    0.5236

>> g = deg2rad(30)

g =

    0.5236
```

```
R_Z_a =

    0.8660   -0.5000         0
    0.5000    0.8660         0
         0         0    1.0000

>> R_Y_b = [cos(b) 0 sin(b); 0 1 0; -sin(b) 0 cos(b)]

R_Y_b =

    0.8660         0    0.5000
         0    1.0000         0
   -0.5000         0    0.8660

>> R_X_g = [1 0 0; 0 cos(g) -sin(g); 0 sin(g) cos(g)]

R_X_g =

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

```
>> A_R_ZYX_B = R_Z_a * R_Y_b * R_X_g

A_R_ZYX_B =

    0.7500   -0.2165    0.6250
    0.4330    0.8750   -0.2165
   -0.5000    0.4330    0.7500
```

```
A_R_XYZ_fixed_B =

    0.7500   -0.2165    0.6250
    0.4330    0.8750   -0.2165
   -0.5000    0.4330    0.7500

>> rotz(30) * roty(30) * rotx(30)

ans =

    0.7500   -0.2165    0.6250
    0.4330    0.8750   -0.2165
   -0.5000    0.4330    0.7500
```

Z-Y-Z Euler angles

- Start with the frame coincident with a known frame $\{A\}$. Rotate $\{B\}$ first about $\hat{Z}_B$ by an angle α, then about $\hat{Y}_B$ by an angle β, and, finally, about Z_b by an angle γ.

- $${}^A_B R_{Z'Y'Z'} = R_Z(\alpha) R_Y(\beta) R_Z(\gamma)$$
$$= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- **Apply post-multiplications of rotations (from the *<u>left to right</u>*) of $R_Z(\alpha)$, then $R_Y(\beta)$, and then $R_Z(\gamma)$.**

- $${}^A_B R_{Z'Y'Z'}(\alpha, \beta, \gamma) = \begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix}$$

Z-Y-Z Euler angles in MATLAB

```
>> syms a b g
>> R_Z_a = [cos(a) -sin(a) 0; sin(a) cos(a) 0; 0 0 1]

R_Z_a =

[cos(a), -sin(a), 0]
[sin(a),  cos(a), 0]
[     0,       0, 1]
```

```
>> R_Y_b = [cos(b) 0 sin(b); 0 1 0; -sin(b) 0 cos(b)]

R_Y_b =

[ cos(b), 0, sin(b)]
[      0, 1,      0]
[-sin(b), 0, cos(b)]
```

```
>> R_Z_g = [cos(g) -sin(g) 0; sin(g) cos(g) 0; 0 0 1]

R_Z_g =

[cos(g), -sin(g), 0]
[sin(g),  cos(g), 0]
[     0,       0, 1]
```

```
>> R_Z_a * R_Y_b * R_Z_g

ans =

[cos(a)*cos(b)*cos(g) - sin(a)*sin(g), - cos(g)*sin(a) - cos(a)*cos(b)*sin(g), cos(a)*sin(b)]
[cos(a)*sin(g) + cos(b)*cos(g)*sin(a),   cos(a)*cos(g) - cos(b)*sin(a)*sin(g), sin(a)*sin(b)]
[                      -cos(g)*sin(b),                          sin(b)*sin(g),        cos(b)]
```

$$
{}^{A}_{B}R_{Z'Y'Z'} = R_Z(\alpha)R_Y(\beta)R_Z(\gamma)
$$

$$
= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{A}_{B}R_{Z'Y'Z'}(\alpha,\beta,\gamma) = \begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix}
$$

Other angle-set conventions

- Total of 24 conventions
  - 12 conventions are for fixed-angle sets,
  - 12 are for Euler-angle sets.
- There is no reason to favor one convention over another
- Any orientation (3D) can be described by composing three elemental rotations
- A minimal representation of an orientation requires three angles *(α, β, γ)*
- **In our course we will use the concept of rotation about the current frame (Z-Y-X Euler angles) throughout the whole term.**

24 angle-set convention

The 12 Euler angle sets are given by

$$R_{X'Y'Z'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\beta c\gamma & -c\beta s\gamma & s\beta \\ s\alpha s\beta c\gamma + c\alpha s\gamma & -s\alpha s\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta \\ -c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha s\beta s\gamma + s\alpha c\gamma & c\alpha c\beta \end{bmatrix},$$

$$R_{X'Z'Y'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\beta c\gamma & -s\beta & c\beta s\gamma \\ c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma \\ s\alpha s\beta c\gamma - c\alpha s\gamma & s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

$$R_{Y'X'Z'}(\alpha,\beta,\gamma)=\begin{bmatrix} s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma & s\alpha c\beta \\ c\beta s\gamma & c\beta c\gamma & -s\beta \\ c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha c\beta \end{bmatrix},$$

$$R_{Y'Z'X'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\alpha c\beta & -c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha s\beta s\gamma + s\alpha c\gamma \\ s\beta & c\beta c\gamma & -c\beta s\gamma \\ -s\alpha c\beta & s\alpha s\beta c\gamma + c\alpha s\gamma & -s\alpha s\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

$$R_{Z'X'Y'}(\alpha,\beta,\gamma)=\begin{bmatrix} -s\alpha s\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta & s\alpha s\beta c\gamma + c\alpha s\gamma \\ c\alpha s\beta s\gamma + s\alpha c\gamma & c\alpha c\beta & -c\alpha s\beta c\gamma + s\alpha s\gamma \\ -c\beta s\gamma & s\beta & c\beta c\gamma \end{bmatrix},$$

$$R_{Z'Y'X'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & -s\alpha s\beta s\gamma + c\alpha c\gamma & -s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix},$$

$$R_{X'Y'X'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\beta & s\beta s\gamma & s\beta c\gamma \\ s\alpha s\beta & -s\alpha c\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta c\gamma - c\alpha s\gamma \\ -c\alpha s\beta & c\alpha c\beta s\gamma + s\alpha c\gamma & c\alpha c\beta c\gamma - s\alpha s\gamma \end{bmatrix},$$

$$R_{X'Z'X'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\beta & -s\beta c\gamma & s\beta s\gamma \\ c\alpha s\beta & c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma \\ s\alpha s\beta & s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

$$R_{Y'X'Y'}(\alpha,\beta,\gamma)=\begin{bmatrix} -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta & s\alpha c\beta c\gamma + c\alpha s\gamma \\ s\beta s\gamma & c\beta & -s\beta c\gamma \\ -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta & c\alpha c\beta c\gamma - s\alpha s\gamma \end{bmatrix},$$

$$R_{Y'Z'Y'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha s\beta & c\alpha c\beta s\gamma + s\alpha c\gamma \\ s\beta c\gamma & c\beta & s\beta s\gamma \\ -s\alpha c\beta c\gamma - c\alpha s\gamma & s\alpha s\beta & -s\alpha c\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

$$R_{Z'X'Z'}(\alpha,\beta,\gamma)=\begin{bmatrix} -s\alpha c\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta c\gamma - c\alpha s\gamma & s\alpha s\beta \\ c\alpha c\beta s\gamma + s\alpha c\gamma & c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha s\beta \\ s\beta s\gamma & s\beta c\gamma & c\beta \end{bmatrix},$$

$$R_{Z'Y'Z'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix}.$$

The 12 fixed angle sets are given by

$$R_{XYZ}(\gamma,\beta,\alpha)=\begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix},$$

$$R_{XZY}(\gamma,\beta,\alpha)=\begin{bmatrix} c\alpha c\beta & -c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha s\beta s\gamma + s\alpha c\gamma \\ s\beta & c\beta c\gamma & -c\beta s\gamma \\ -s\alpha c\beta & s\alpha s\beta c\gamma + c\alpha s\gamma & -s\alpha s\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

$$R_{YXZ}(\gamma,\beta,\alpha)=\begin{bmatrix} -s\alpha s\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta & s\alpha s\beta c\gamma + c\alpha s\gamma \\ c\alpha s\beta s\gamma + s\alpha c\gamma & c\alpha c\beta & -c\alpha s\beta c\gamma + s\alpha s\gamma \\ -c\beta s\gamma & s\beta & c\beta c\gamma \end{bmatrix},$$

$$R_{YZX}(\gamma,\beta,\alpha)=\begin{bmatrix} c\beta c\gamma & -s\beta & c\beta s\gamma \\ c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma \\ s\alpha s\beta c\gamma - c\alpha s\gamma & s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

$$R_{ZXY}(\gamma,\beta,\alpha)=\begin{bmatrix} s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma & s\alpha c\beta \\ c\beta s\gamma & c\beta c\gamma & -s\beta \\ c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha c\beta \end{bmatrix},$$

$$R_{ZYX}(\gamma,\beta,\alpha)=\begin{bmatrix} c\beta c\gamma & -c\beta s\gamma & s\beta \\ s\alpha s\beta c\gamma + c\alpha s\gamma & -s\alpha s\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta \\ -c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha s\beta s\gamma + s\alpha c\gamma & c\alpha c\beta \end{bmatrix},$$

$$R_{XYX}(\gamma,\beta,\alpha)=\begin{bmatrix} c\beta & s\beta s\gamma & s\beta c\gamma \\ s\alpha s\beta & -s\alpha c\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta c\gamma - c\alpha s\gamma \\ -c\alpha s\beta & c\alpha c\beta s\gamma + s\alpha c\gamma & c\alpha c\beta c\gamma - s\alpha s\gamma \end{bmatrix},$$

$$R_{XZX}(\gamma,\beta,\alpha)=\begin{bmatrix} c\beta & -s\beta c\gamma & s\beta s\gamma \\ c\alpha s\beta & c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma \\ s\alpha s\beta & s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

$$R_{YXY}(\gamma,\beta,\alpha)=\begin{bmatrix} -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta & s\alpha c\beta c\gamma + c\alpha s\gamma \\ s\beta s\gamma & c\beta & -s\beta c\gamma \\ -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta & c\alpha c\beta c\gamma - s\alpha s\gamma \end{bmatrix},$$

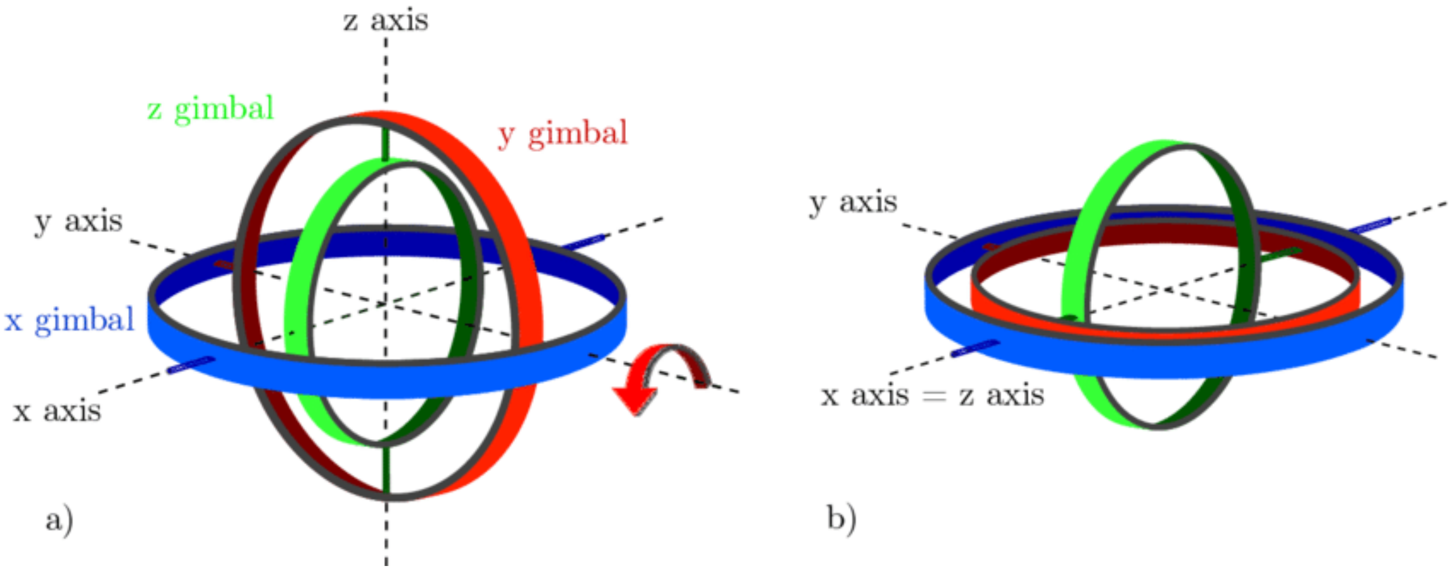
$$R_{YZY}(\gamma,\beta,\alpha)=\begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha s\beta & c\alpha c\beta s\gamma + s\alpha c\gamma \\ s\beta c\gamma & c\beta & s\beta s\gamma \\ -s\alpha c\beta c\gamma - c\alpha s\gamma & s\alpha s\beta & -s\alpha c\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

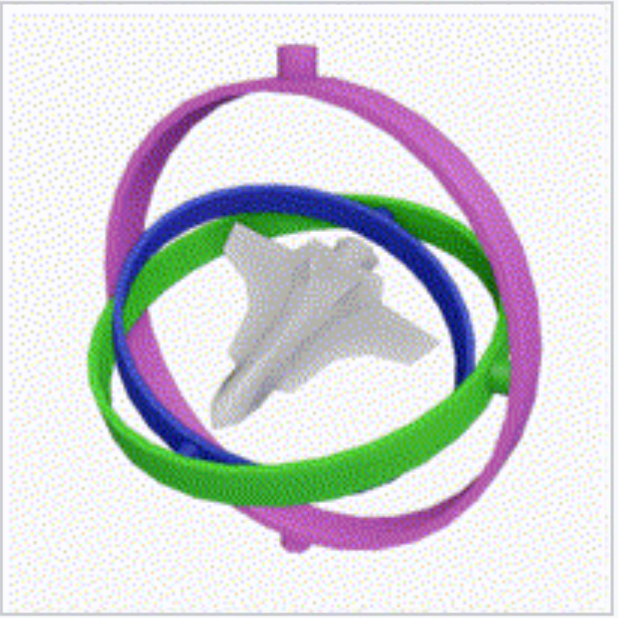
$$R_{ZXZ}(\gamma,\beta,\alpha)=\begin{bmatrix} -s\alpha c\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta c\gamma - c\alpha s\gamma & s\alpha s\beta \\ c\alpha c\beta s\gamma + s\alpha c\gamma & c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha s\beta \\ s\beta s\gamma & s\beta c\gamma & c\beta \end{bmatrix},$$

$$R_{ZYZ}(\gamma,\beta,\alpha)=\begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix}.$$

Gimbal Lock

- Gimbal lock is a phenomenon where the loss of one degree of freedom occurs due to the alignment of two of the three rotational axes
- It is a critical issue in Euler angle representations
- Solution → Quaternions



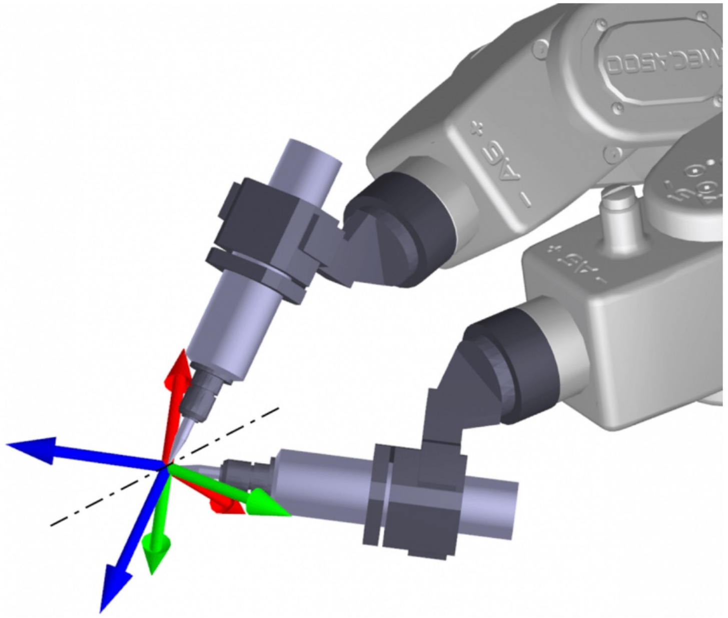


Gimbal locked airplane. When the pitch (green) and yaw (magenta) gimbals become aligned, changes to roll (blue) and yaw apply the same rotation to the airplane.

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

Quaternions

- Avoid gimbal lock—a phenomenon that occurs in Euler angle representations
- Gimbal lock arises when the rotation axes align, restricting the degrees of freedom
- Being a four-dimensional extension of complex numbers, provide a more robust solution by avoiding the inherent limitations of Euler angles
- Interpolate smoothly between rotations, preventing the occurrence of gimbal lock

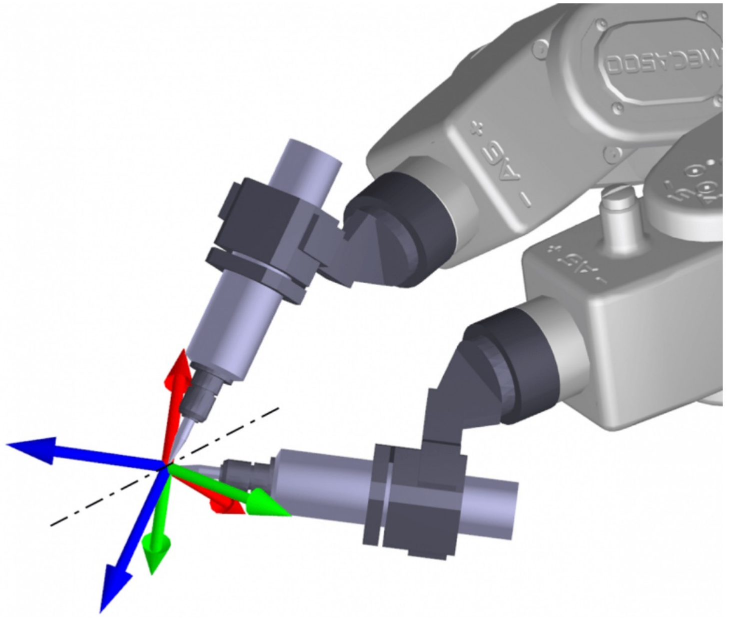


https://en.wikipedia.org/wiki/Conversion_between_quaternions_and_Euler_angles

Quaternions - Euler parameters

- $\epsilon_1 = k_x \sin\frac{\theta}{2},$
- $\epsilon_2 = k_y \sin\frac{\theta}{2},$
- $\epsilon_3 = k_z \sin\frac{\theta}{2},$
- $\epsilon_4 = \cos\frac{\theta}{2}.$
- $\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 + \epsilon_4^2 = 1$

- $\epsilon_1 = \frac{r_{32} - r_{23}}{4\epsilon_4},$
- $\epsilon_2 = \frac{r_{13} - r_{31}}{4\epsilon_4},$
- $\epsilon_3 = \frac{r_{21} - r_{12}}{4\epsilon_4},$
- $\epsilon_4 = \frac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}}.$



- $R_\epsilon = \begin{bmatrix} 1 - 2\epsilon_2^2 - 2\epsilon_3^2 & 2(\epsilon_1\epsilon_2 - \epsilon_3\epsilon_4) & 2(\epsilon_1\epsilon_3 + \epsilon_2\epsilon_4) \\ 2(\epsilon_1\epsilon_2 + \epsilon_3\epsilon_4) & 1 - 2\epsilon_1^2 - 2\epsilon_3^2 & 2(\epsilon_2\epsilon_3 - \epsilon_1\epsilon_4) \\ 2(\epsilon_1\epsilon_3 - \epsilon_2\epsilon_4) & 2(\epsilon_2\epsilon_3 + \epsilon_1\epsilon_4) & 1 - 2\epsilon_1^2 - 2\epsilon_2^2 \end{bmatrix}$

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

… end of Lecture 3