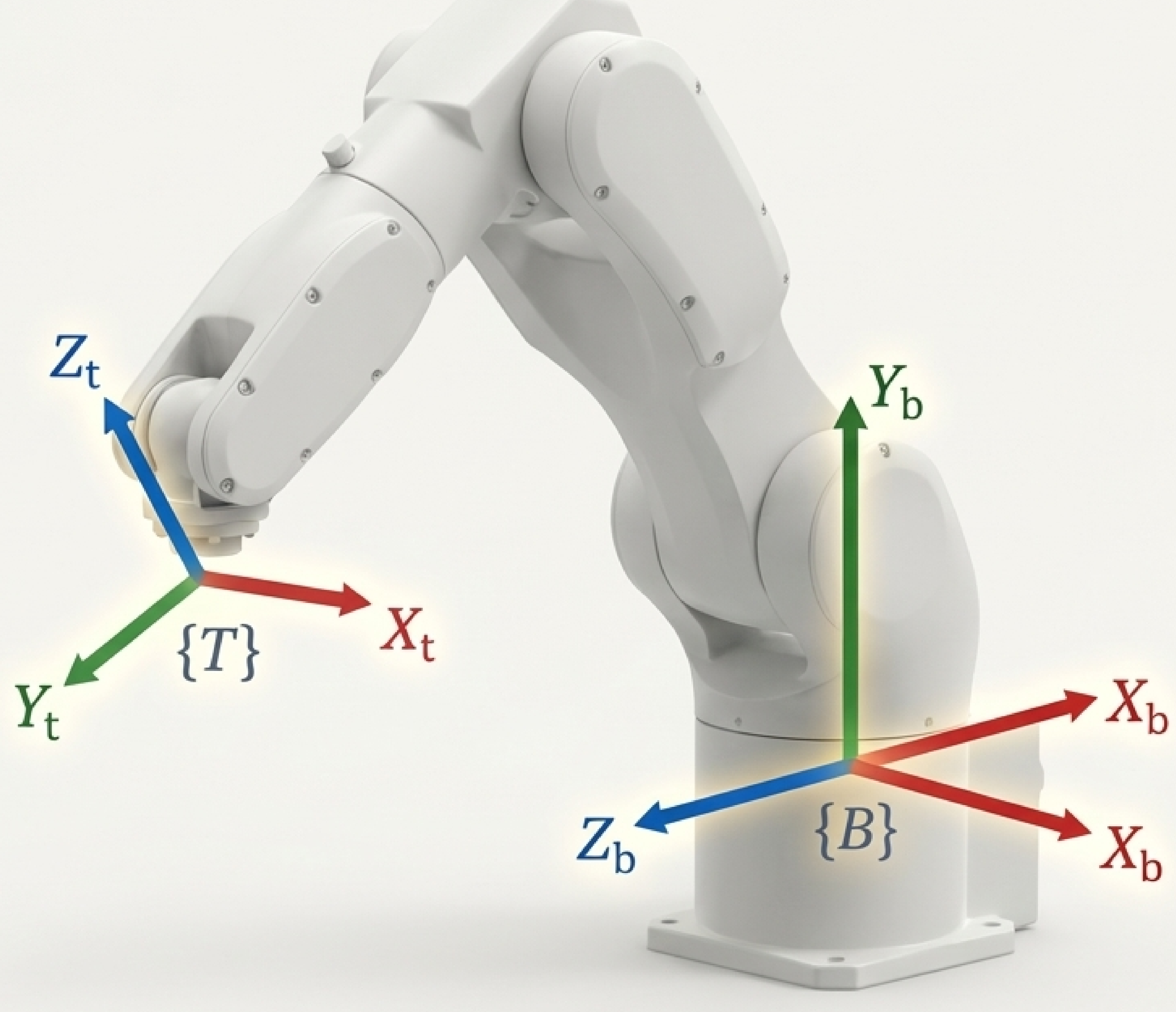
ME 5751: 3D Robot Kinematics

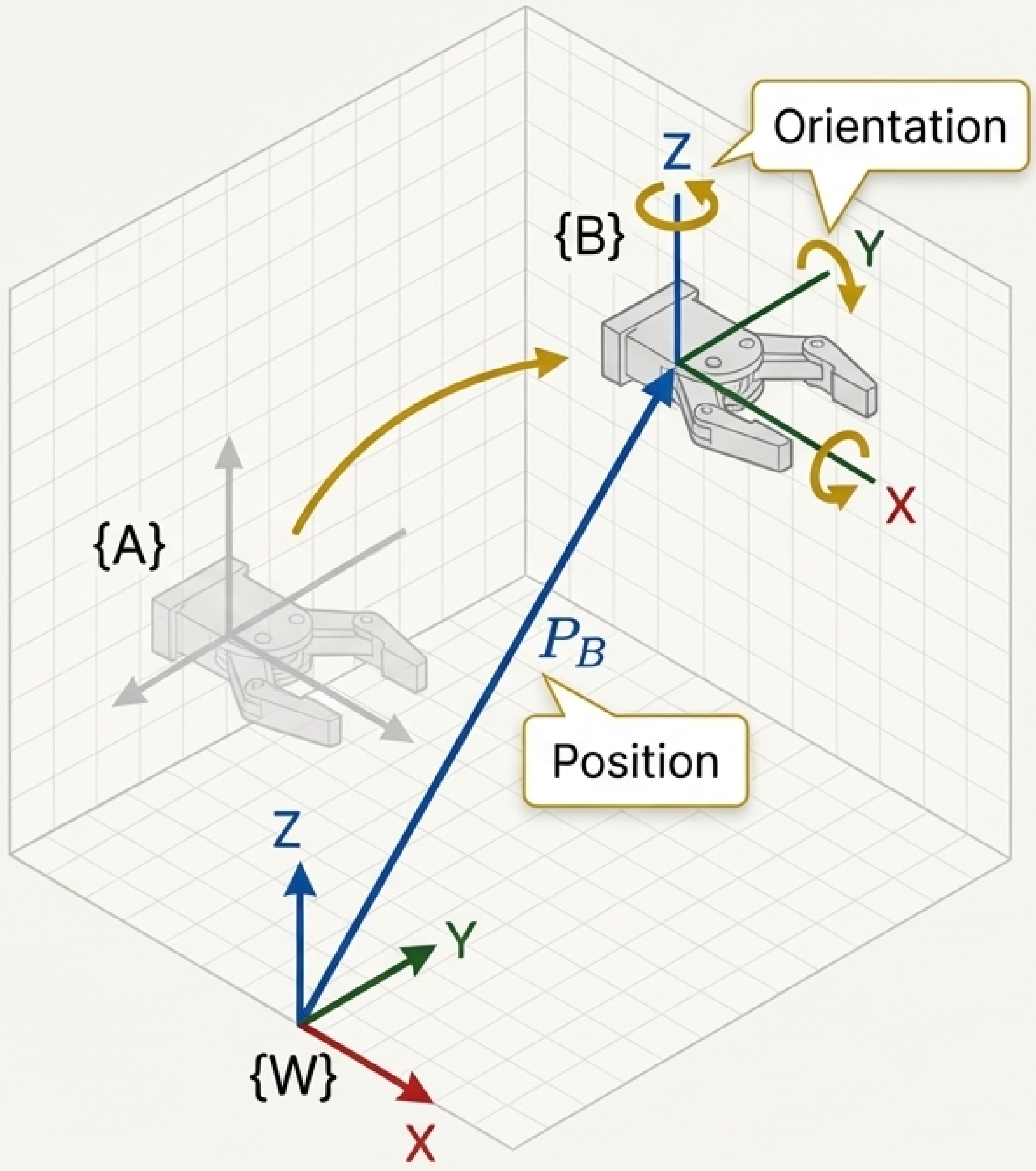
The Language of Pose in Inter



The Challenge: How do we we command a robot to a target pose?

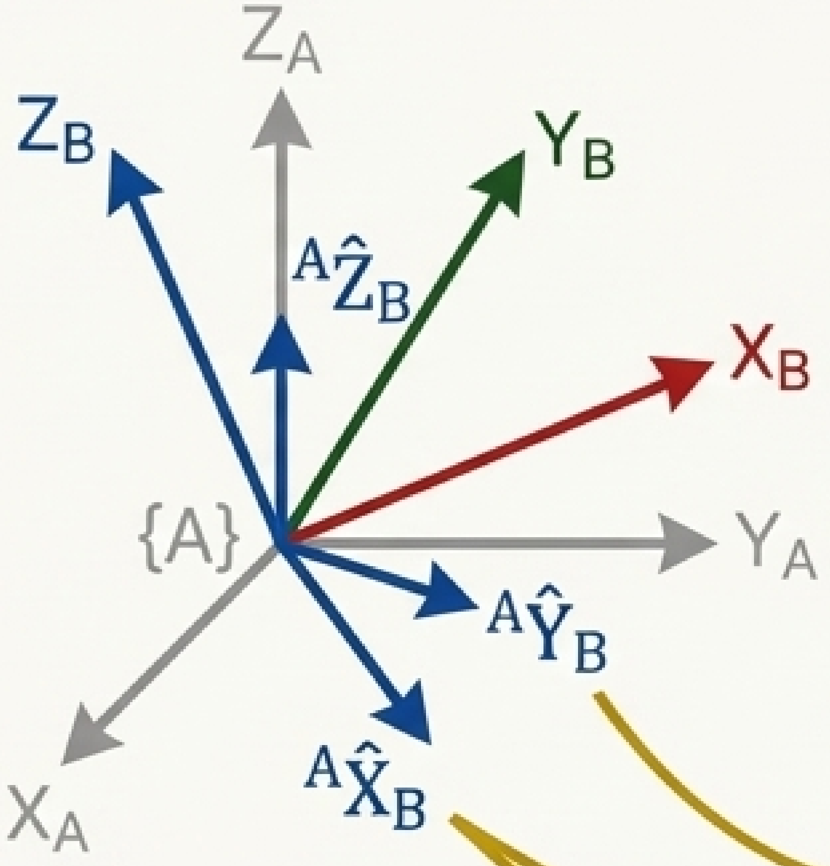
To move a robot's tool from an initial state {A} to a target state {B}, we need an unambiguous mathematical description of its **position** and its **orientation**. This combined description is its **pose**.

- **Position**: Where is the object in space? (3 degrees of freedom: x, y, z)
- **Orientation**: How is the object rotated? (3 degrees of freedom: e.g., roll, pitch, yaw)



Describing Orientation: The Rotation Matrix ${}^{A}_{B}R$

- Our strategy: Attach a coordinate frame {B}, rigidly, to the body.
- The orientation of frame {B} relative to a reference frame {A} is then fully described by the unit vectors of {B}'s principal axes, as seen from frame {A}.
- We arrange these unit vectors, expressed in the coordinates of frame {A}, as the columns of a 3x3 matrix. This is the **Rotation Matrix**.



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

Working with Rotation Matrices: Properties and Composition

Properties

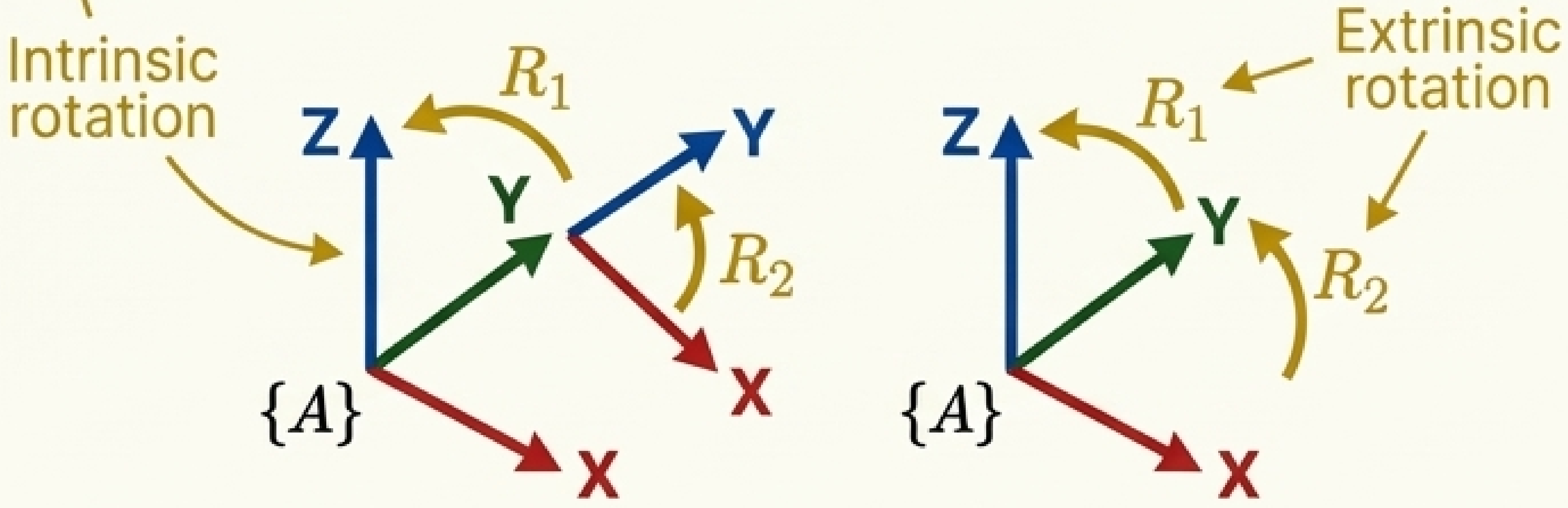
- A rotation matrix is special. Its columns are mutually orthogonal unit vectors. This property is called **orthonormality**.
- A key consequence: The inverse is simply the transpose. This avoids computationally expensive matrix inversion.
- The determinant is always +1.

$$R^T R = I$$

$${}^B_A R = \left({}^A_B R\right)^{-1} = \left({}^A_B R\right)^T$$

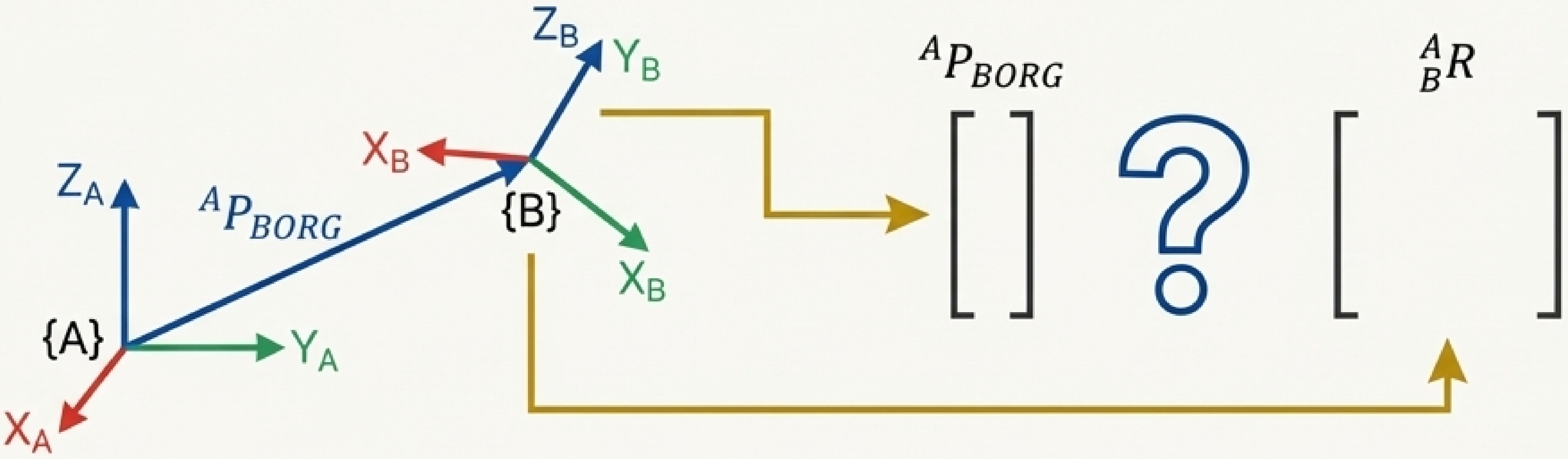
Composition

- To combine successive rotations, we multiply their matrices. The order of multiplication is critical.
- **Intrinsic Rotations** (about the body's current axes): Post-multiply. Final rotation $R = R_1 R_2$.
- **Extrinsic Rotations** (about the fixed reference axes): Pre-multiply. Final rotation $R = R_2 R_1$.



The Next Challenge: Unifying Position and Orientation

We can now describe the pose of a frame {B} relative to {A}. This requires two separate mathematical objects:



This is cumbersome. Performing operations like composing multiple movements would require separate matrix and vector algebra.

Can we combine these into a single mathematical entity for a unified description and simpler manipulation?

Solution: The 4x4 Homogeneous Transformation Matrix ${}^A_BT$

By augmenting our mathematical space, we can create a single matrix to represent a complete pose.

Position Vectors are augmented from 3x1 to 4x1 by adding a ‘1’ as the fourth element. The Transformation Matrix is a 4x4 matrix that embeds the 3x3 rotation matrix and the 3x1 position vector in a single structure.

$$\left[\begin{array}{cc} \overbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}}^{\text{Rotation } ({}^A_BR)} & \overbrace{\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}}^{\text{Position } ({}^AP_{BORG})} \\ \underbrace{[0 \quad 0 \quad 0]}_{\text{1x3}} & \underbrace{[1]}_{\text{1x1}} \end{array}\right]$$

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \xrightarrow{\text{Augment}} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

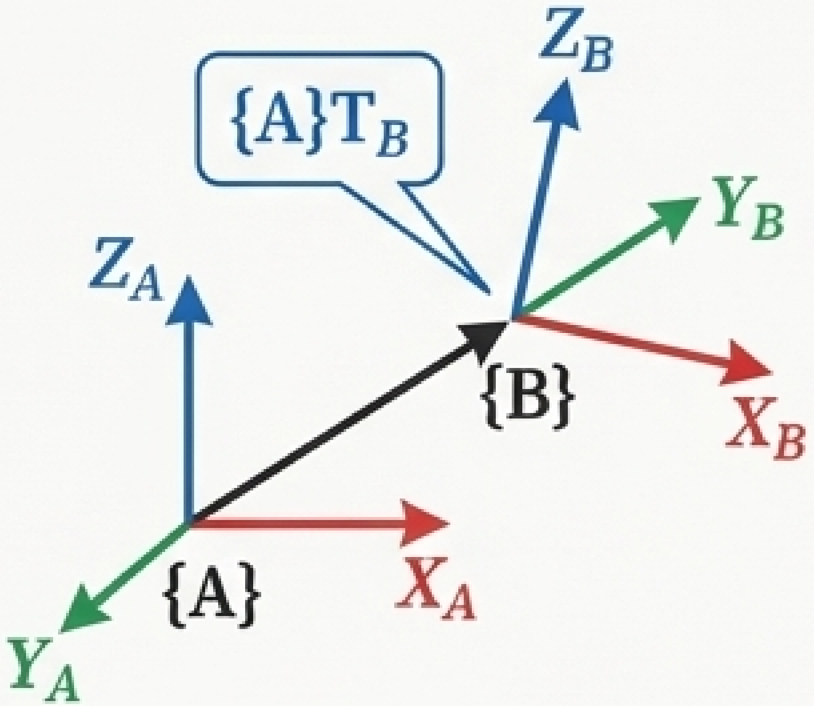
The equation ${}^AP = {}^A_BR\,{}^BP + {}^AP_{BORG}$ becomes:

$$\begin{bmatrix} {}^AP \\ 1 \end{bmatrix} = \begin{bmatrix} {}^A_BR & {}^AP_{BORG} \\ 0 \;\; 0 \;\; 0 & 1 \end{bmatrix} \begin{bmatrix} {}^BP \\ 1 \end{bmatrix}$$

The Power of Unification: Three Interpretations of ${}^{A}_{B}T$

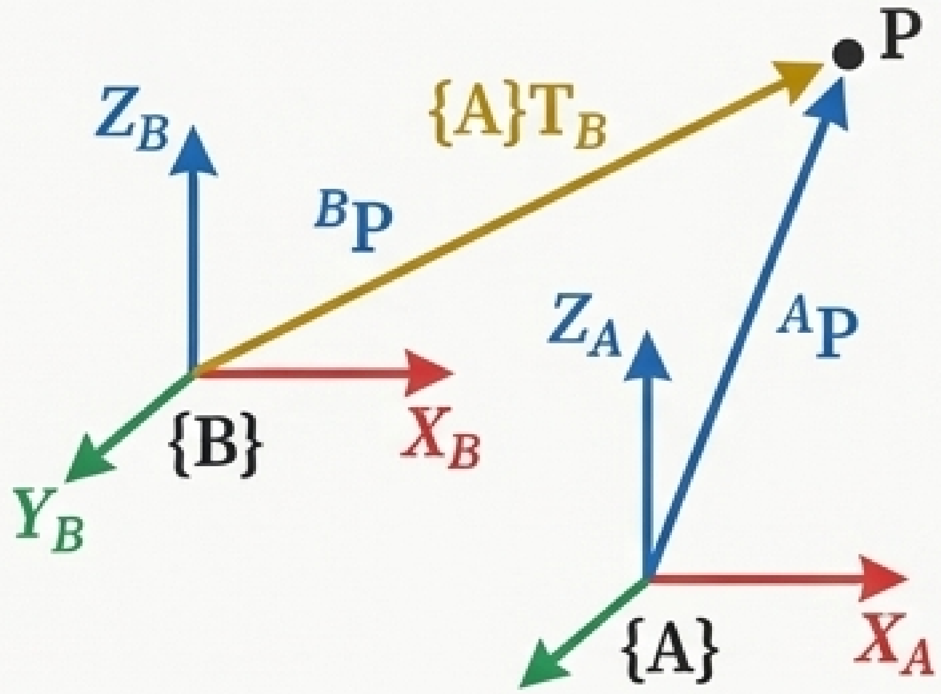
A homogeneous transform is a powerful tool precisely because it can be interpreted in three distinct ways. Mastering these views is the key to fluency in kinematics.

1. A Description



It provides a full description of the pose (position and orientation) of frame {B} relative to frame {A}.

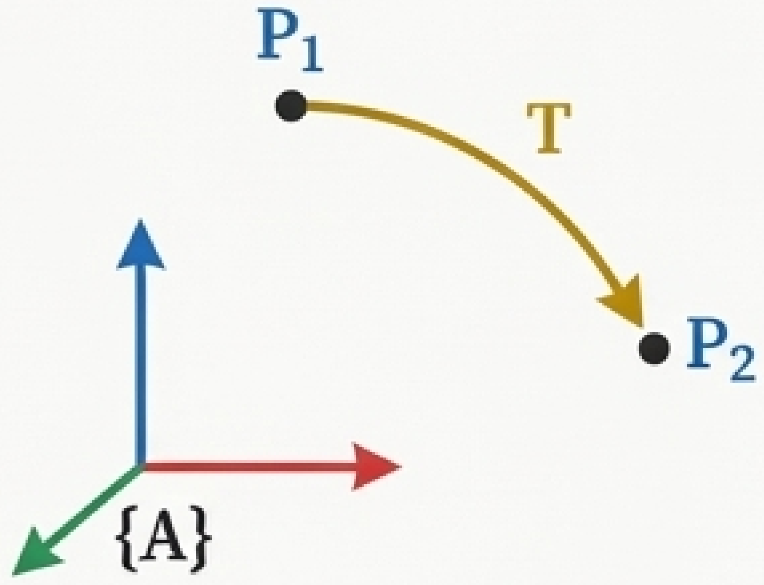
2. A Mapping



It is a function that maps the coordinates of a point from its description in frame {B} to its description in frame {A}.

$$ {}^{A}P = {}^{A}_{B}T \cdot {}^{B}P $$

3. An Operator



It is an operator that takes a point P1 and moves it to a new point P2 within the same frame {A}.

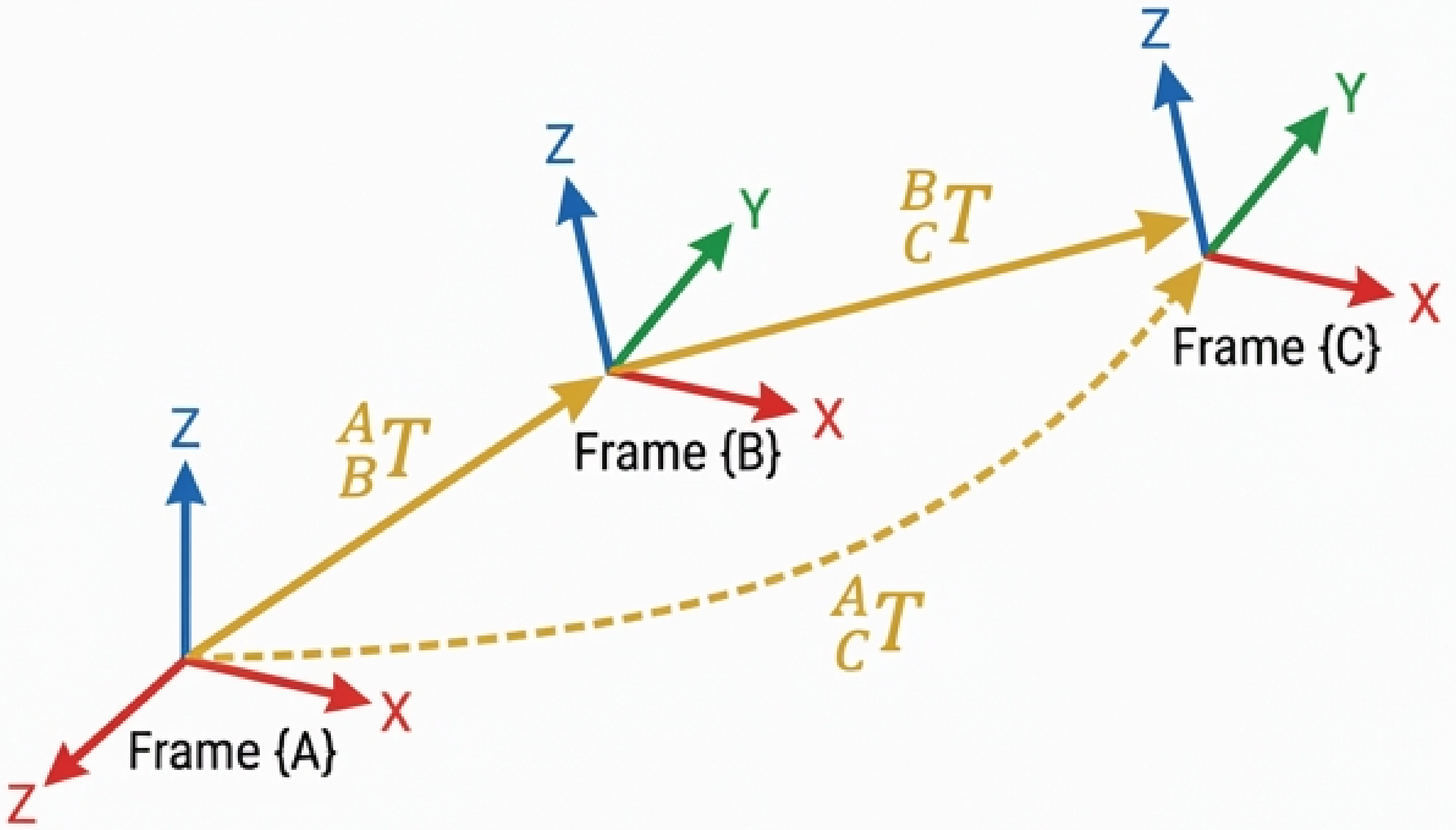
$$ {}^{A}P_2 = T \cdot {}^{A}P_1 $$

Transformation Arithmetic: Composition and Inversion

Composition

Composition: To find the description of a frame {C} relative to {A}, when we know ${}^{A}_{B}T$ and ${}^{B}_{C}T$, we simply multiply the matrices. Note the 'subscript cancellation' pattern, which helps in keeping track of transformations.

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



Inversion

Inversion: To find the description of frame {A} relative to {B}, we compute the inverse of ${}^{A}_{B}T$. A computationally efficient formula exists that leverages the special structure of the transform, avoiding a generic 4×4 matrix inversion.

$$ ({}^{A}_{B}T)^{-1} = {}^{B}_{A}T = \begin{bmatrix} ({}^{A}_{B}R)^{T} & -({}^{A}_{B}R)^{T}\,{}^{A}P_{BORG} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} $$

The Orientation Problem: Redundancy and Intuition

A 3x3 rotation matrix uses nine numbers to represent an orientation, which is a three degree-of-freedom quantity. The nine numbers are constrained by six orthonormality conditions. This representation is **redundant** and **non-intuitive**.

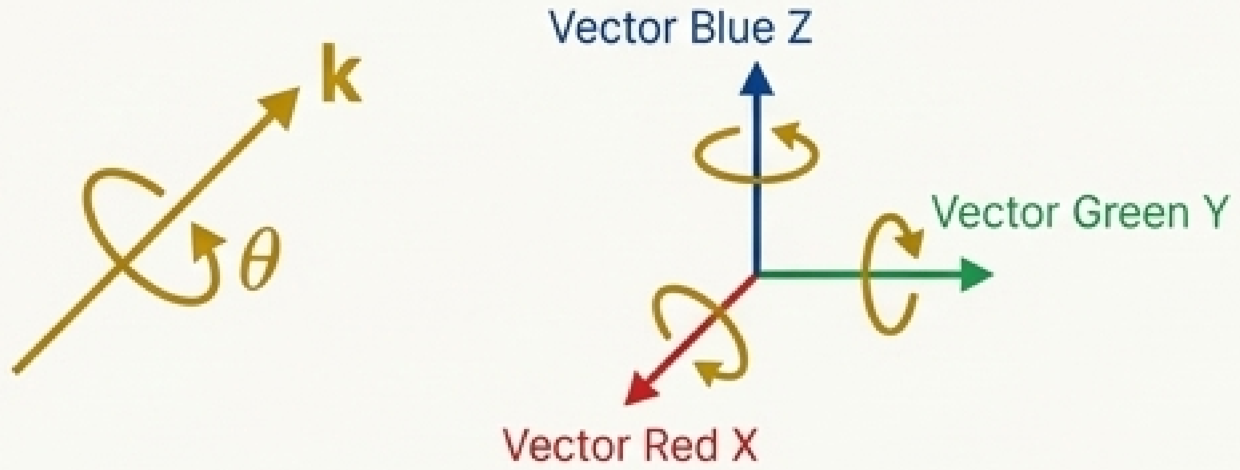
Can we find a more compact, intuitive, 3-parameter representation?

9 numbers

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Vs.

3 degrees of freedom



Equivalent Angle-Axis: Based on Euler's theorem, any orientation tentation can be described by a single rotation (θ) about a single axis (k). This is the most physically meaningful description.

Euler/Fixed Angles: Describes orientation as a sequence of three rotations about principal axes (e.g., Z-Y-X). Intuitive for humans, but suffers from representational singularities known as 'gimbal lock,' where a degree of freedom is lost.

The Power Solution: Unit Quaternions (Euler Parameters)

A four-parameter representation $\{\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4\}$ derived from the angle-axis representation (k, θ). Also known as a **unit quaternion**.

Pros:

- Avoids singularities (no gimbal lock).
- Computationally more efficient and numerically stable for composing rotations (interpolation and integration).
- Globally non-singular.

Cons:

Less intuitive to visualize directly than Euler angles or angle-axis.

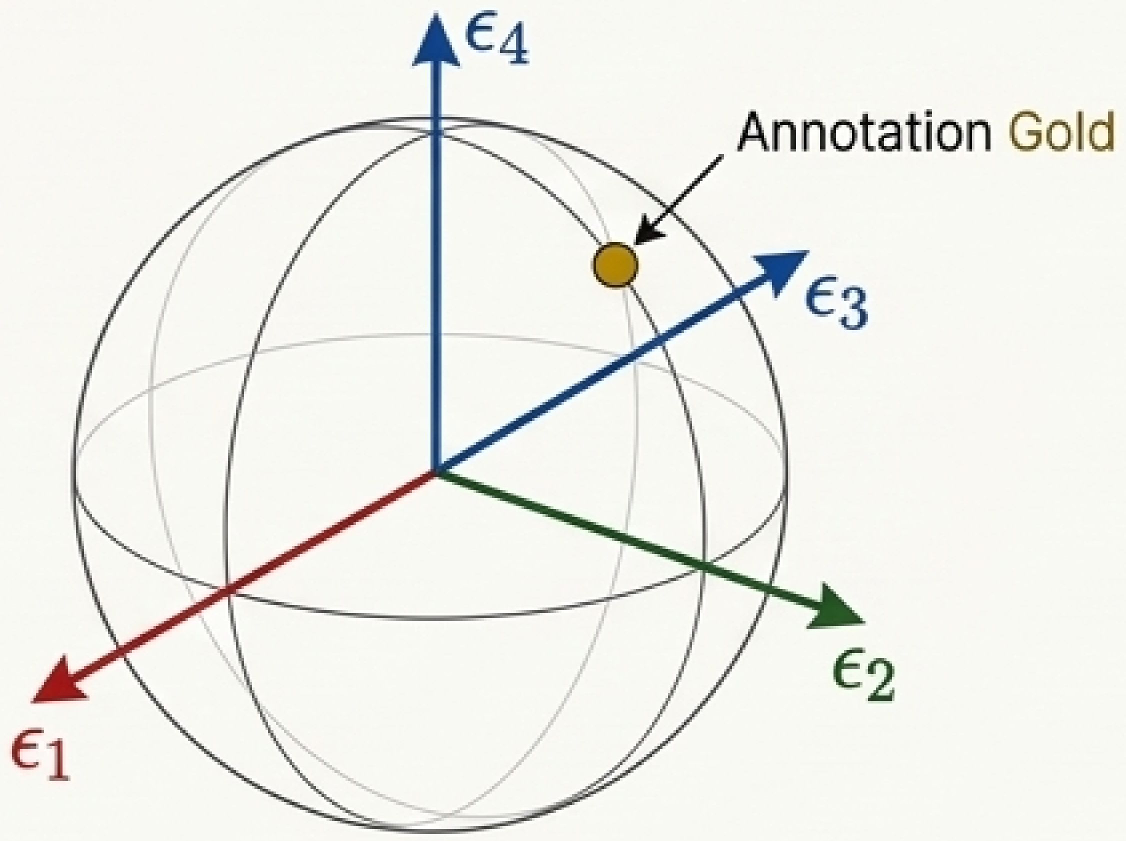
Bottom Line:

This is the standard representation used in modern 3D graphics, aerospace simulation, and professional robotics software.

Conversion from Angle-Axis

$$\epsilon_1 = k_x \sin(\theta/2)$$
$$\epsilon_2 = k_y \sin(\theta/2)$$
$$\epsilon_3 = k_z \sin(\theta/2)$$
$$\epsilon_4 = \cos(\theta/2)$$

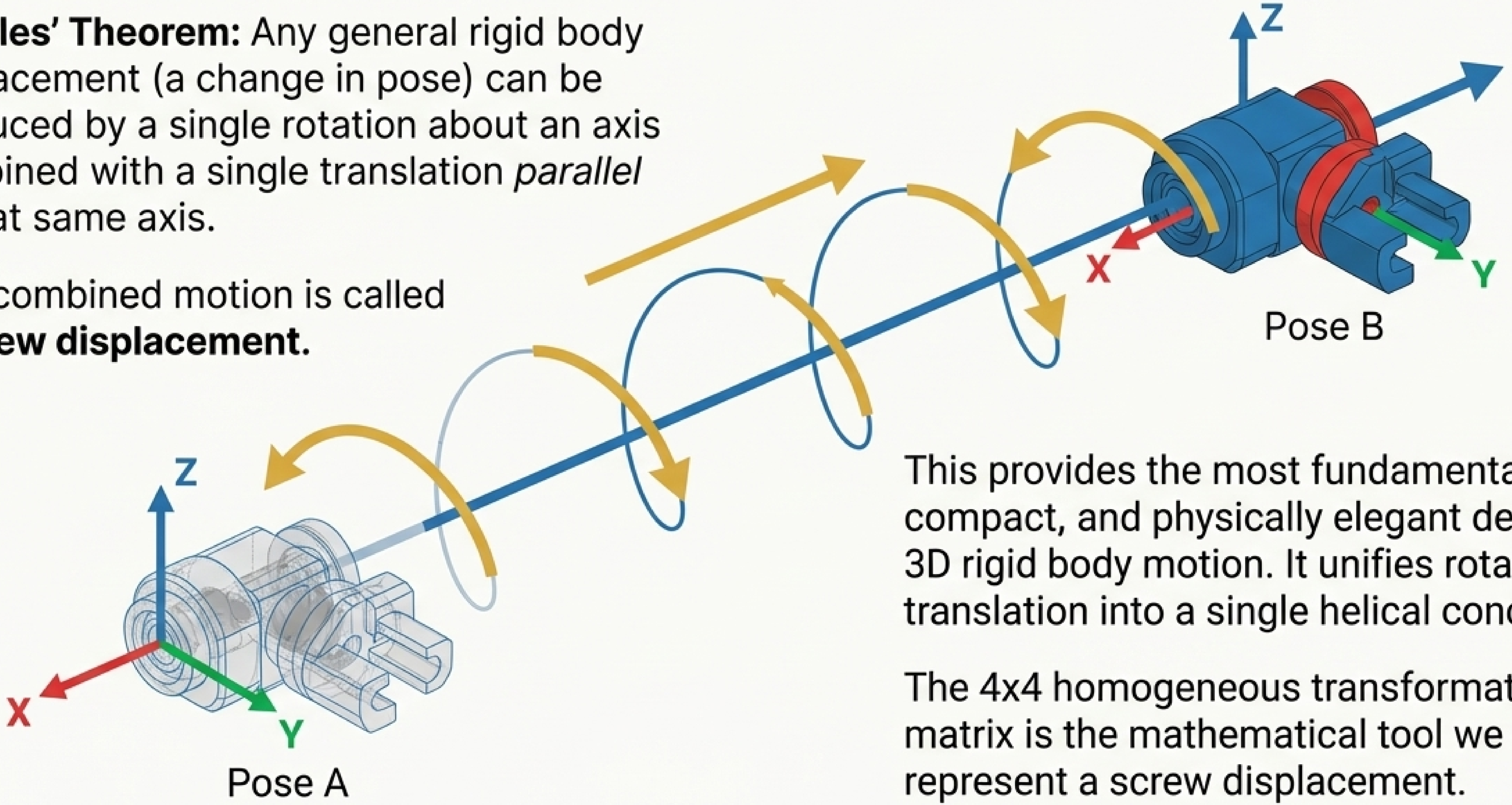
Constraint: $\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 + \epsilon_4^2 = 1$



The Grand Unification: The Screw Axis

Chasles' Theorem: Any general rigid body displacement (a change in pose) can be produced by a single rotation about an axis combined with a single translation *parallel* to that same axis.

This combined motion is called a **screw displacement.**



This provides the most fundamental, compact, and physically elegant description of 3D rigid body motion. It unifies rotation and translation into a single helical concept.

The 4x4 homogeneous transformation matrix is the mathematical tool we use to represent a screw displacement.

Our Kinematic Toolkit for Positional Analysis

Concept	Mathematical Tool	Primary Use
Position	3x1 Vector (P)	Describes the location of a point.
Orientation	3x3 Rotation Matrix (R)	Describes rotation; maps vector coordinates.
Pose	4x4 Transformation Matrix (T)	Unified description of pose; maps/operates on points.
Composition	Matrix Multiplication	Chaining descriptions of frames together.
Compact Orientation	Angle-Axis, Quaternions	Intuitive specification, robust computation.
General Displacement	Screw Axis	The fundamental physical description of motion.

Next Steps: From Describing Position to Commanding Motion

We have now built a complete toolkit for **static position analysis**—the language to describe any pose in 3D space. The next logical step is to analyze and command motion.

Upcoming Topics:

- **Inverse Kinematics:** Given a desired pose, how do we calculate the required joint angles?
- **Velocities & Jacobians:** How do we relate joint speeds to the end-effector's linear and angular velocity?
- **Dynamics:** What forces and torques are required to create a desired motion?