Arduinobot

Introduction

Setup

Digital Twin

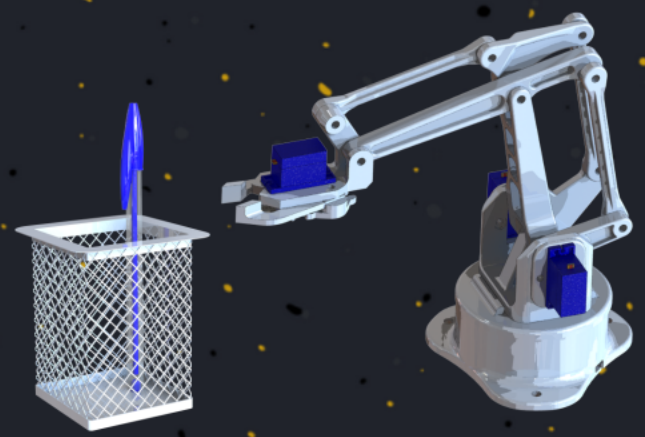
ROS 2

Control

Kinematics

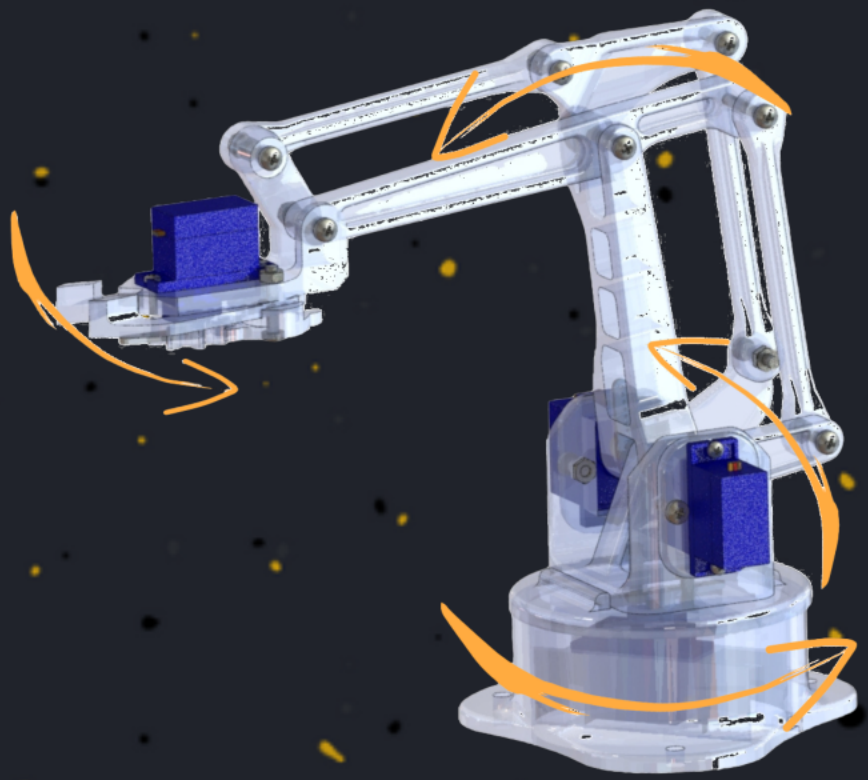
Application

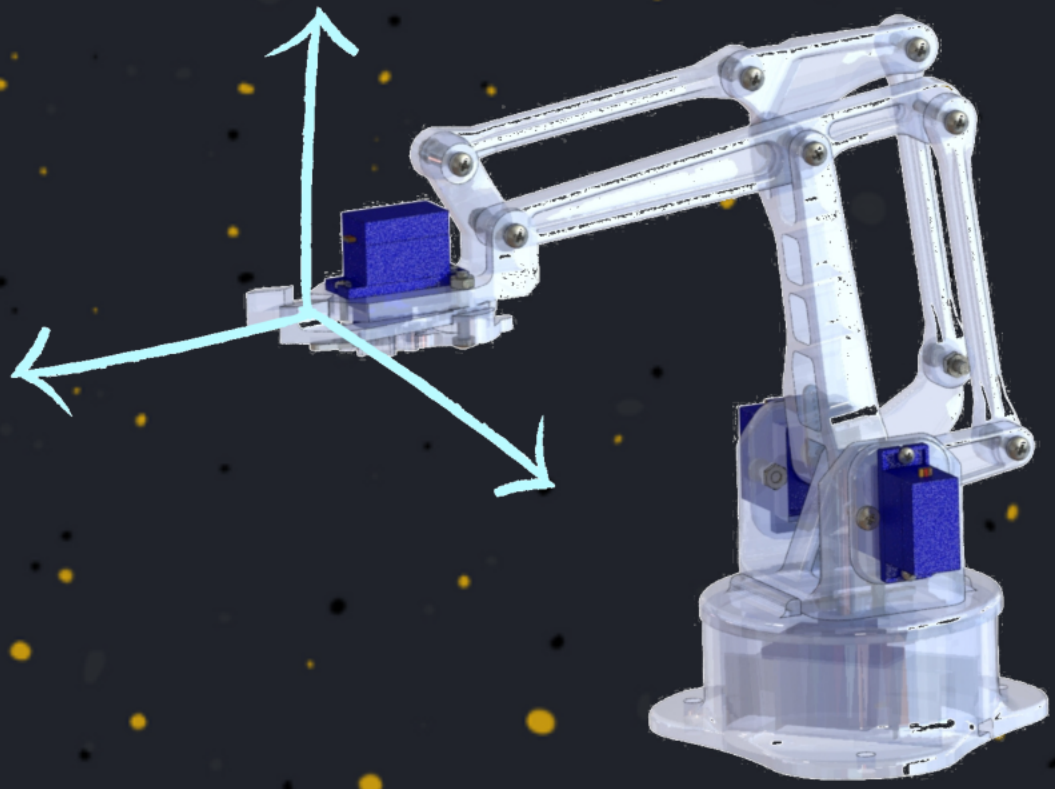
Alexa

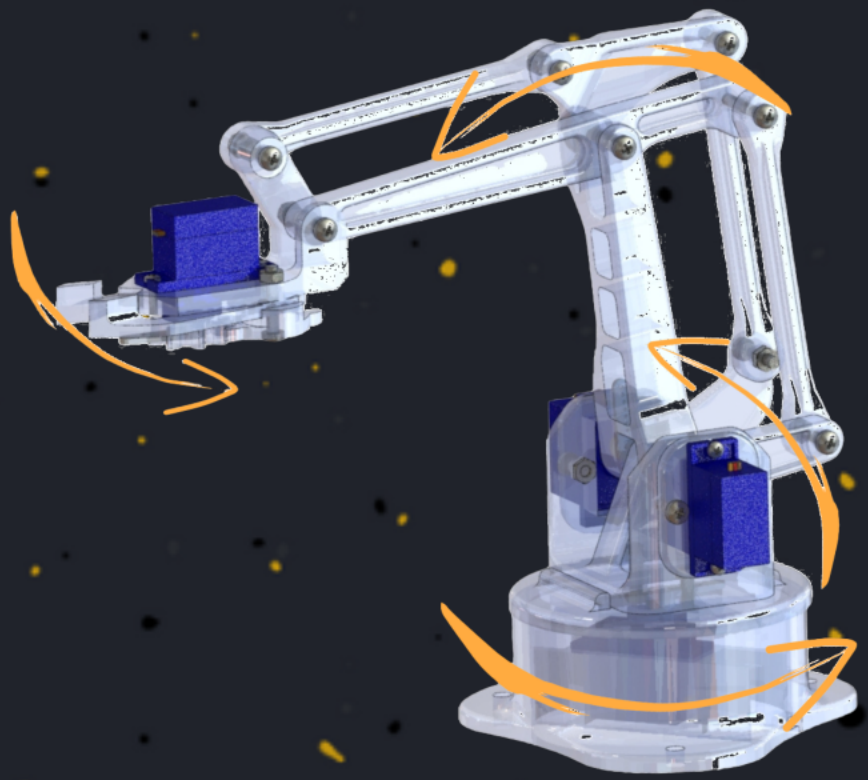


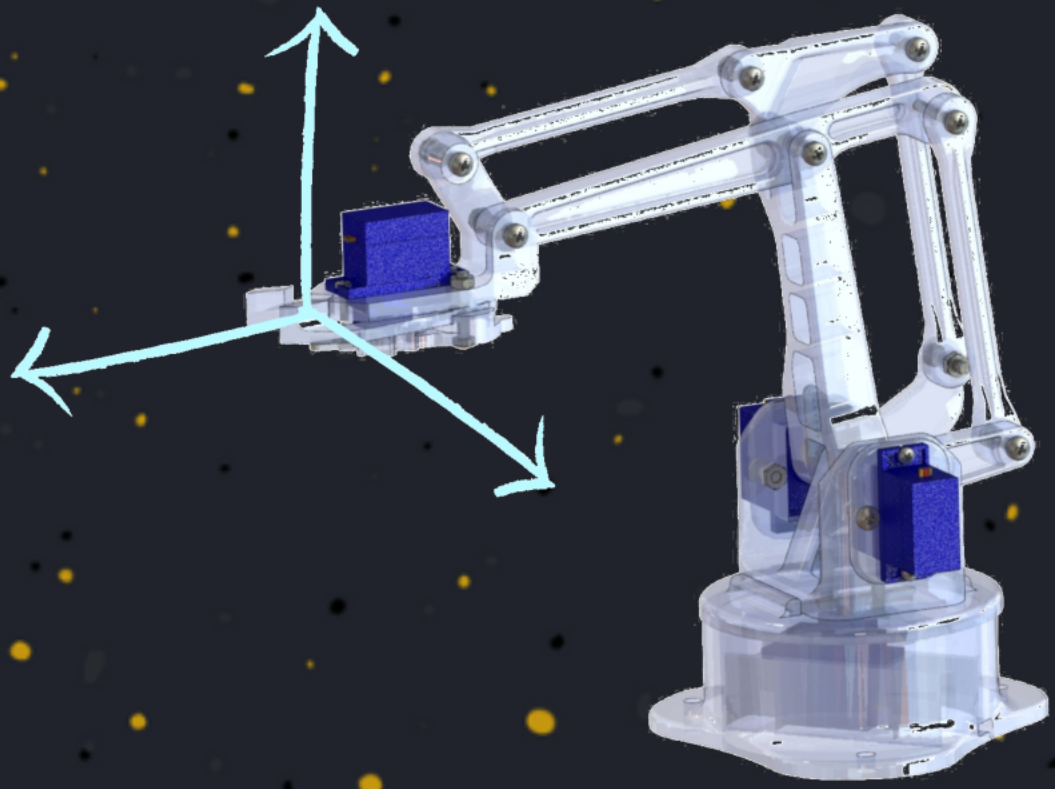
Conclusions

Build

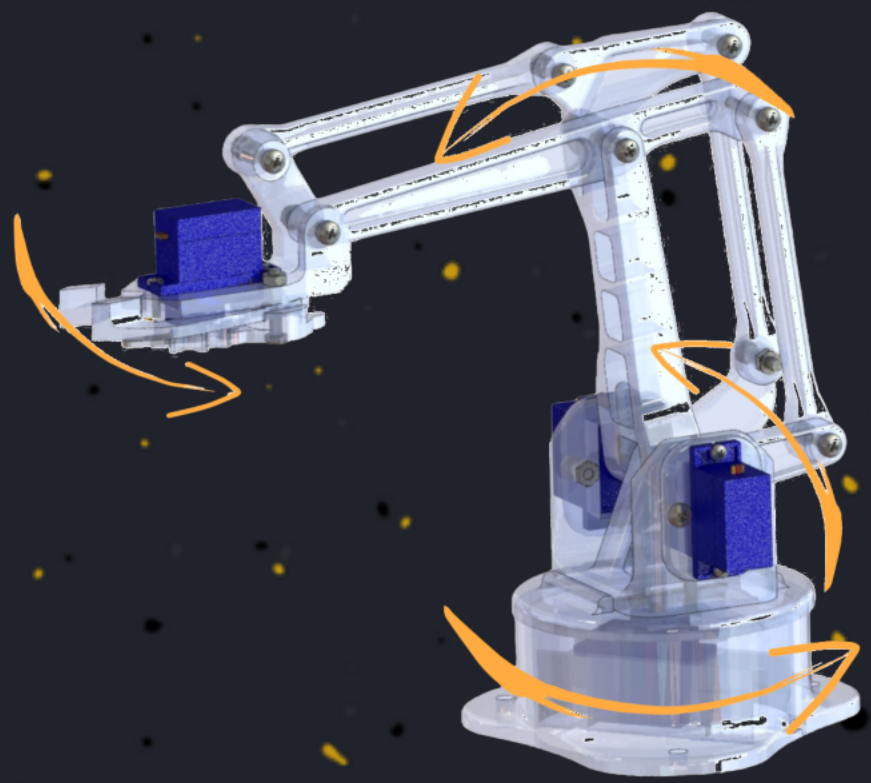


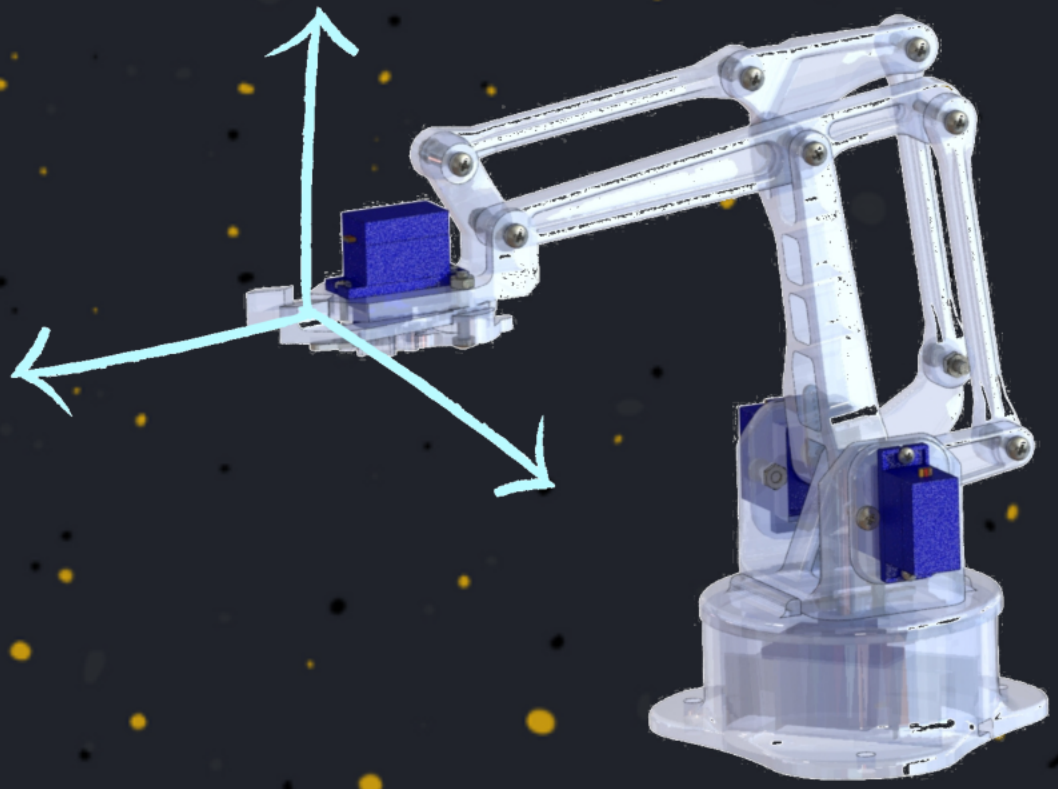




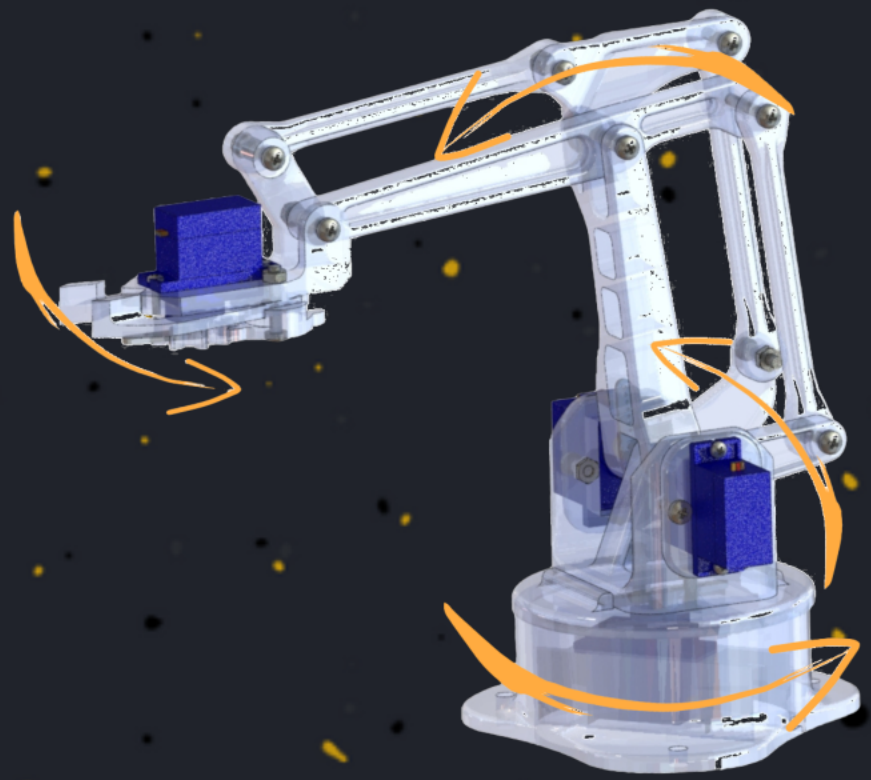


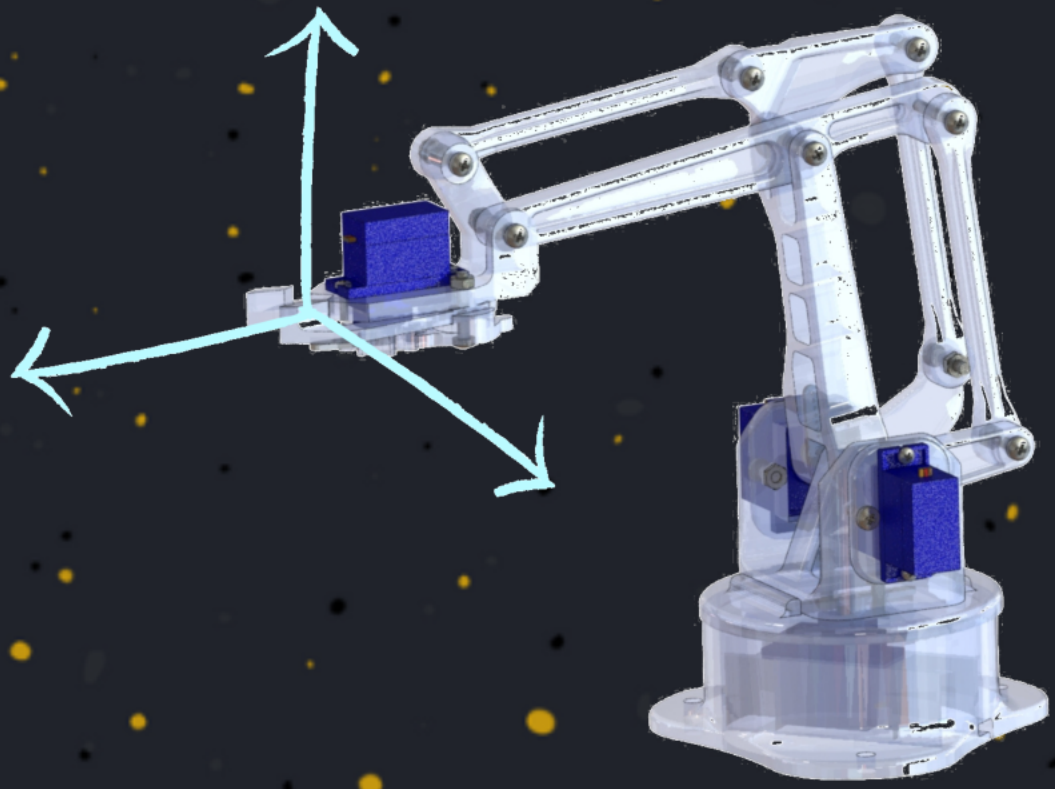
Forward



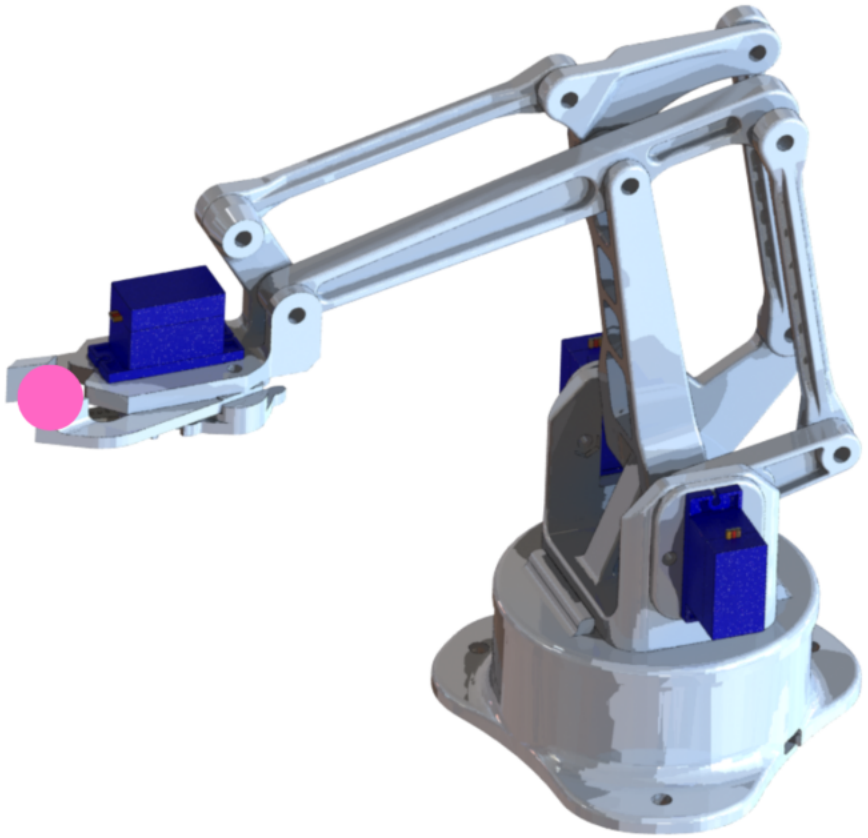


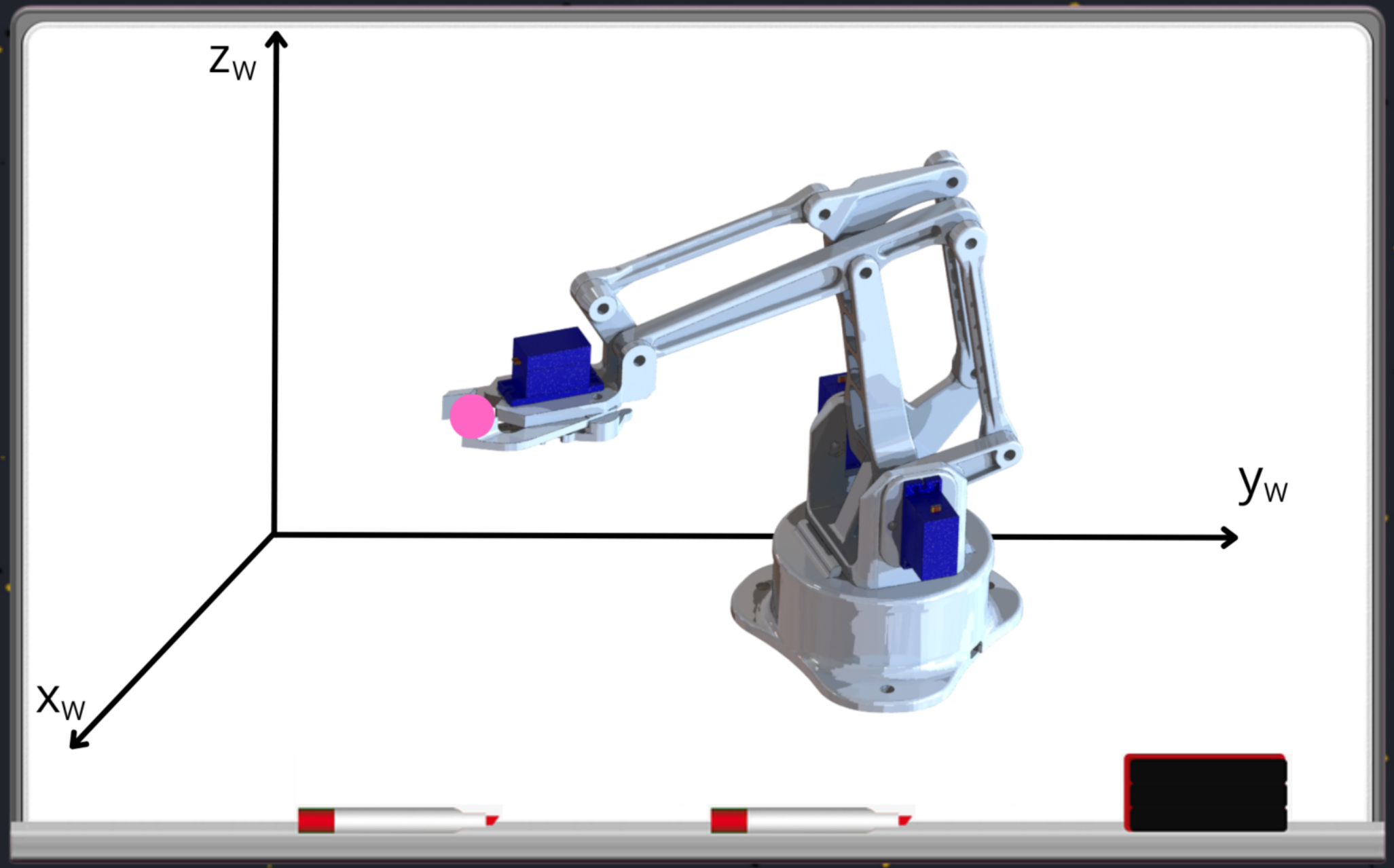
Forward

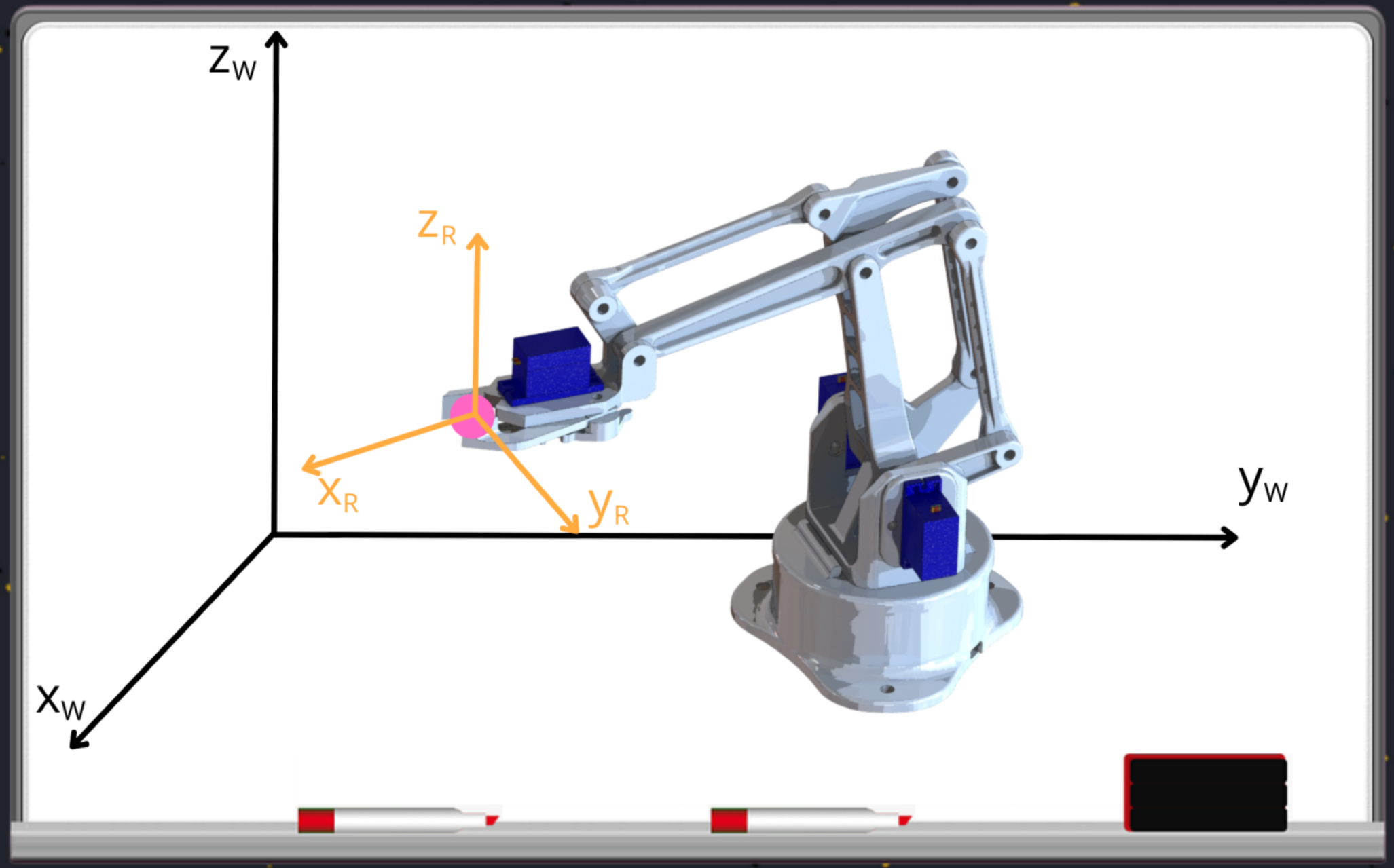


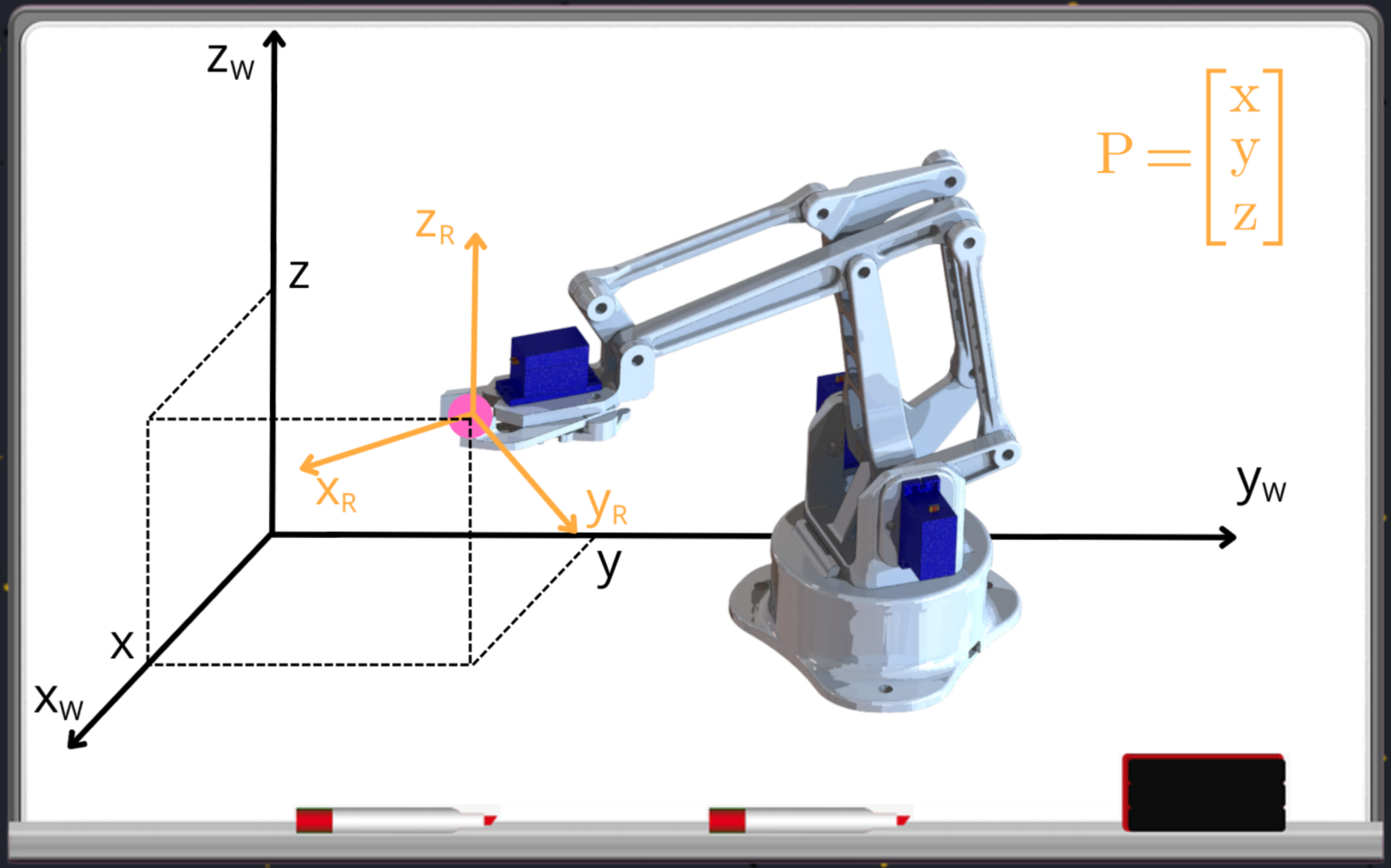


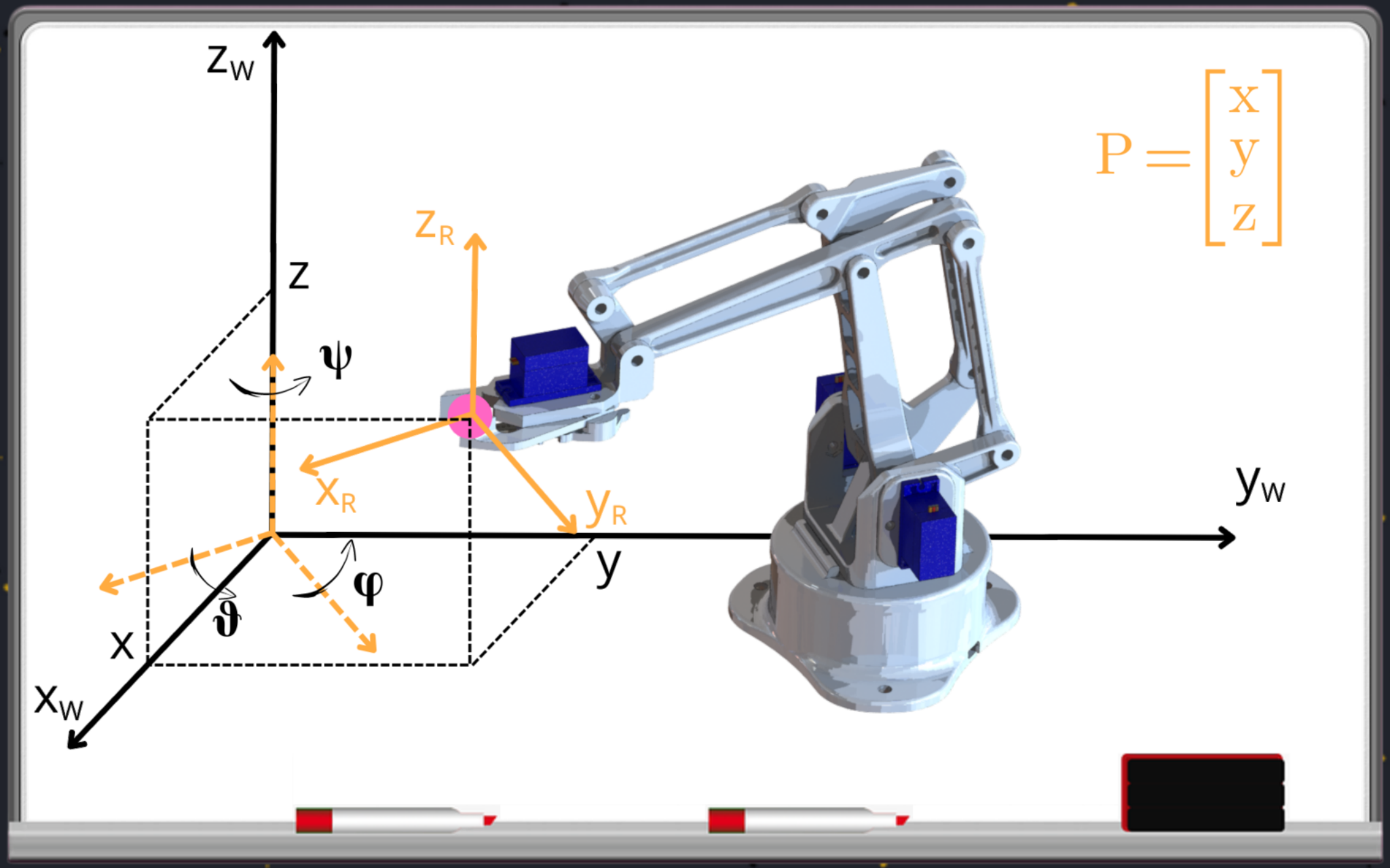
Inverse

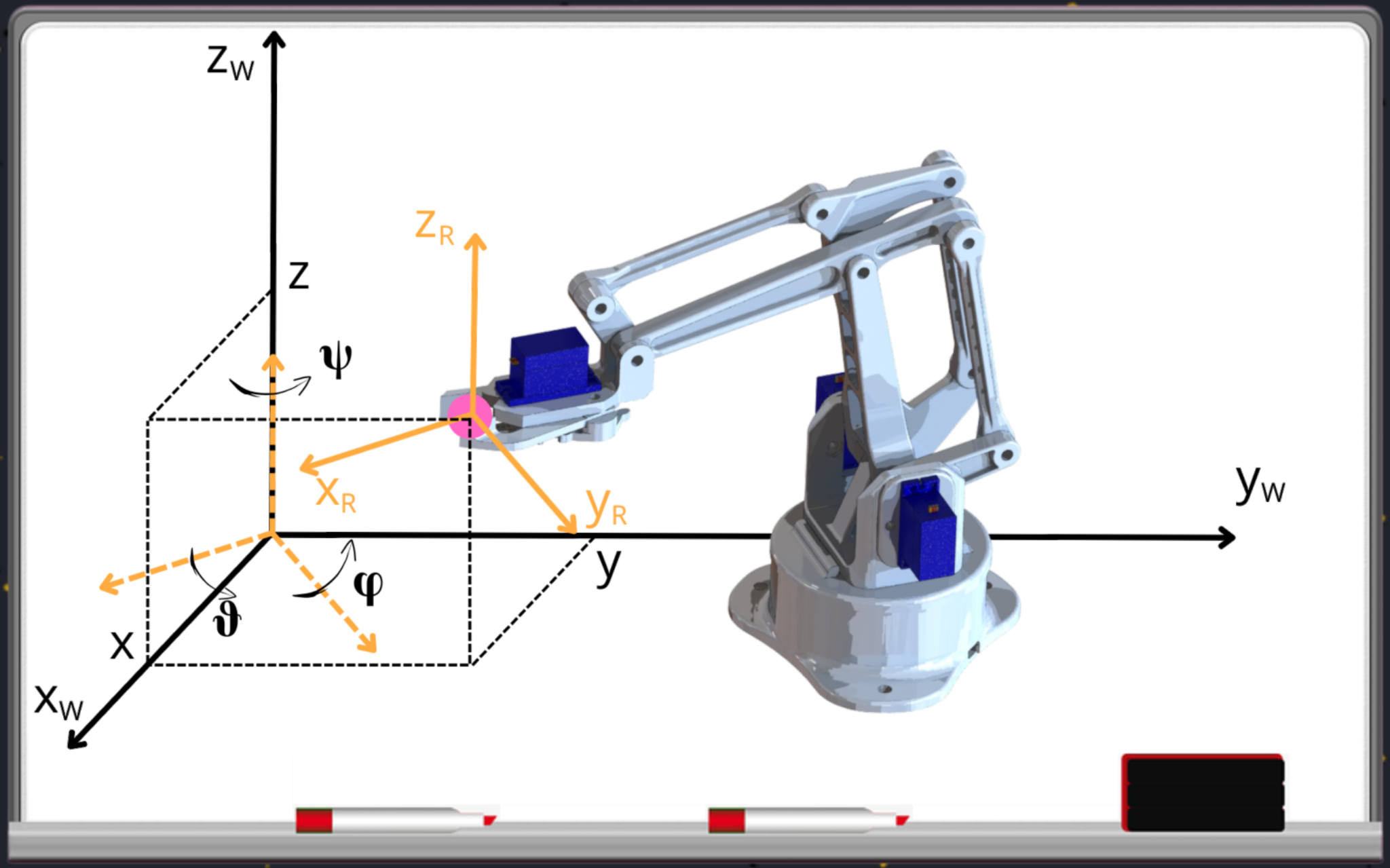


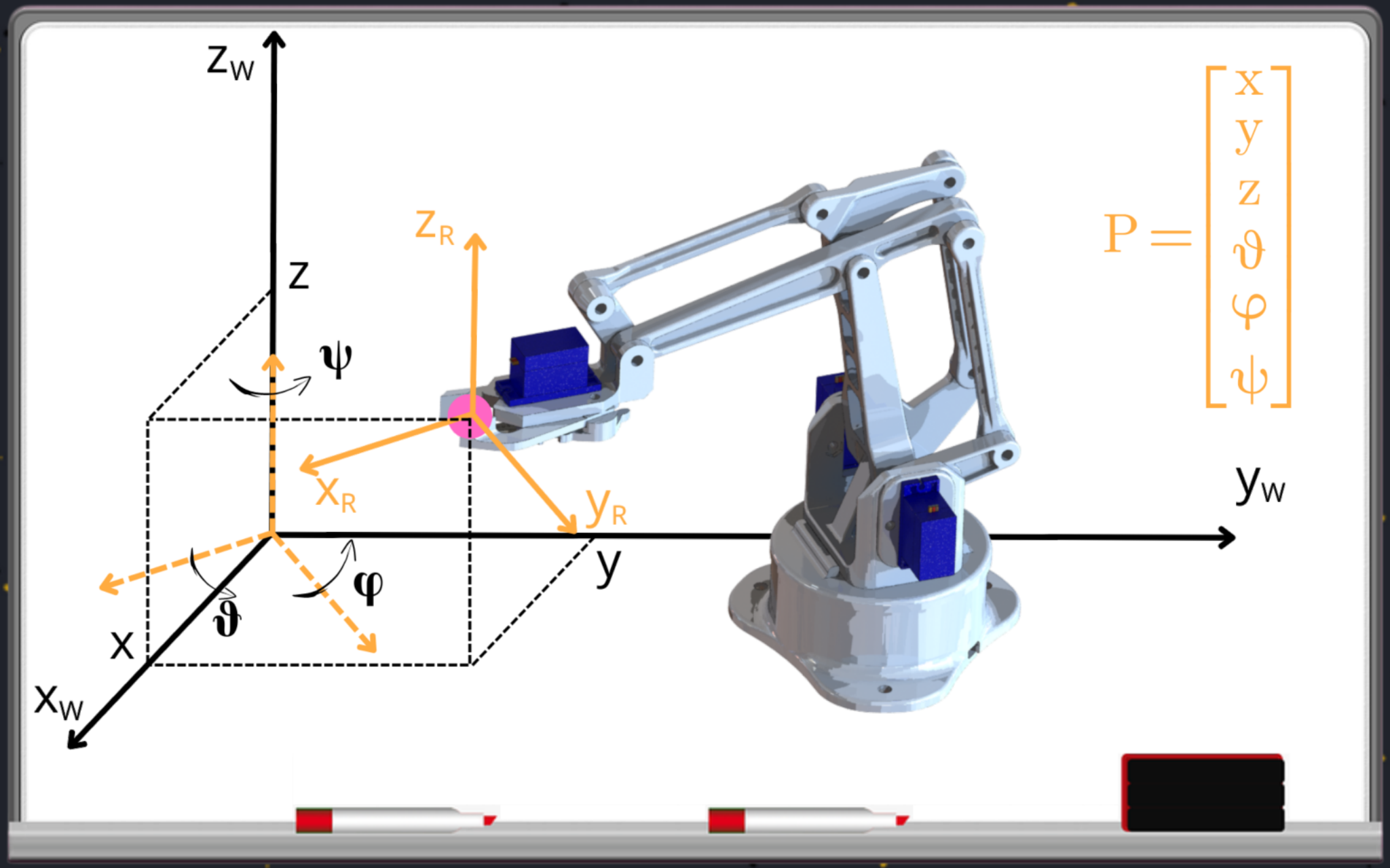

z_w
y_w
x_w

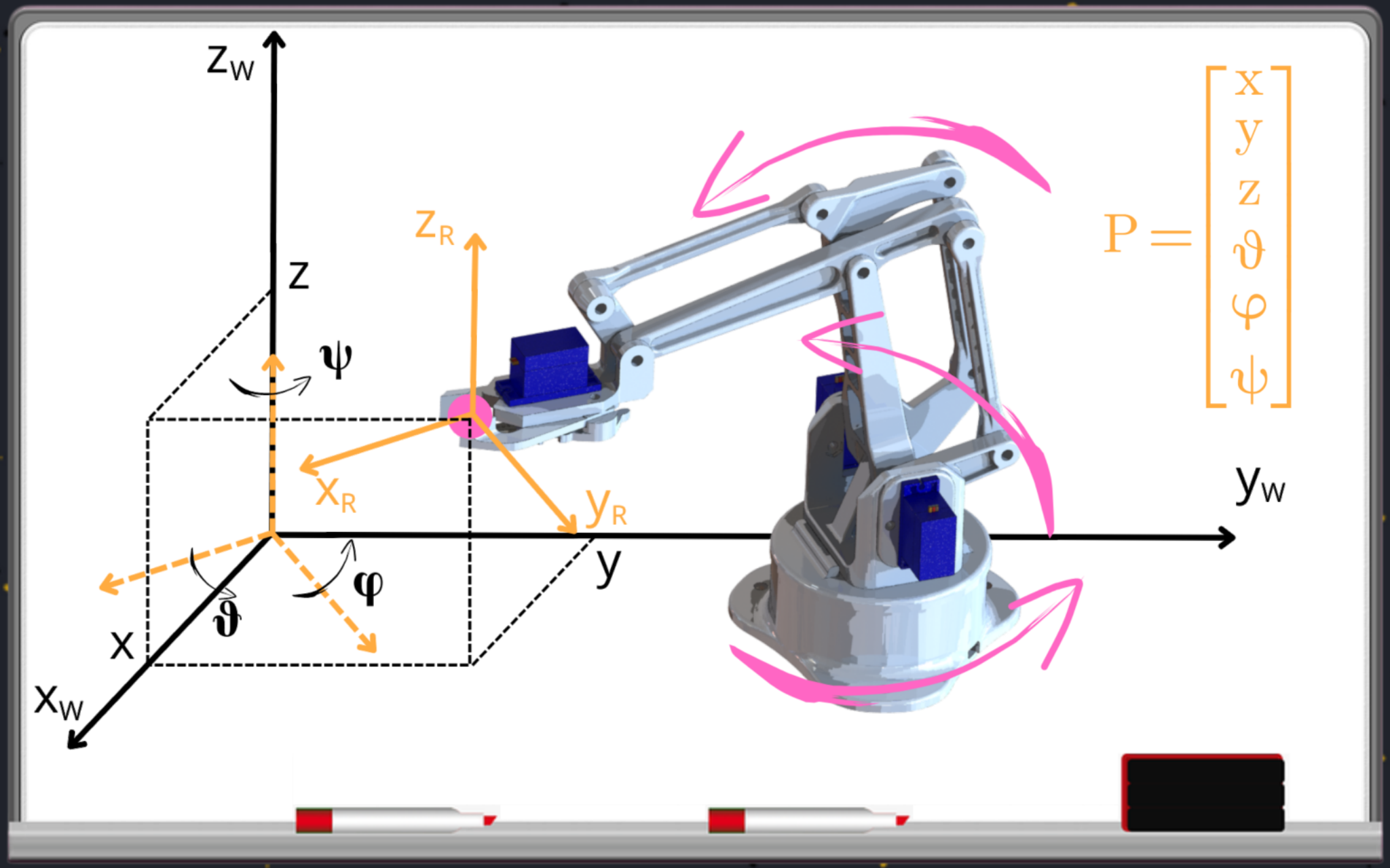

z_W
z_R
x_R
y_R
y_W
x_W

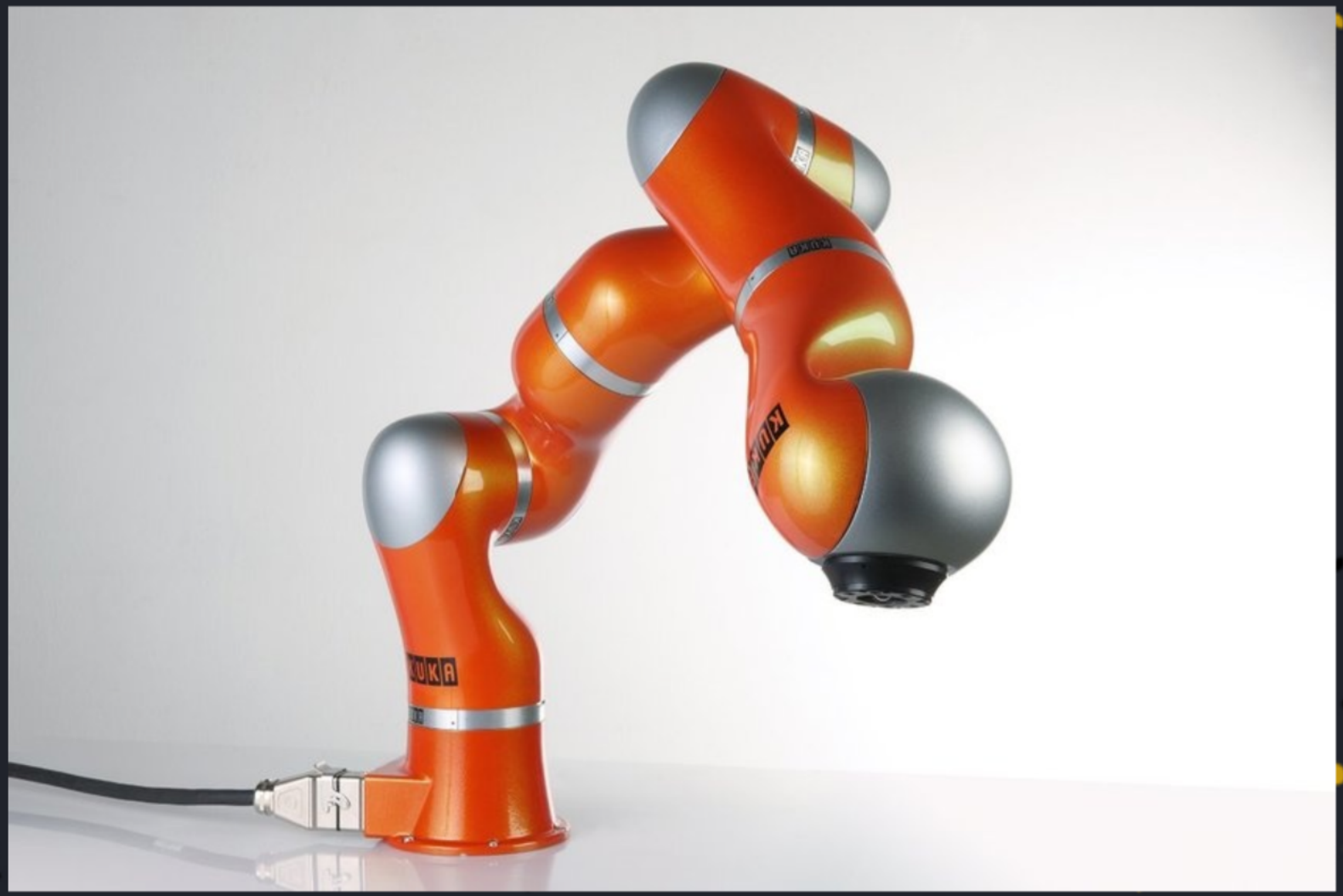

z_W
z
z_R
x_R
y_R
y_W
y
x
x_W
$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

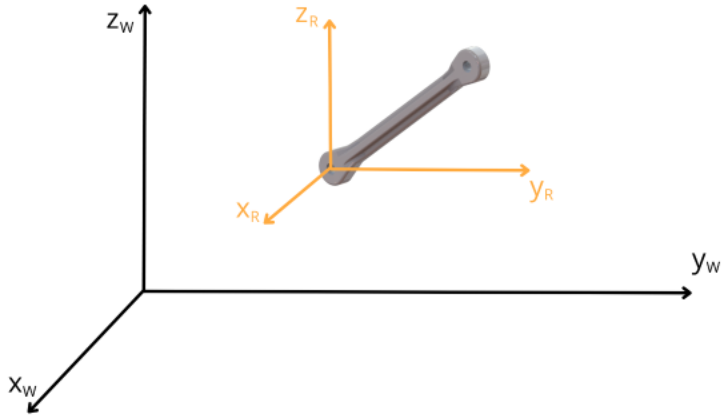

z_W
z
ψ
z_R
x_R
y_R
y_W
y
φ
ϑ
x
x_W
$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$


z_W
z
z_R
ψ
x_R
y_R
y_W
y
φ
ϑ
x
x_W

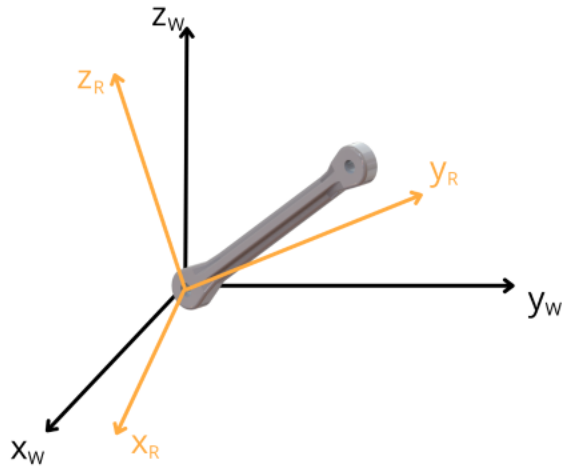

z_W
z
ψ
z_R
x_R
y_R
y_W
y
φ
ϑ
x
x_W
$P = \begin{bmatrix} x \\ y \\ z \\ \vartheta \\ \varphi \\ \psi \end{bmatrix}$


z_W
z
ψ
z_R
x_R
y_R
y_W
y
φ
ϑ
x
x_W
$P = \begin{bmatrix} x \\ y \\ z \\ \vartheta \\ \varphi \\ \psi \end{bmatrix}$

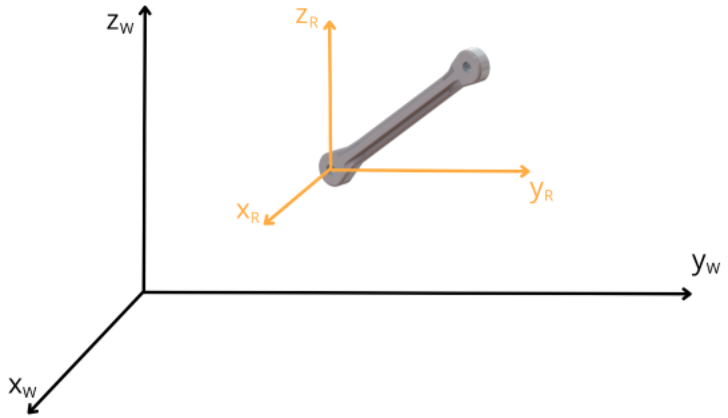

KUKA



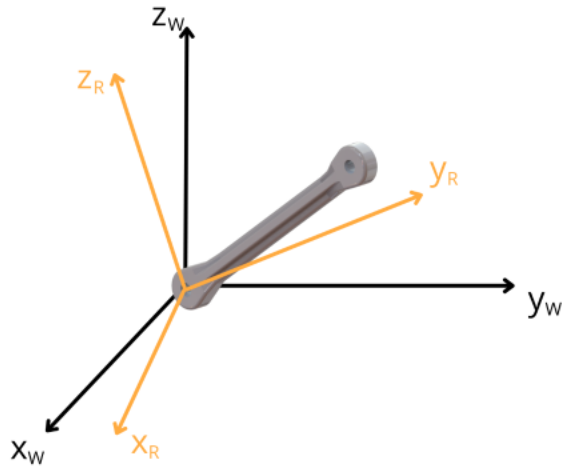
Translation



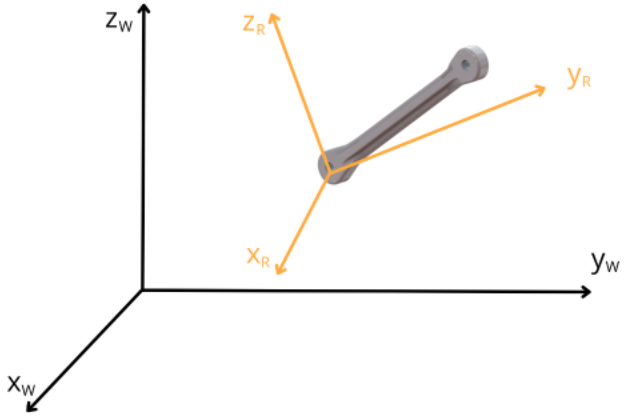
Rotation



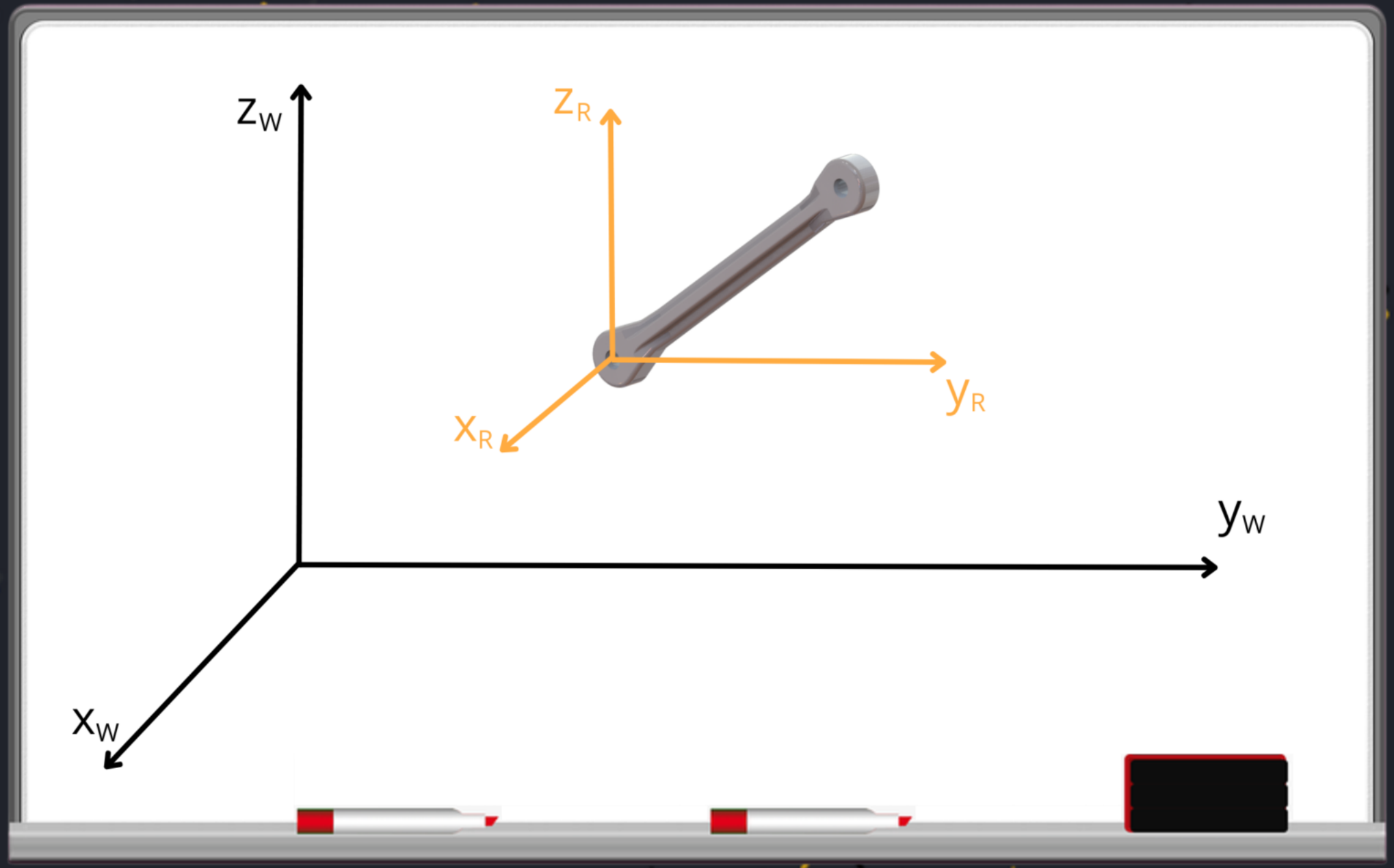
Translation

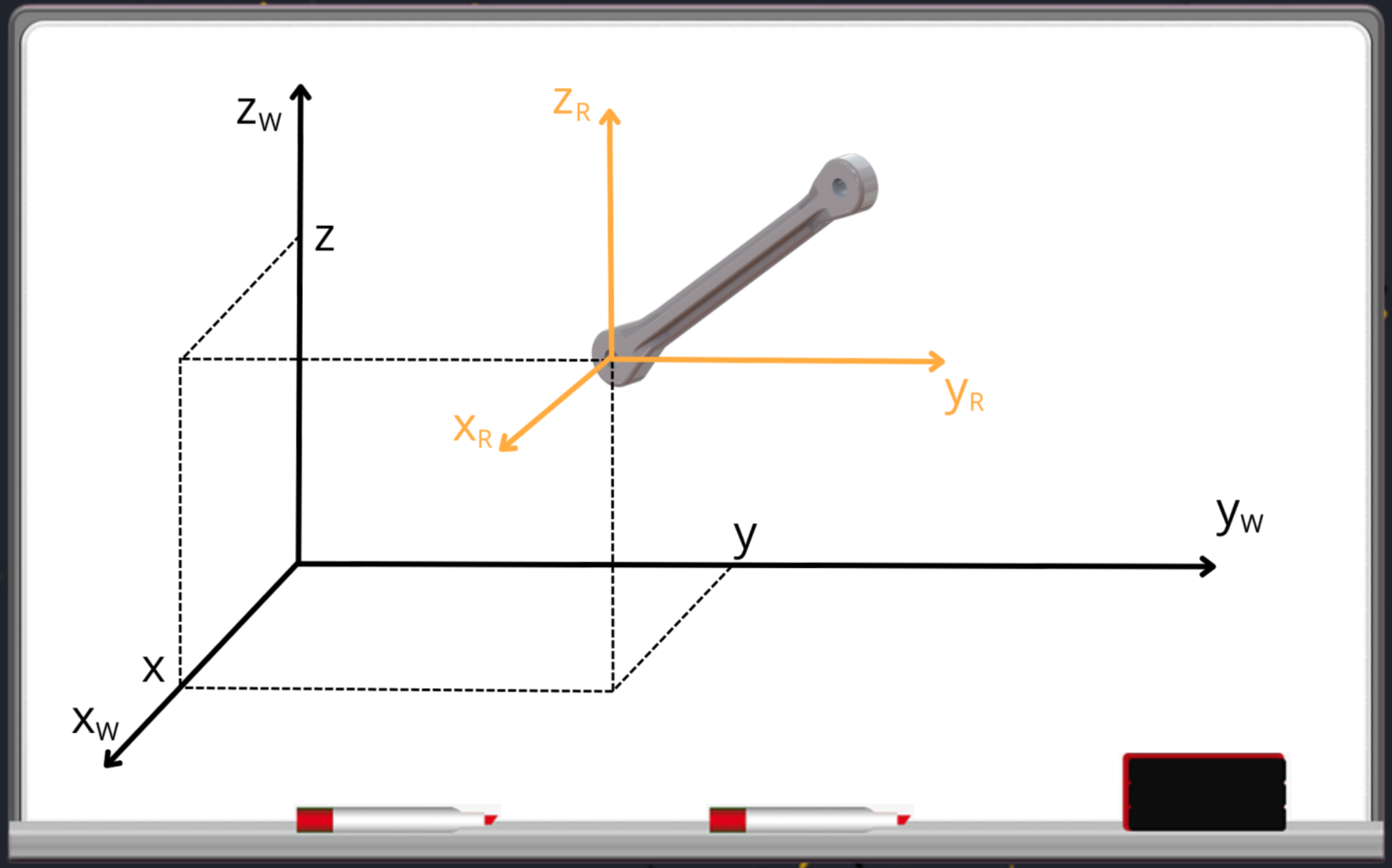


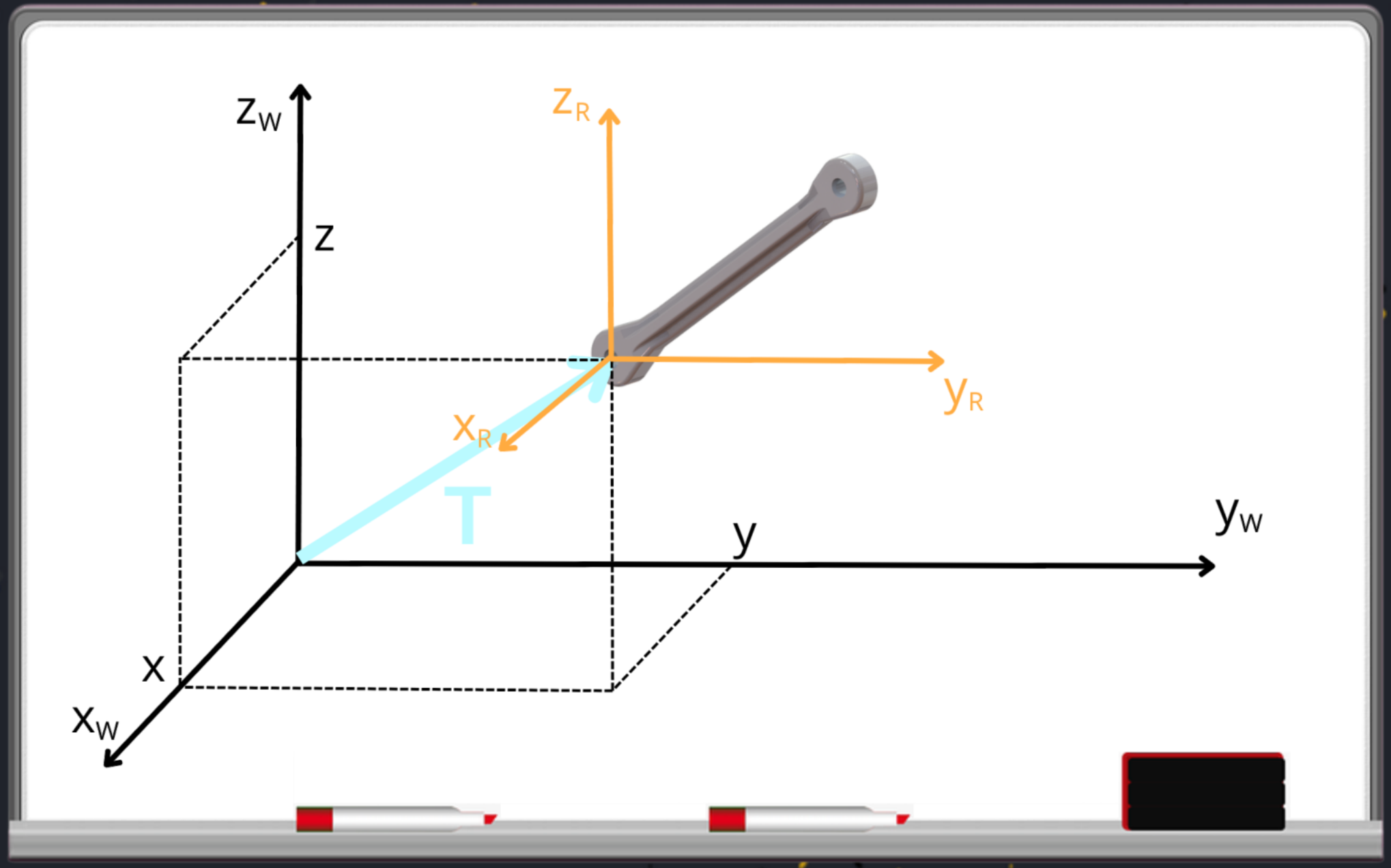
Rotation

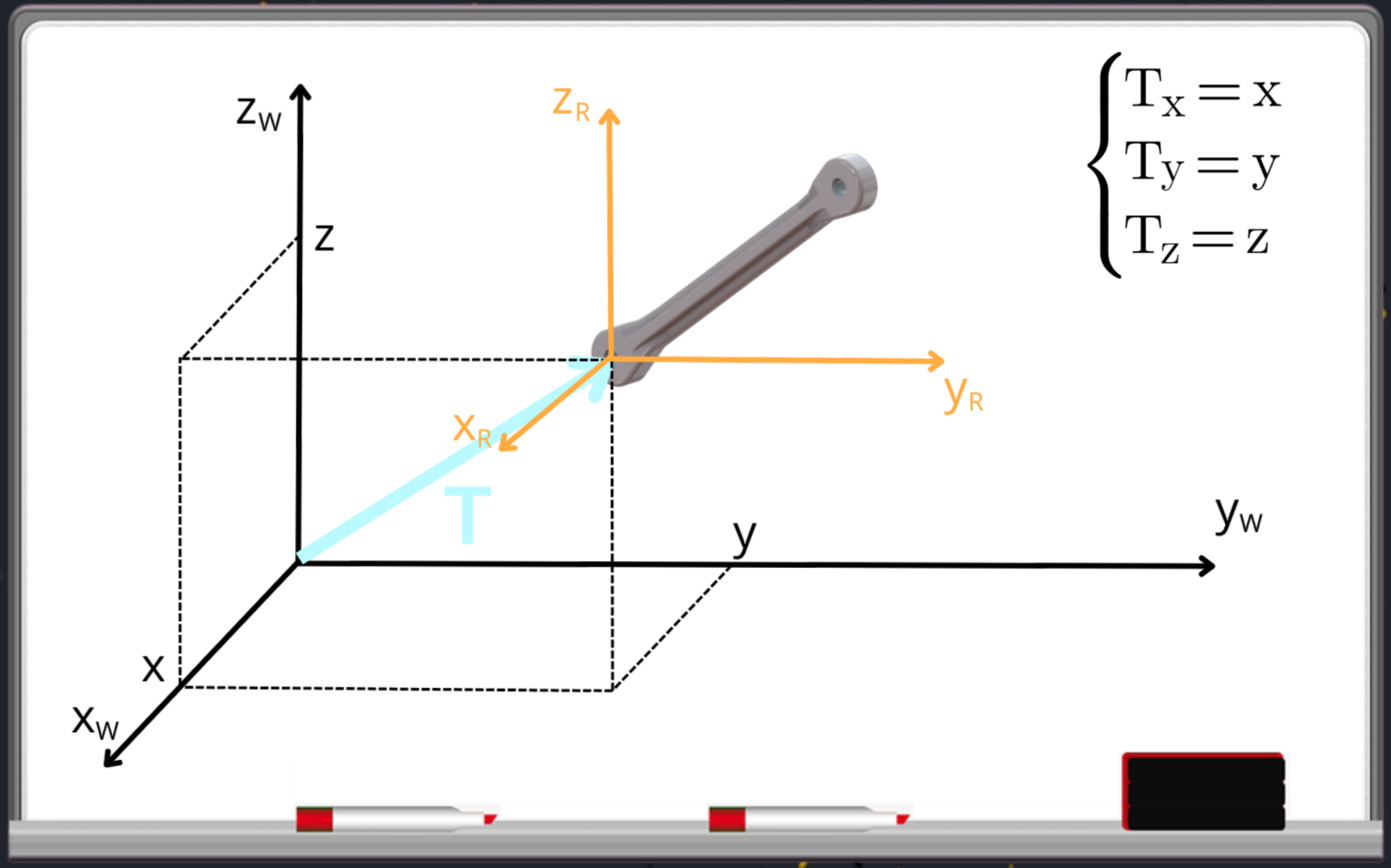


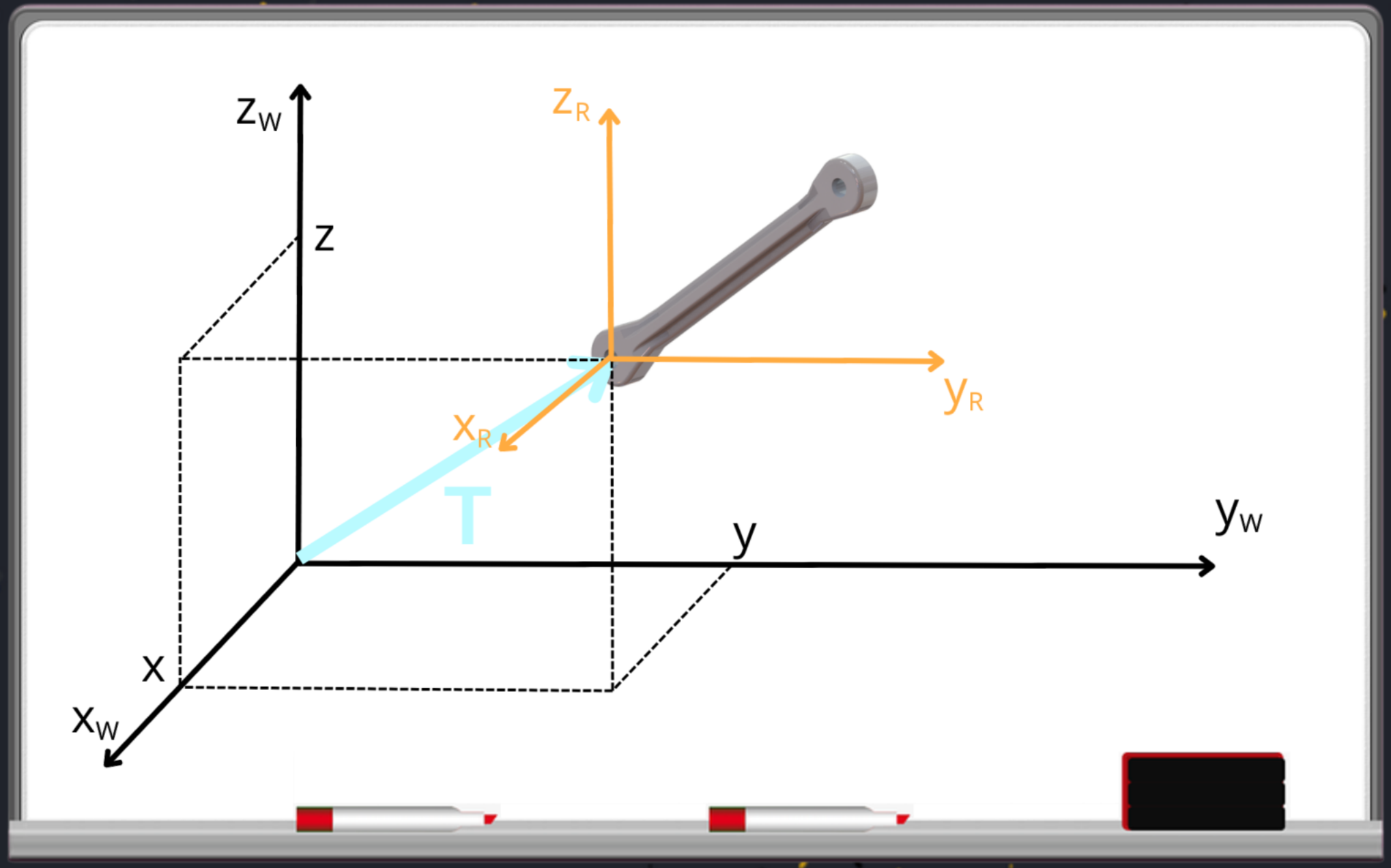
Roto-Translation

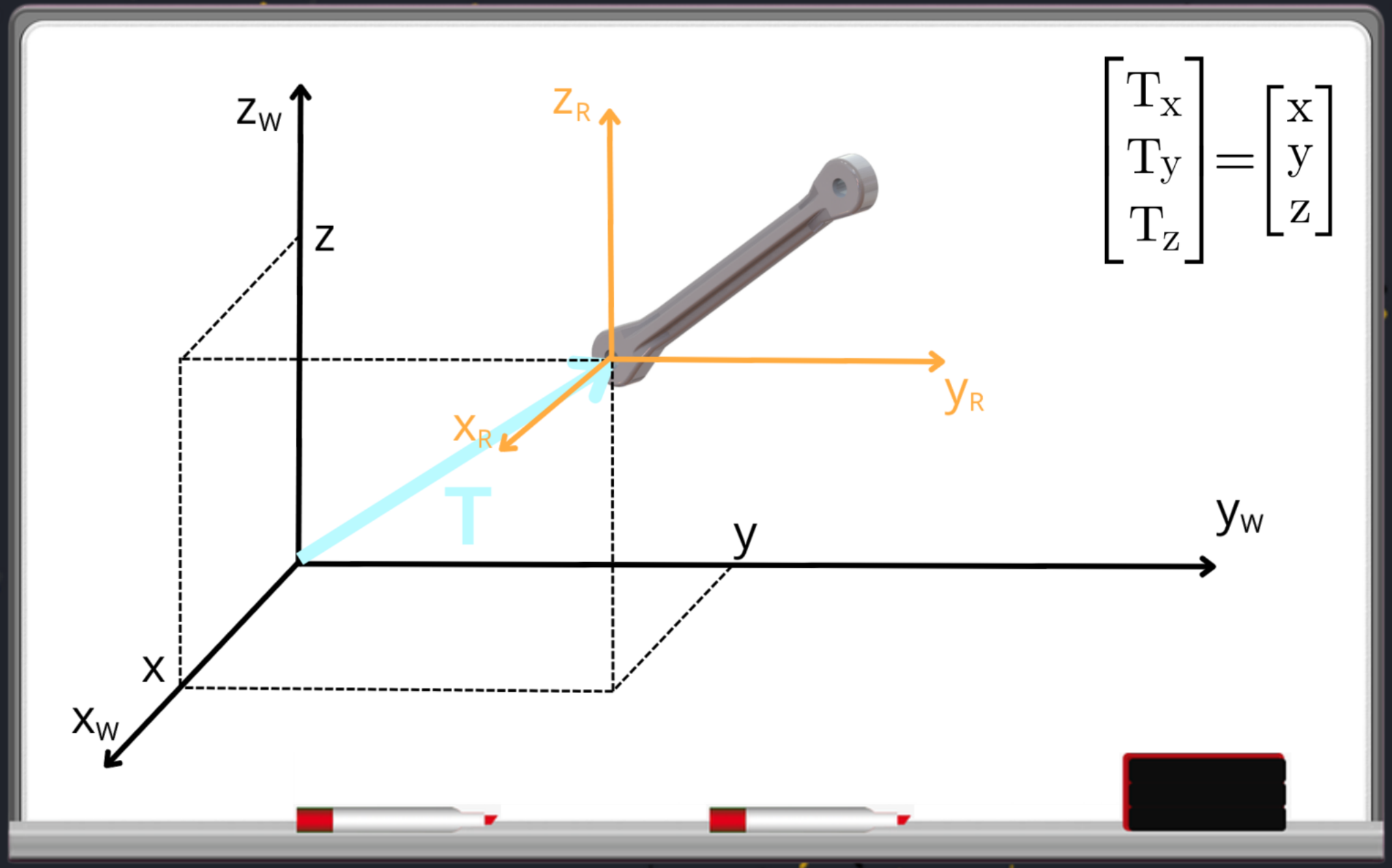

z_W
z_R
y_R
x_R
y_W
x_W

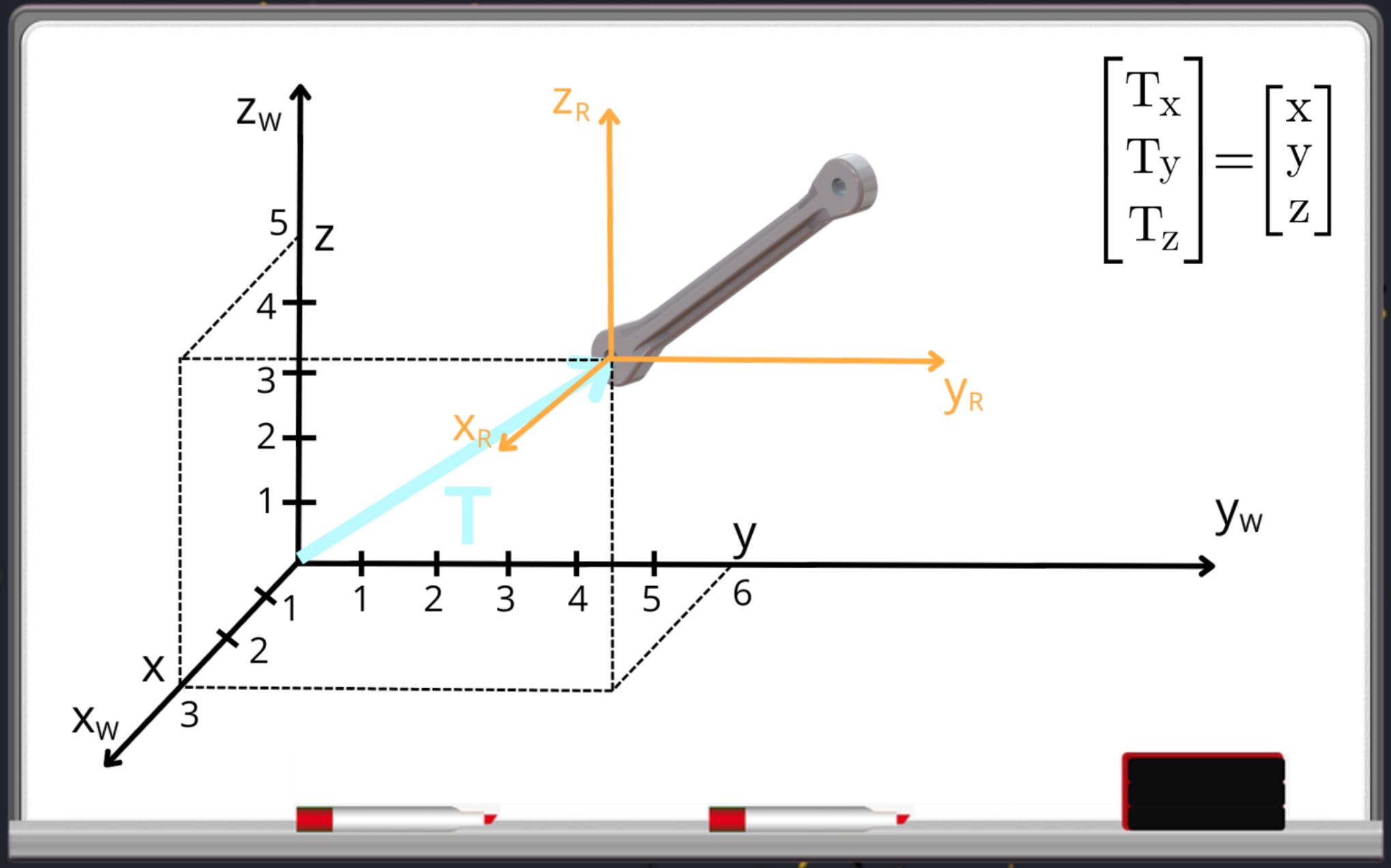

z_W
z
z_R
y_R
x_R
y
y_W
x
x_W

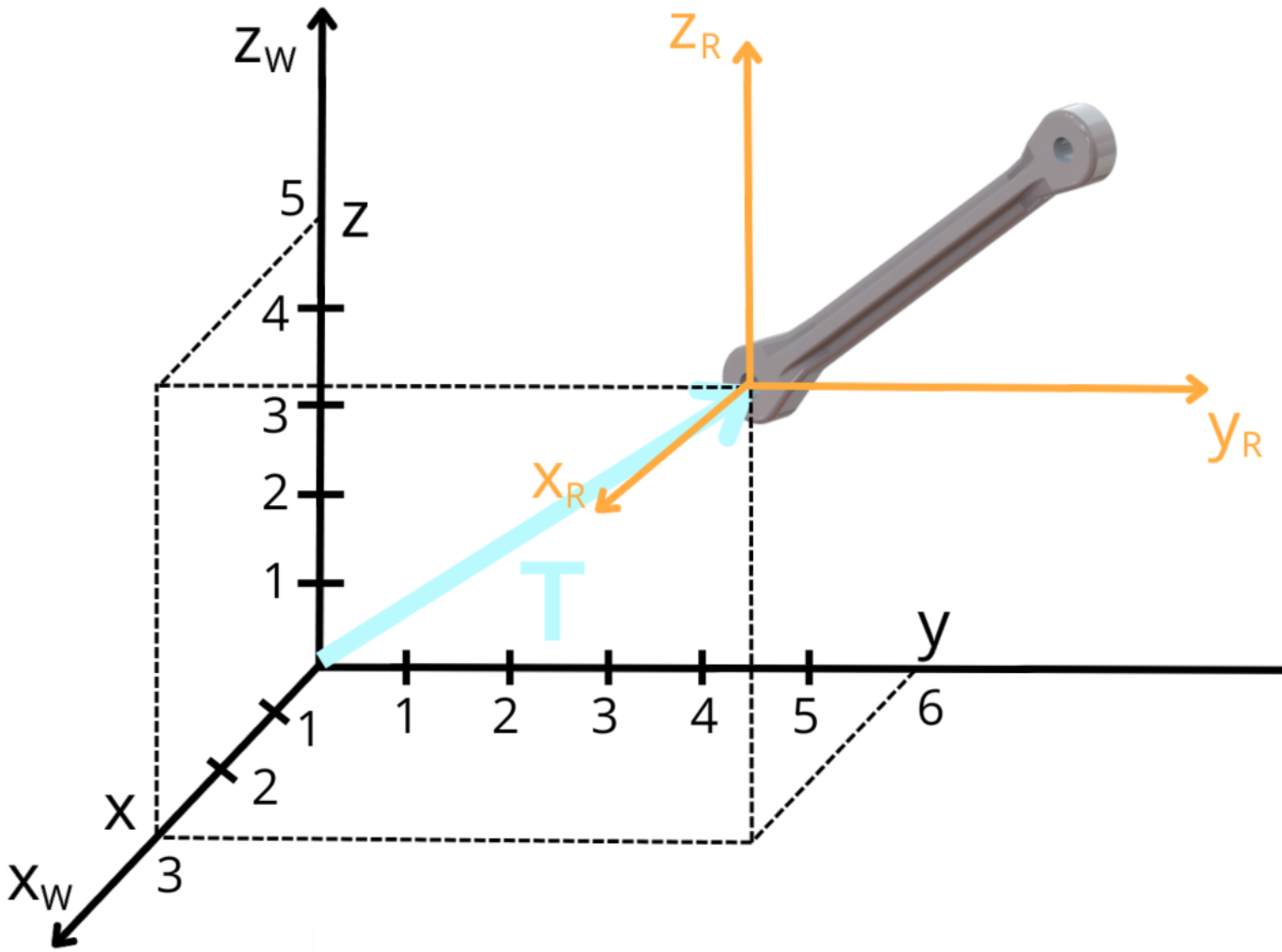

z_W
z
z_R
y_R
x_R
T
y
y_W
x
x_W


z_W
z_R
z
y_R
x_R
T
y_W
y
x
x_W
$\begin{cases} T_x = x \\ T_y = y \\ T_z = z \end{cases}$


z_W
z
z_R
y_R
x_R
T
y
y_W
x
x_W

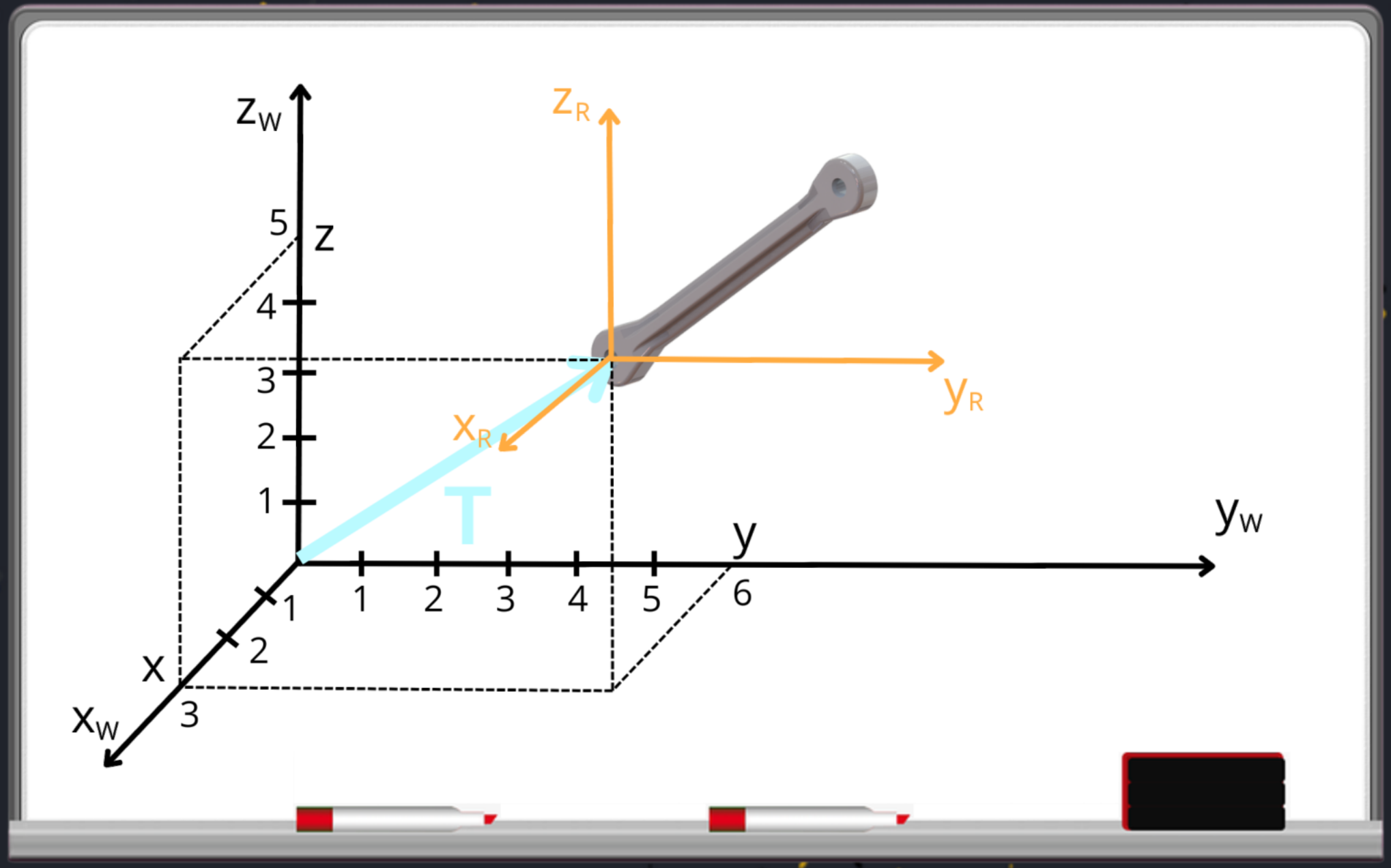

z_W
z
z_R
y_R
x_R
T
y
y_W
x
x_W
$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

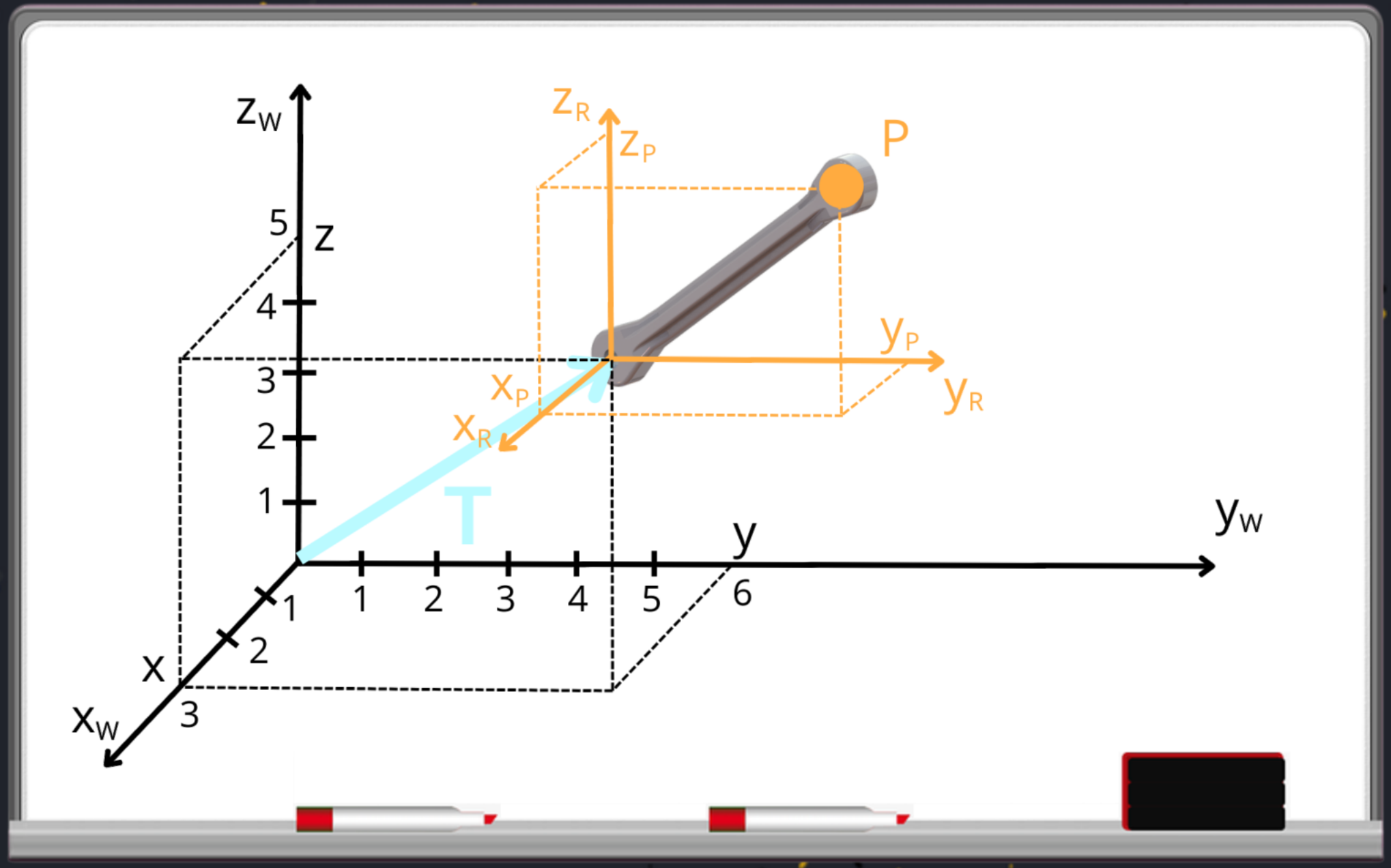

z_W
5
z
4
3
2
1
z_R
y_R
x_R
T
y_W
y
1
2
3
4
5
6
x
1
2
3
x_W
$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$



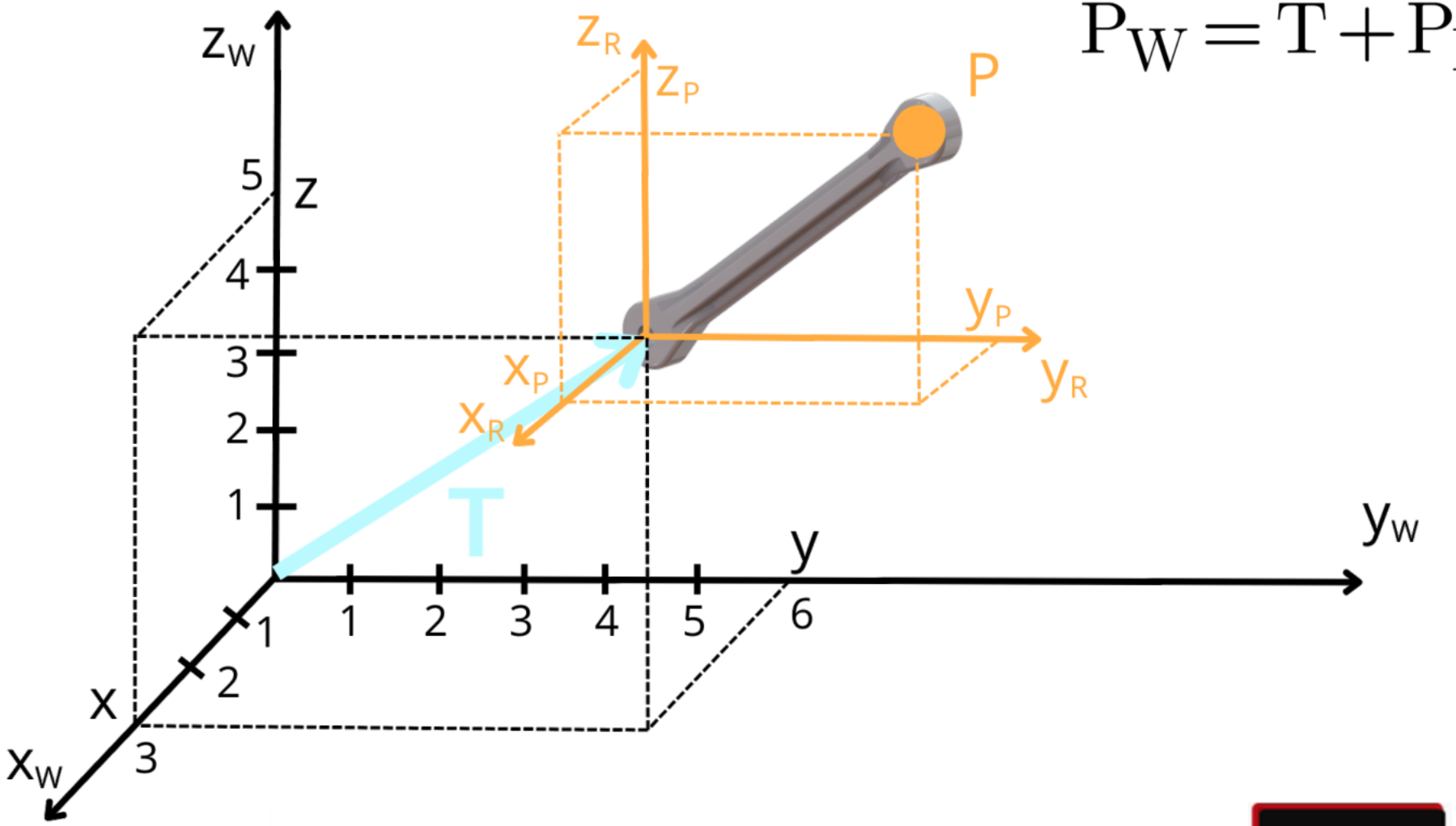
$$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

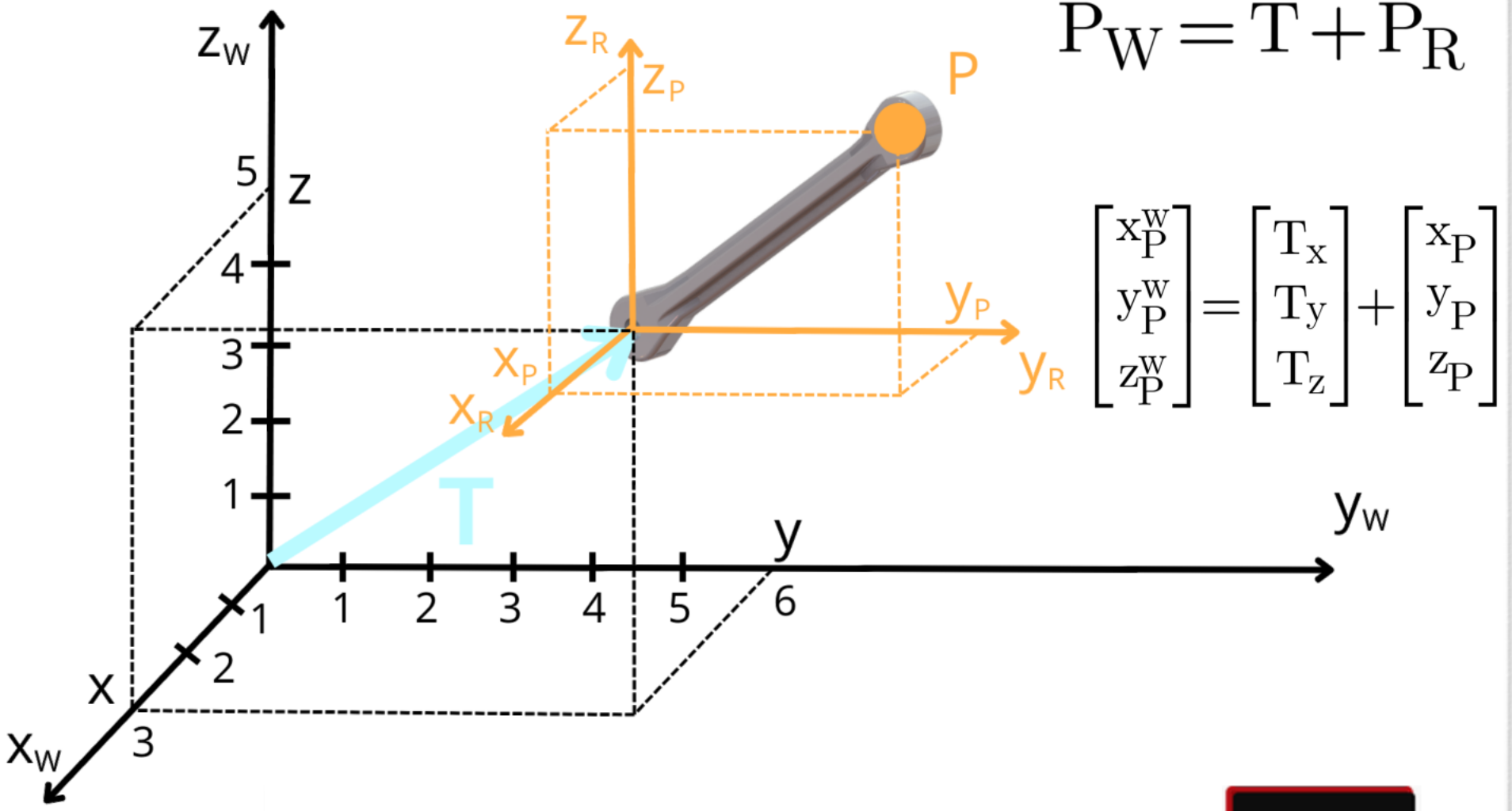
$$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 5 \end{bmatrix}$$


z_W
5
z
4
3
2
1
z_R
y_R
x_R
T
y_W
y
1
2
3
4
5
6
x
1
2
3
x_W


z_W
5
z
4
3
2
1
z_R
z_P
P
y_P
y_R
x_P
x_R
T
y_W
y
1
2
3
4
5
6
x
1
2
3
x_W

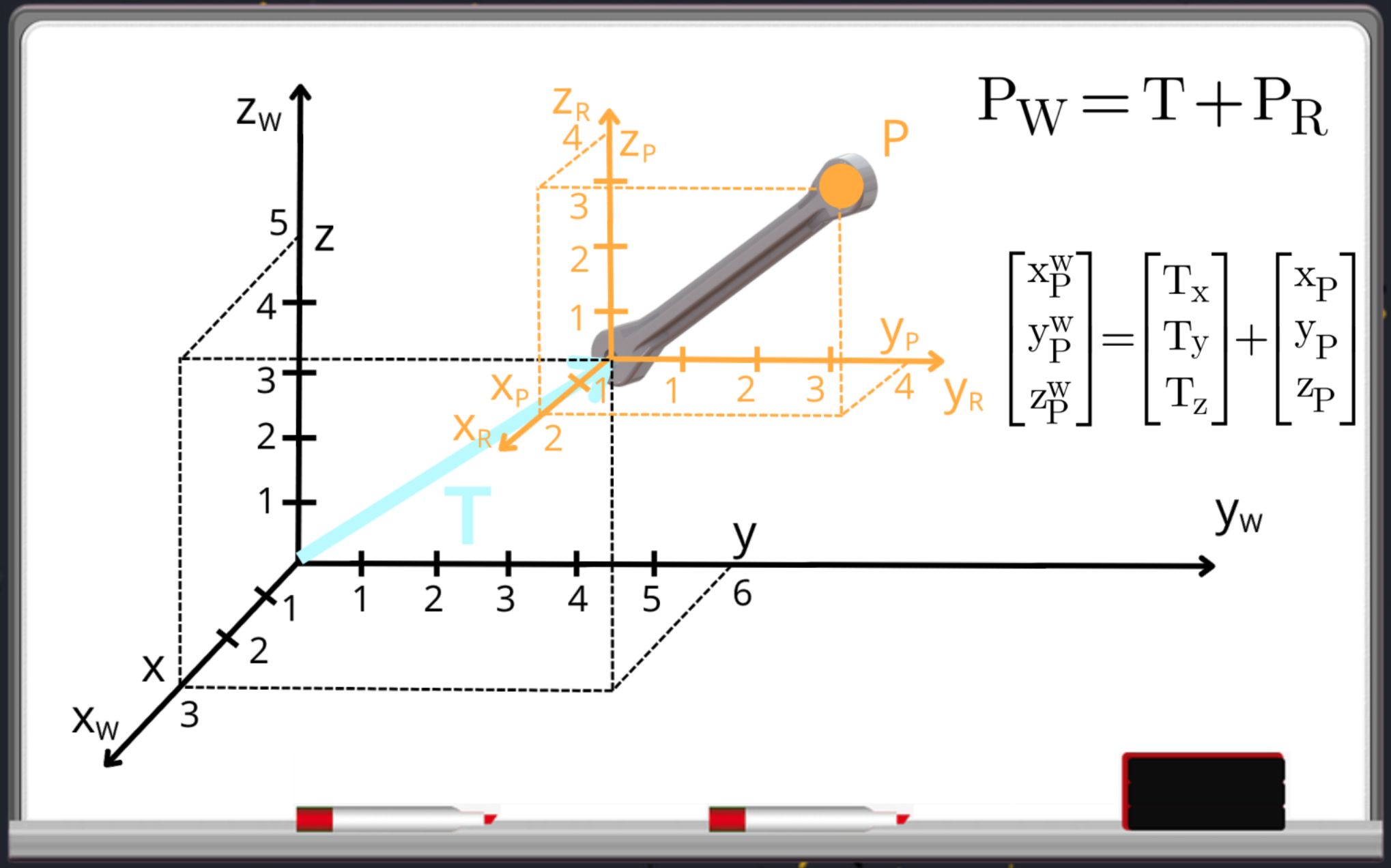
$$P_W = T + P_R$$



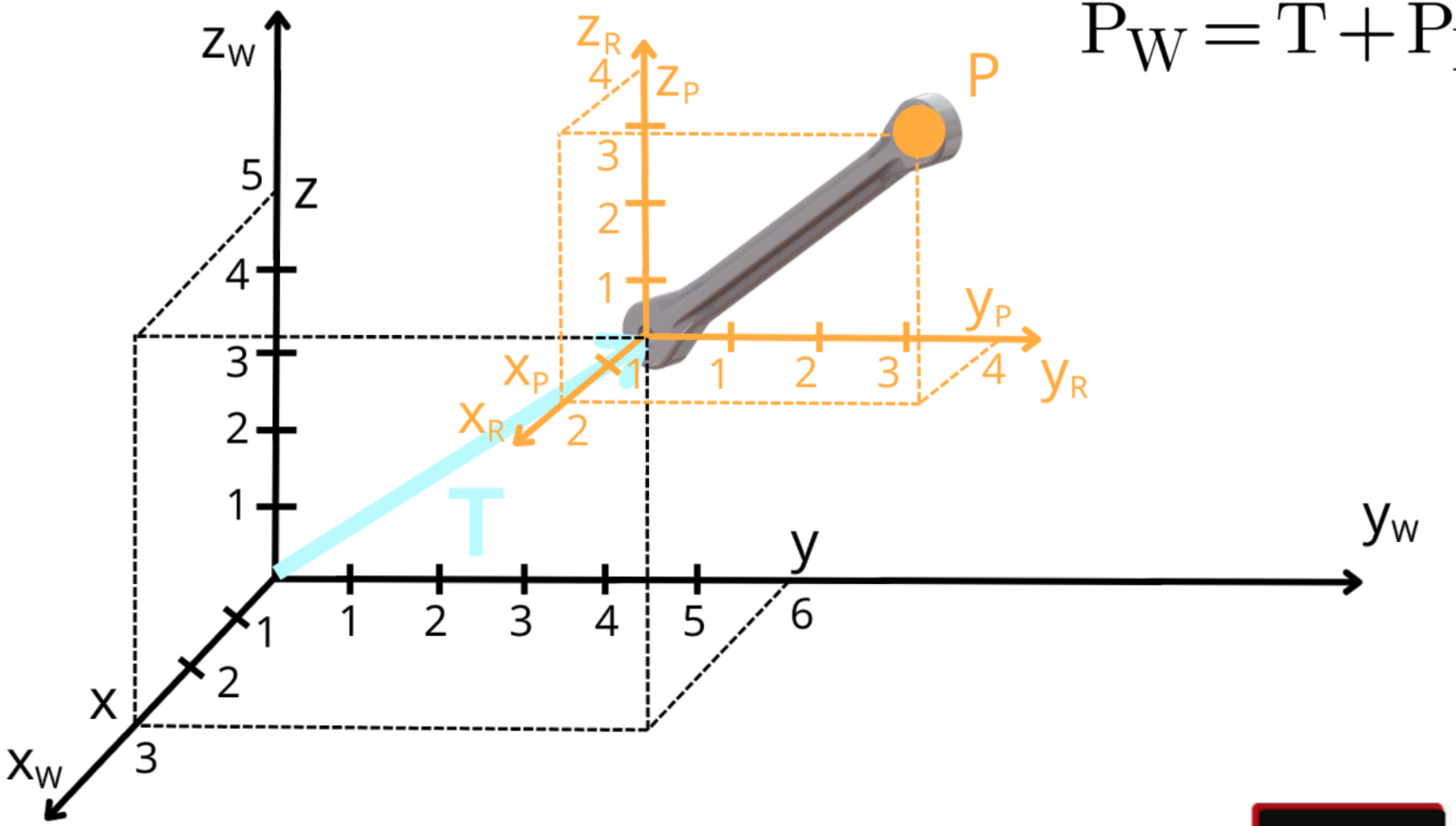


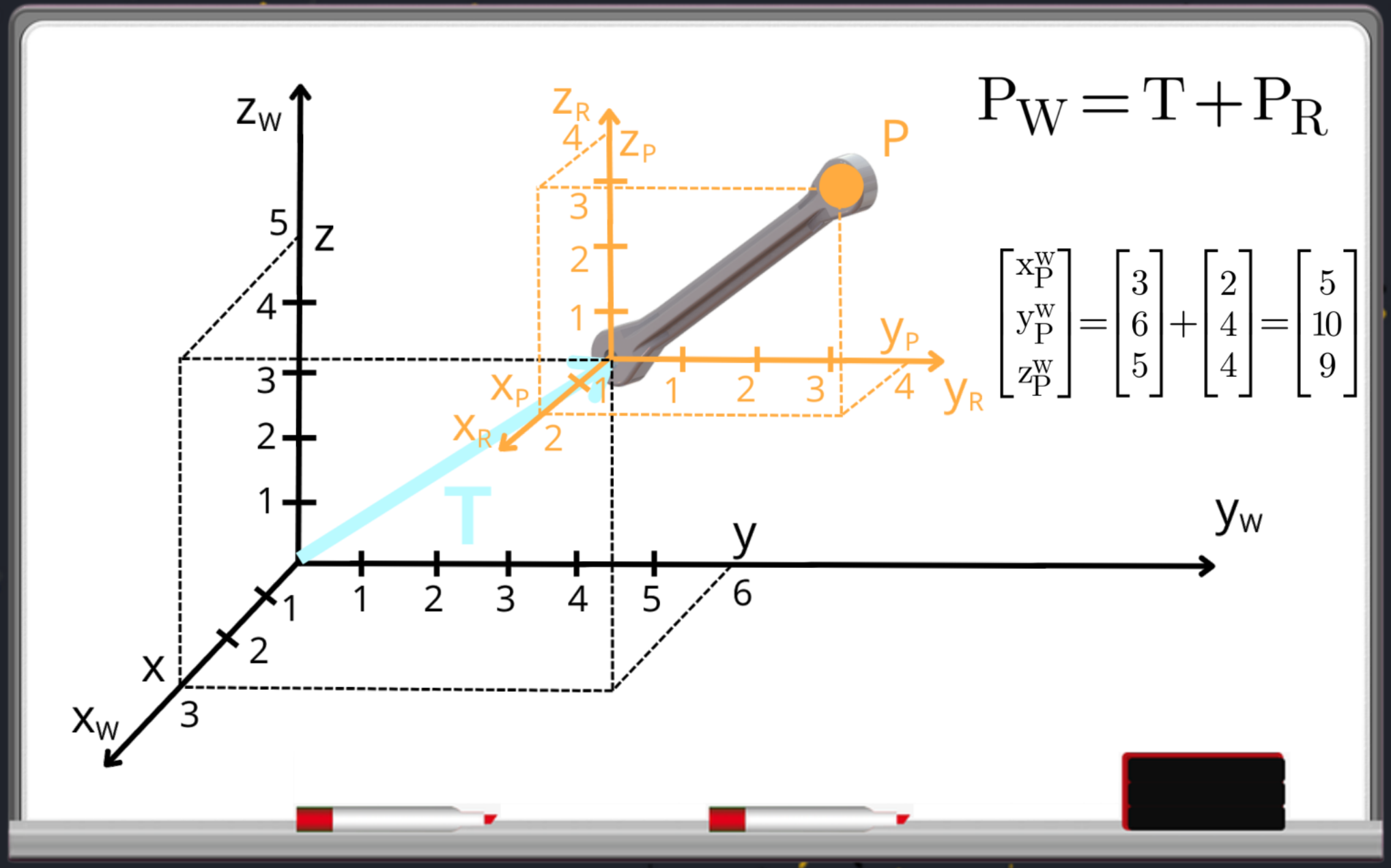
$$P_W = T + P_R$$

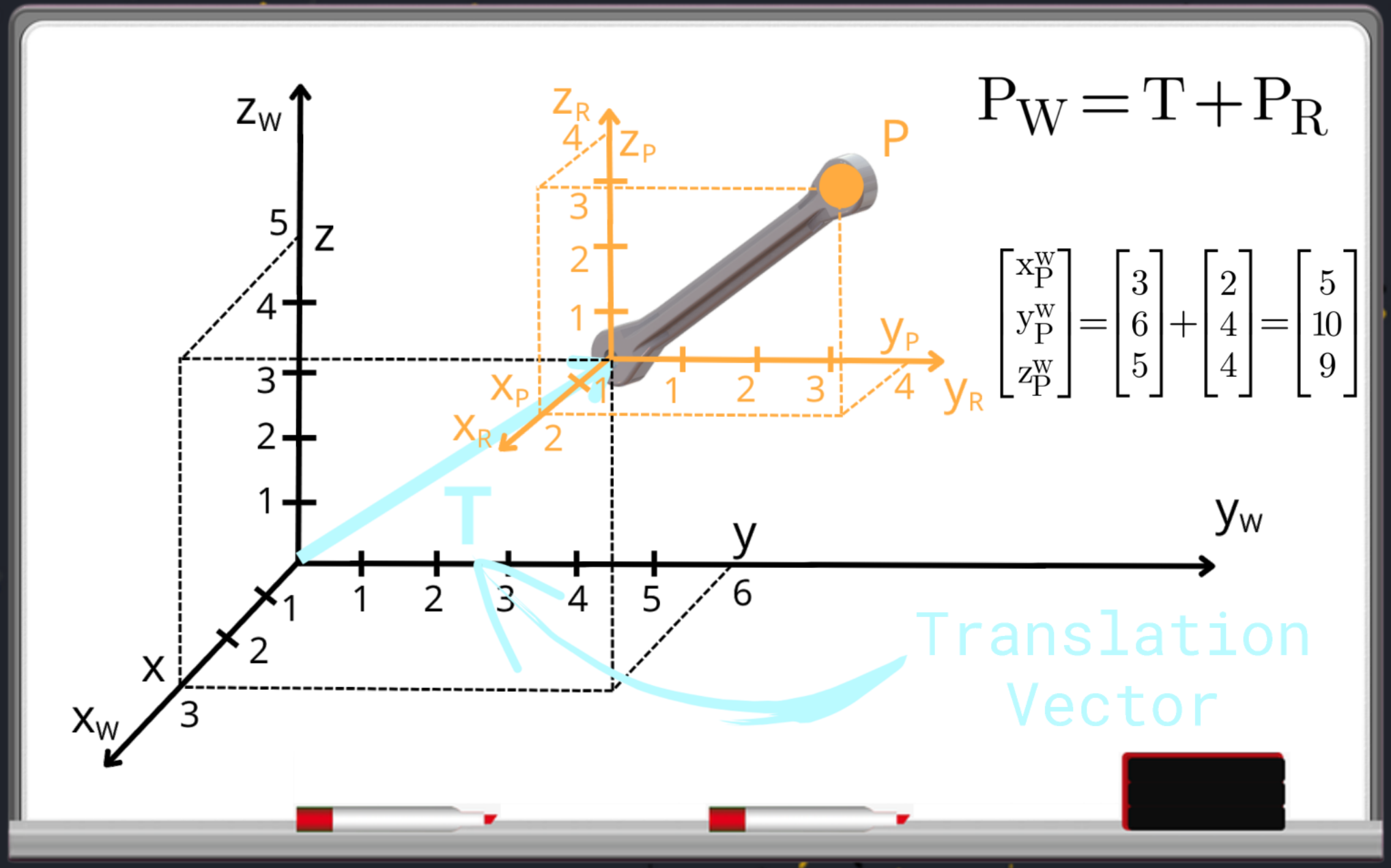
$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}$$

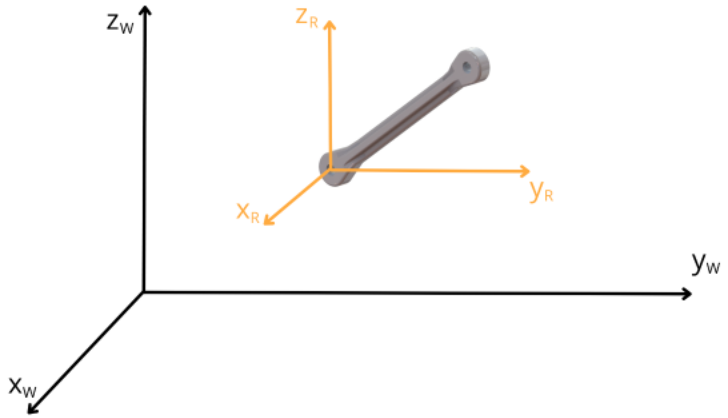

$P_W = T + P_R$
$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}$
z_W
z
z_R
z_P
P
y_P
y_R
x_P
x_R
T
y
y_W
x
x_W

$$P_W = T + P_R$$

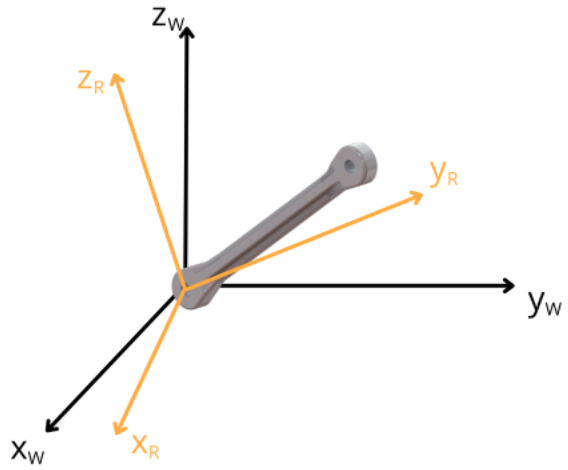



$P_W = T + P_R$
$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 5 \end{bmatrix} + \begin{bmatrix} 2 \\ 4 \\ 4 \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \\ 9 \end{bmatrix}$
z_W
z
z_R
z_P
P
y_P
y_R
x_P
x_R
T
y_W
y
x
x_W

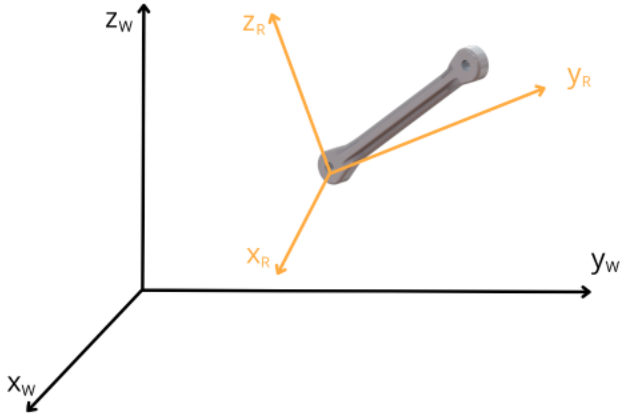

$P_W = T + P_R$
$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 5 \end{bmatrix} + \begin{bmatrix} 2 \\ 4 \\ 4 \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \\ 9 \end{bmatrix}$
z_W
z_R
z_P
P
z
y_P
y_R
x_P
x_R
T
y_W
y
x
x_W
Translation Vector



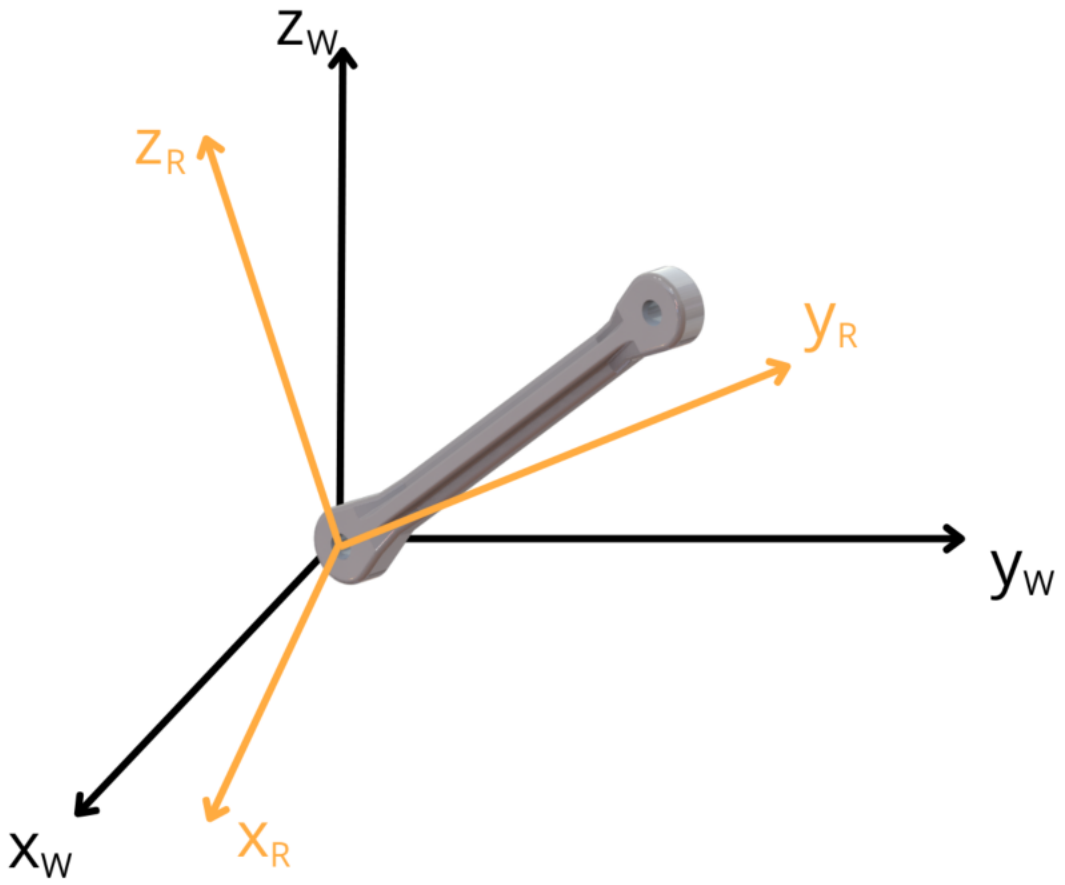
Translation

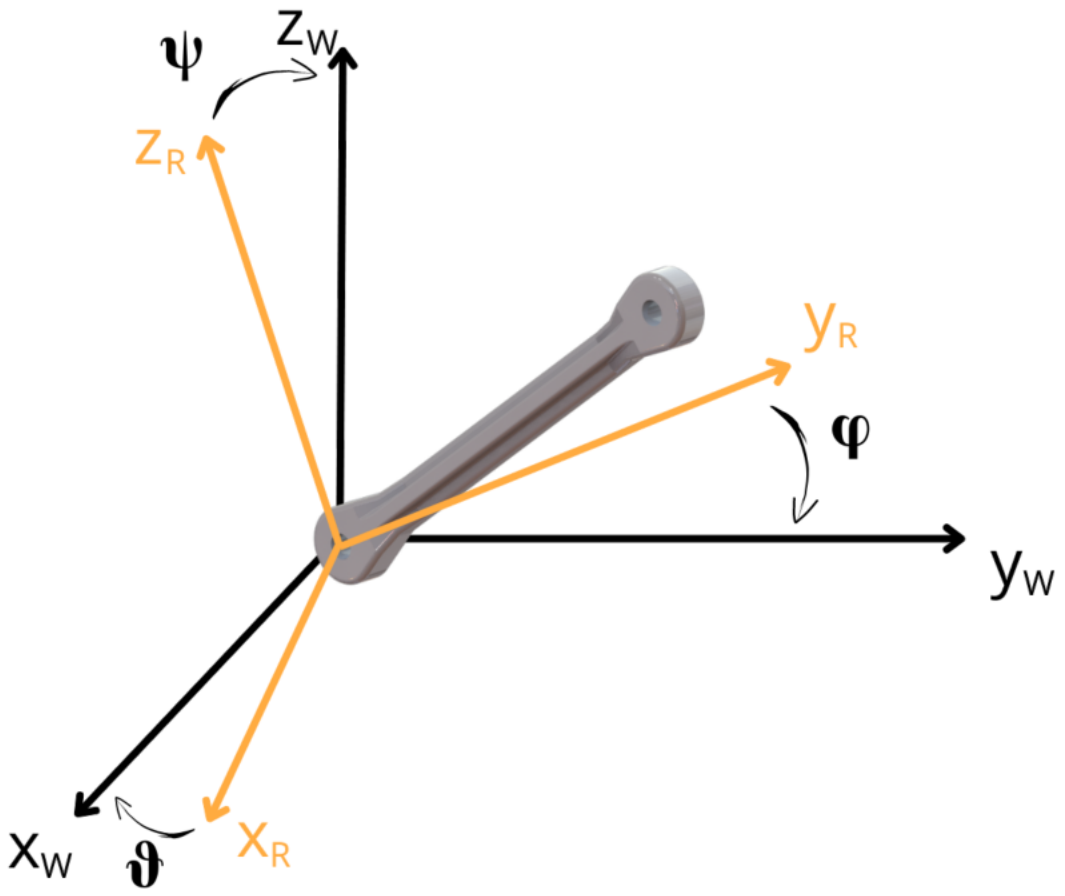


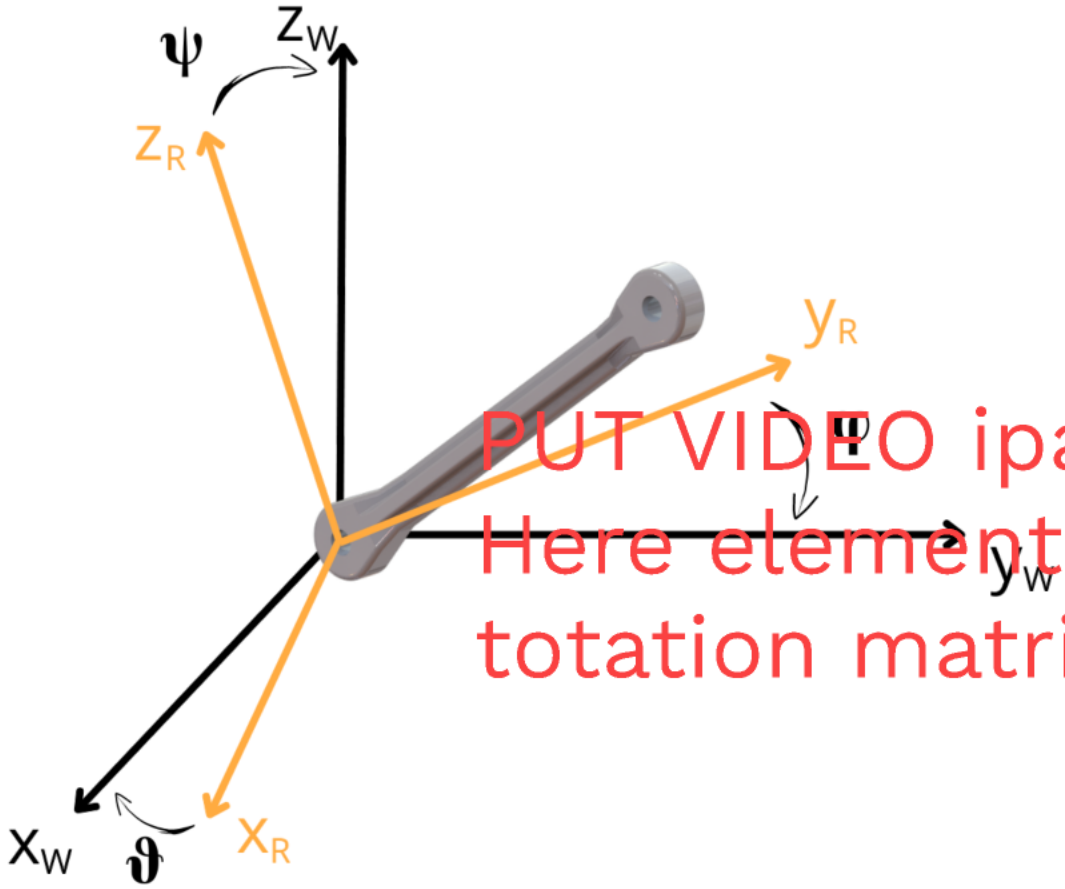
Rotation



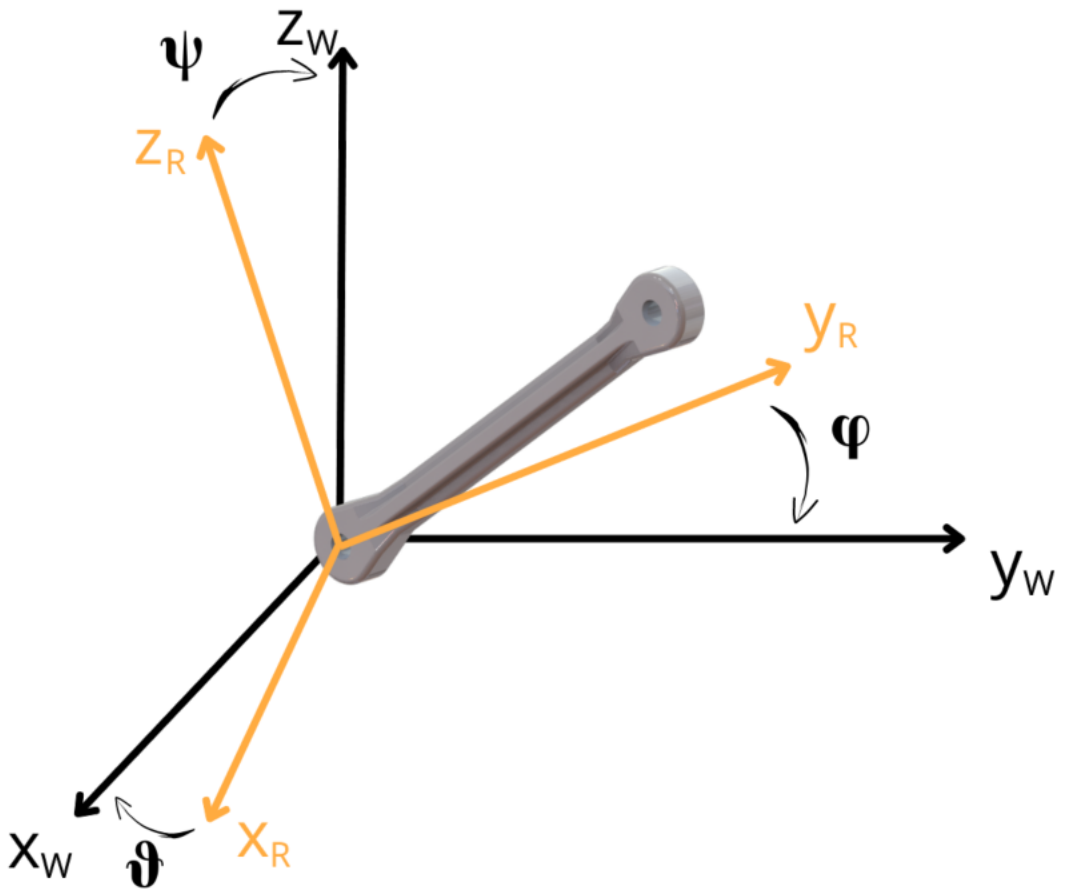
Roto-Translation

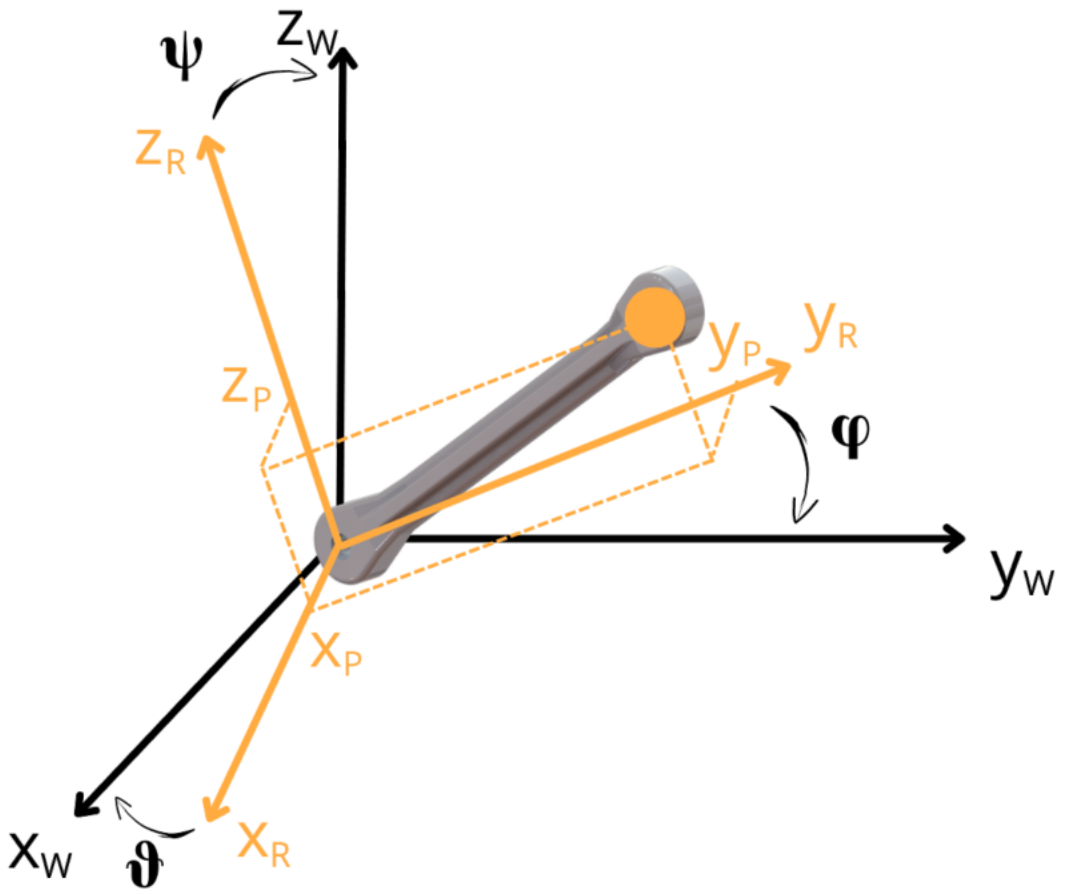

z_W
z_R
y_R
y_W
x_W
x_R

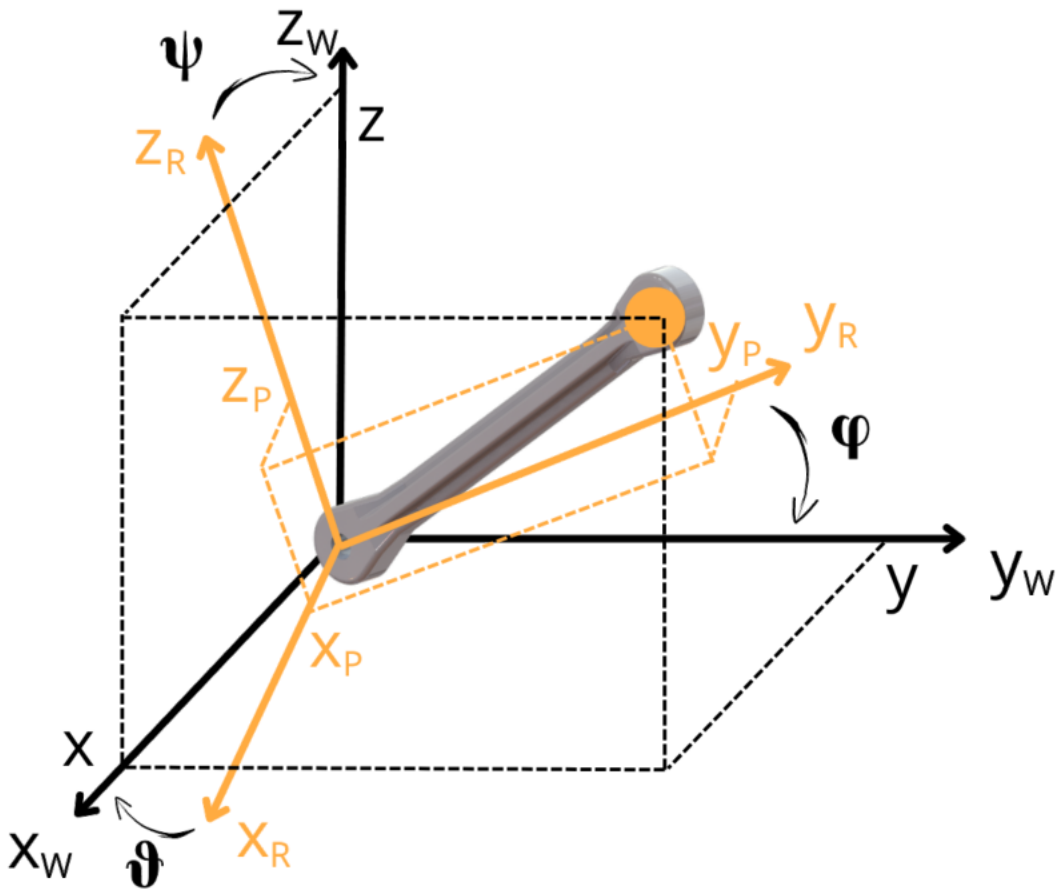

ψ
z_W
z_R
y_R
φ
y_W
x_W
ϑ
x_R

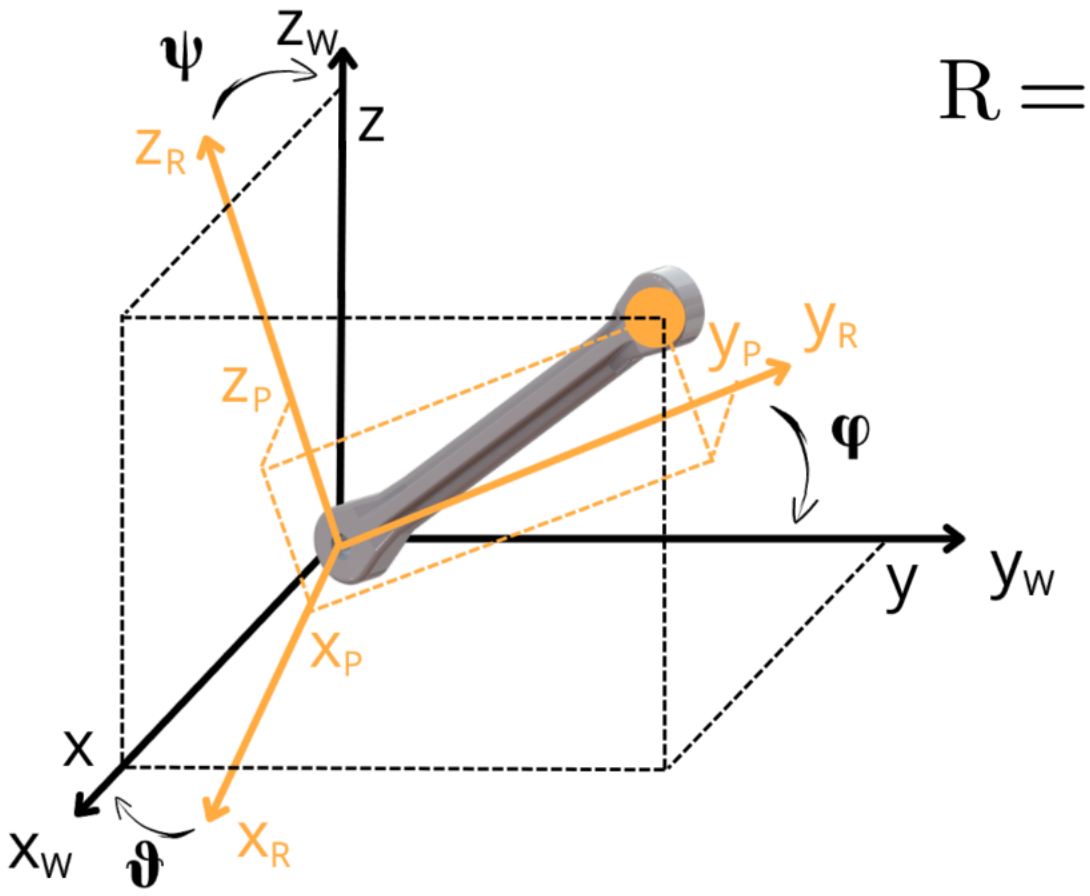


PUT VIDEO ipad
Here elementary
totation matrices

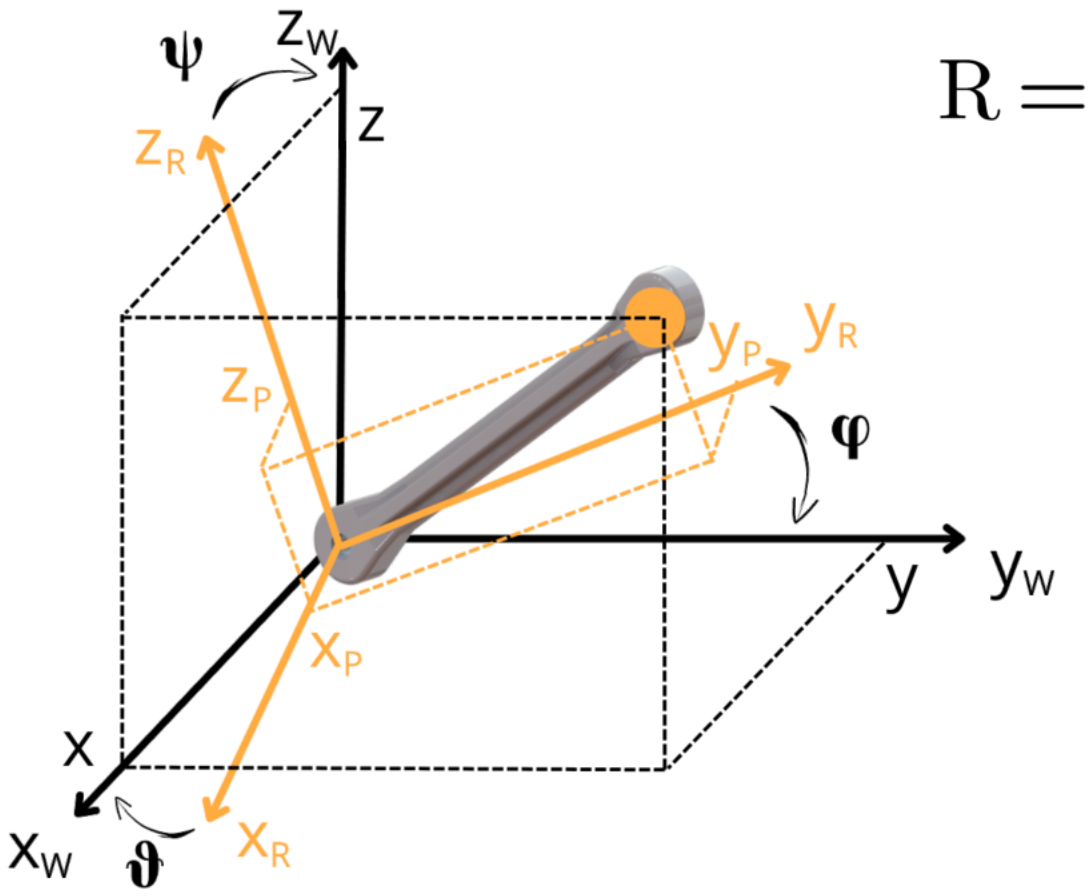

ψ
z_W
z_R
y_R
φ
y_W
x_W
ϑ
x_R


ψ
z_W
z_R
z_P
y_P
y_R
φ
y_W
x_P
x_W
ϑ
x_R


ψ
z_W
z
z_R
z_P
y_P
y_R
φ
y
y_W
x_P
x
x_W
ϑ
x_R

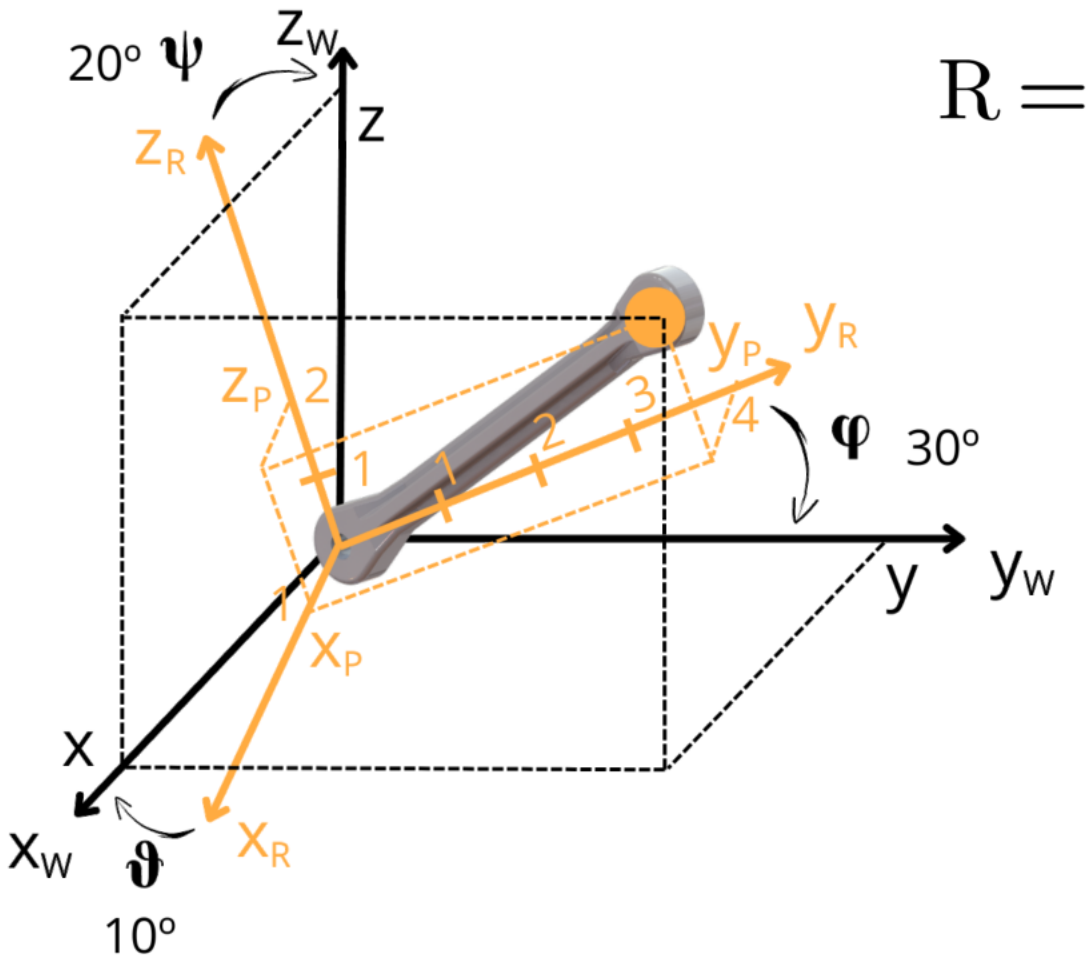


$$R = R_z(\psi) \cdot R_y(\varphi) \cdot R_x(\vartheta)$$



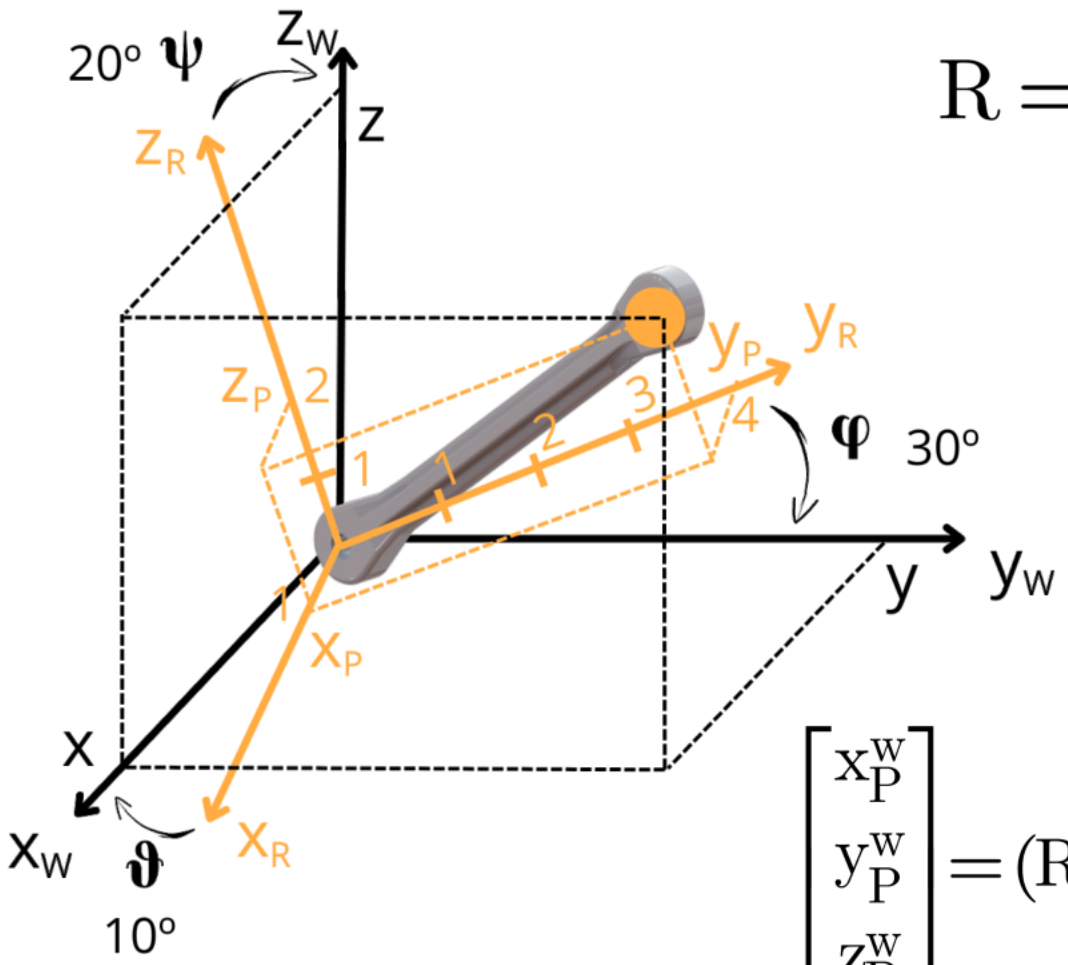
$$R = R_z(\psi) \cdot R_y(\varphi) \cdot R_x(\vartheta)$$

$$\begin{bmatrix} x_P^w \\ y_P^w \\ z_P^w \end{bmatrix} = R \cdot \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}$$



$$R = R_z(\psi) \cdot R_y(\varphi) \cdot R_x(\vartheta)$$

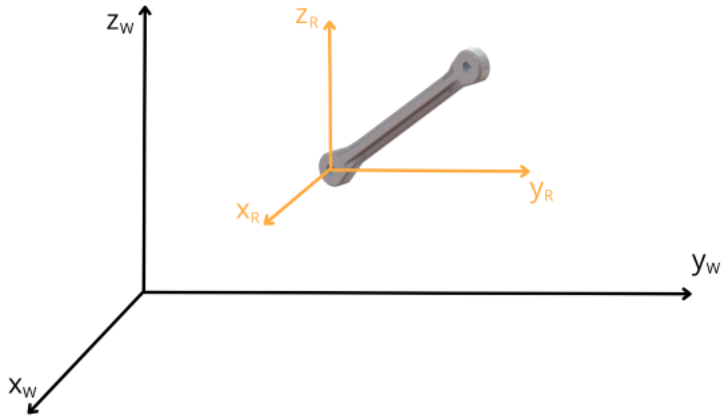
$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = R \cdot \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}$$



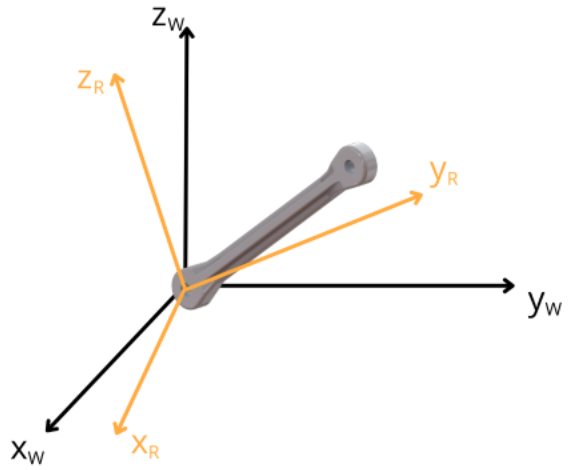
$$R = R_z(\psi) \cdot R_y(\varphi) \cdot R_x(\vartheta)$$

$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = R \cdot \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}$$

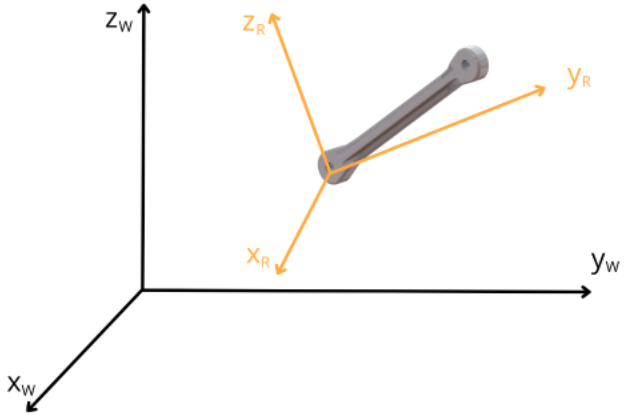
$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = (R_z(20^\circ) \cdot R_y(30^\circ) \cdot R_x(10^\circ)) \cdot \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix}$$



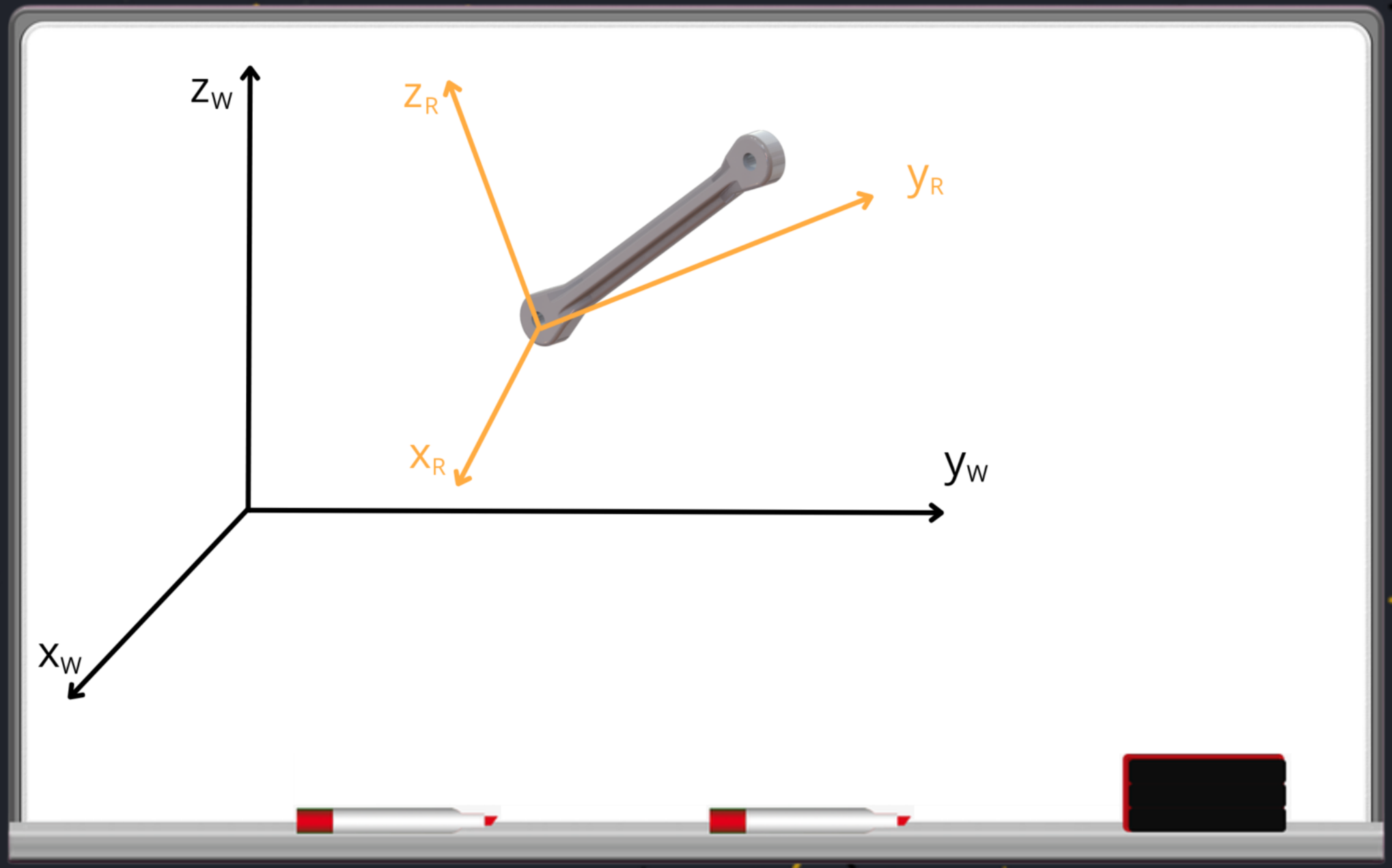
Translation

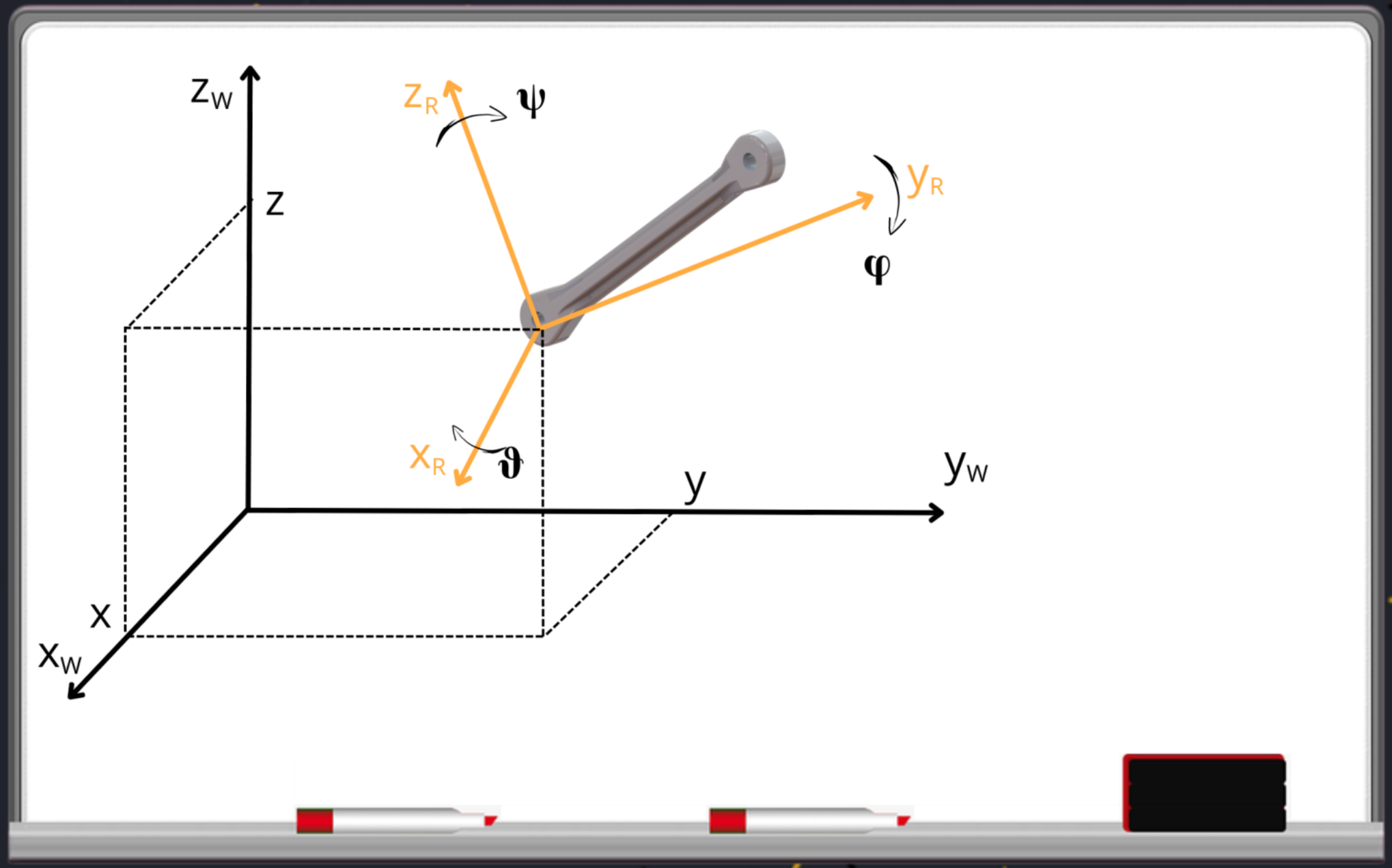


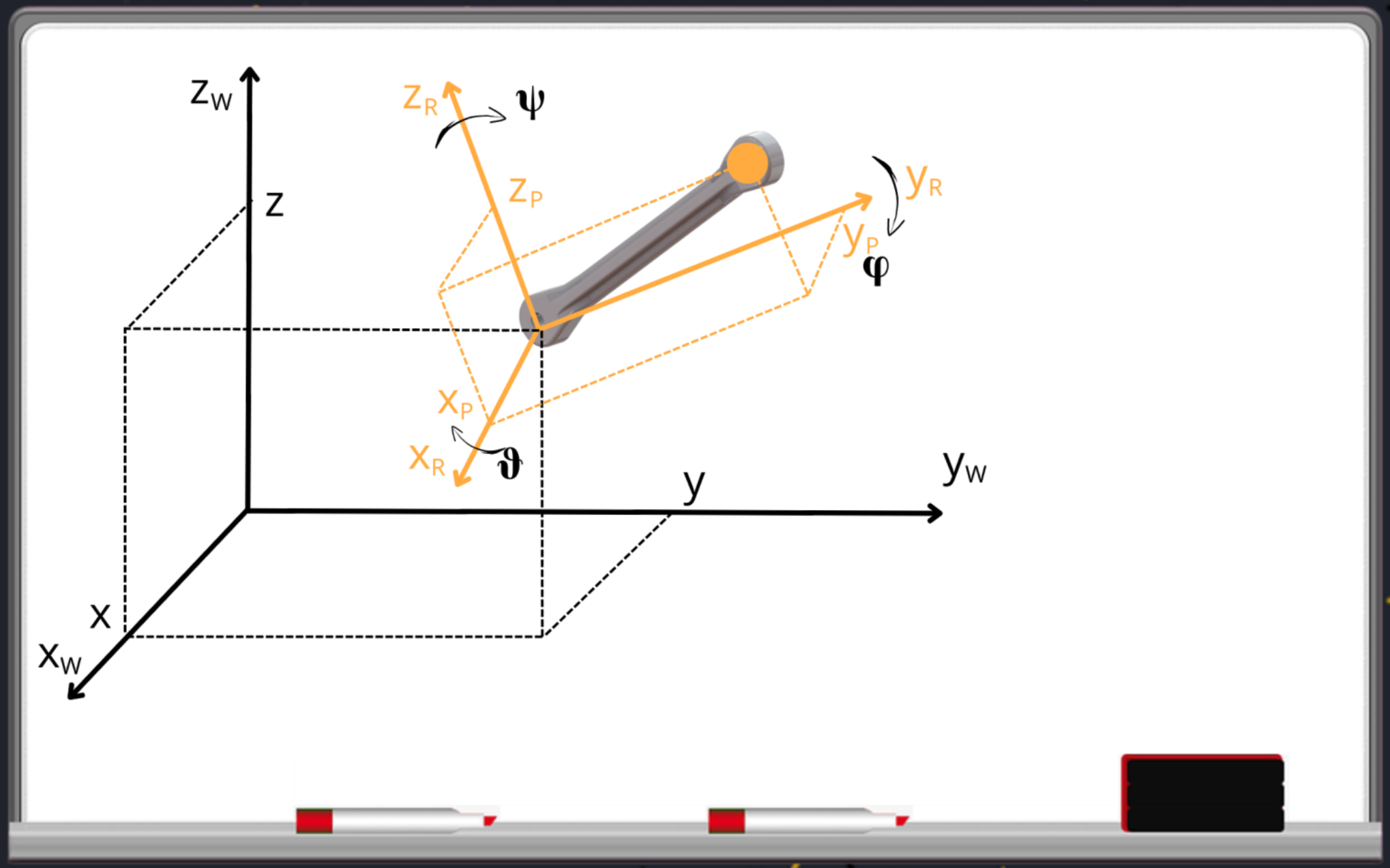
Rotation

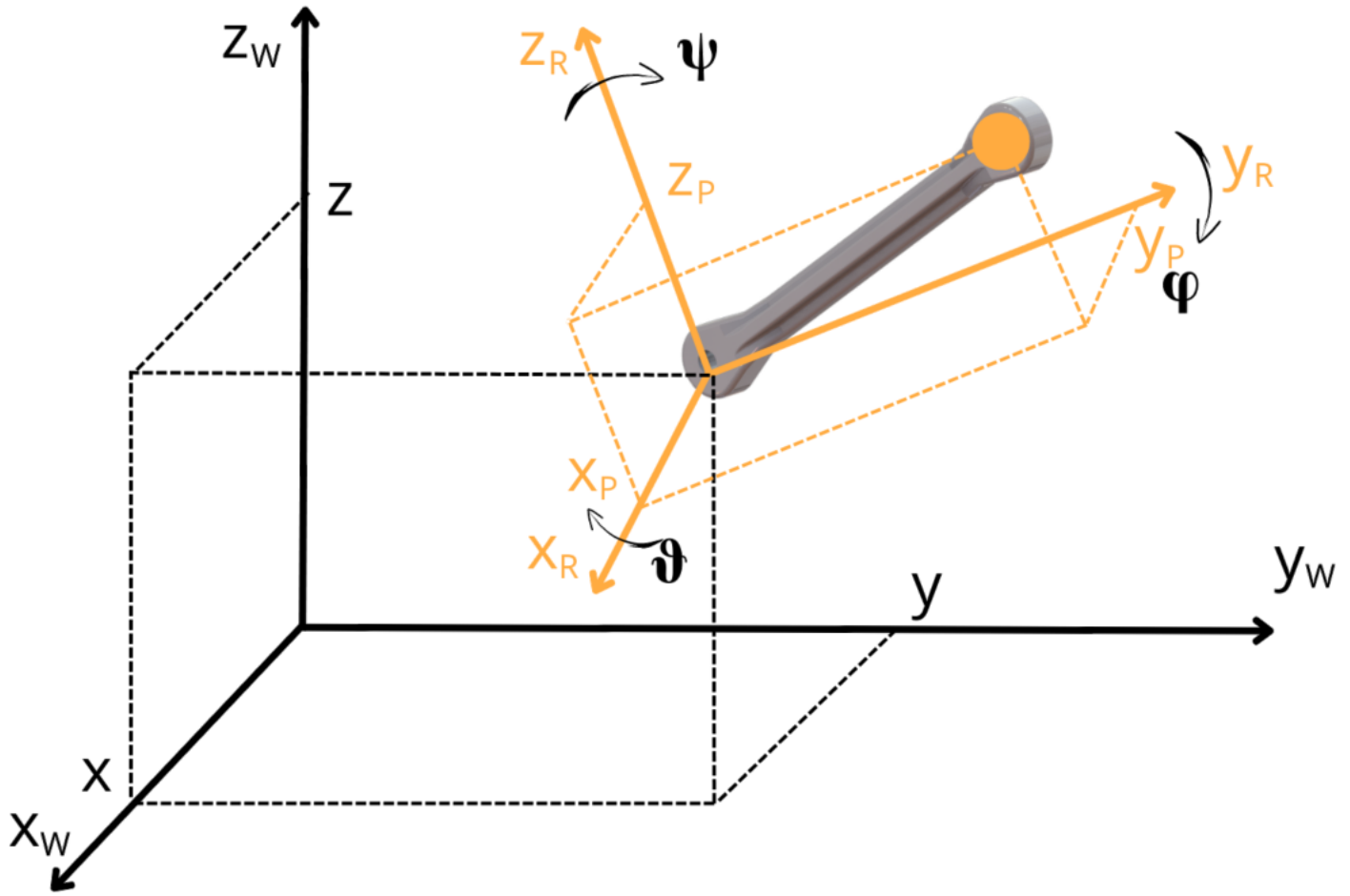


Roto-Translation


z_W
z_R
y_R
x_R
y_W
x_W

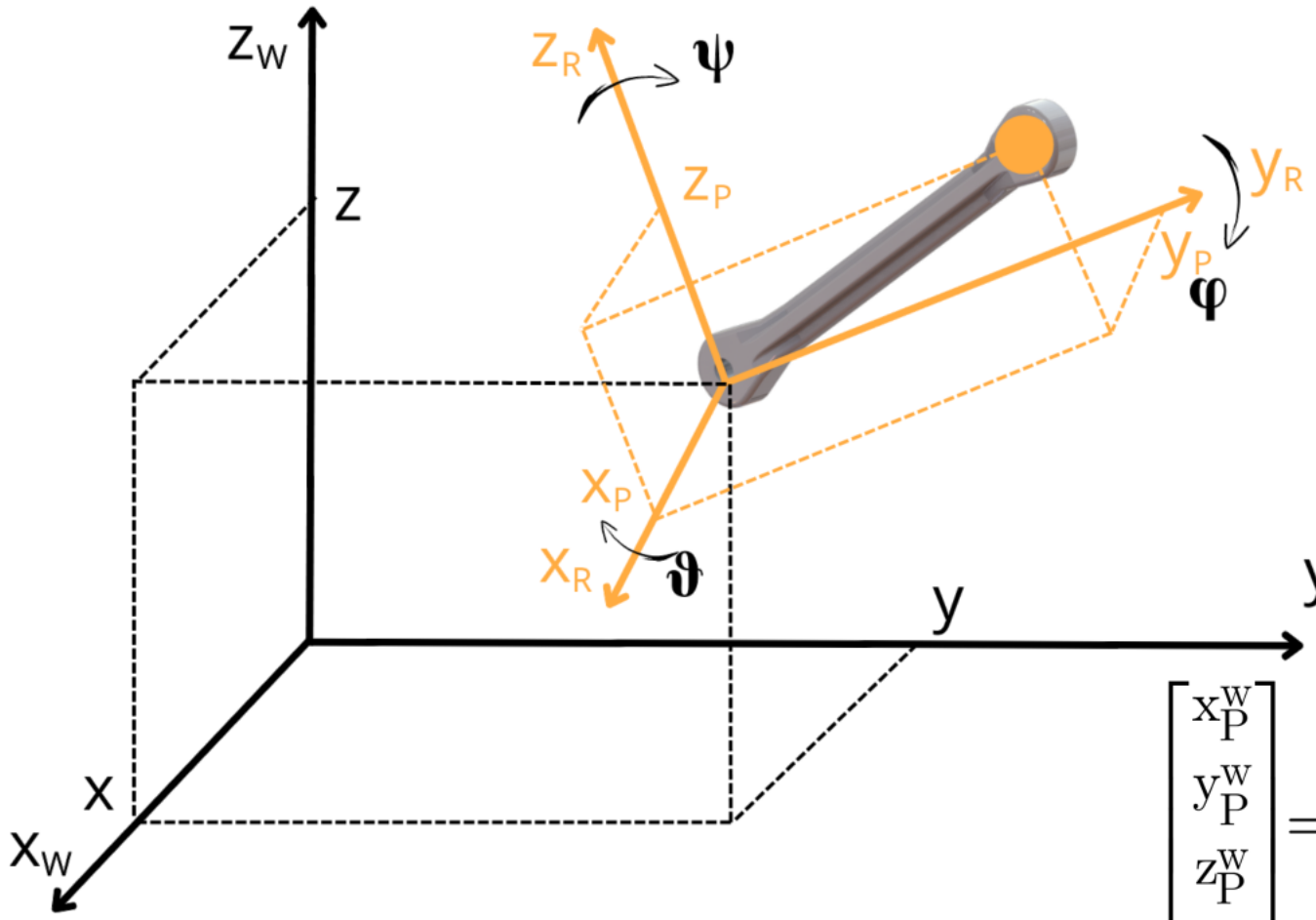

z_W
z
z_R
ψ
y_R
φ
x_R
ϑ
y
y_W
x
x_W


z_W
z
z_R
ψ
z_P
y_R
y_P
φ
x_P
x_R
ϑ
y
y_W
x
x_W



$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}$$

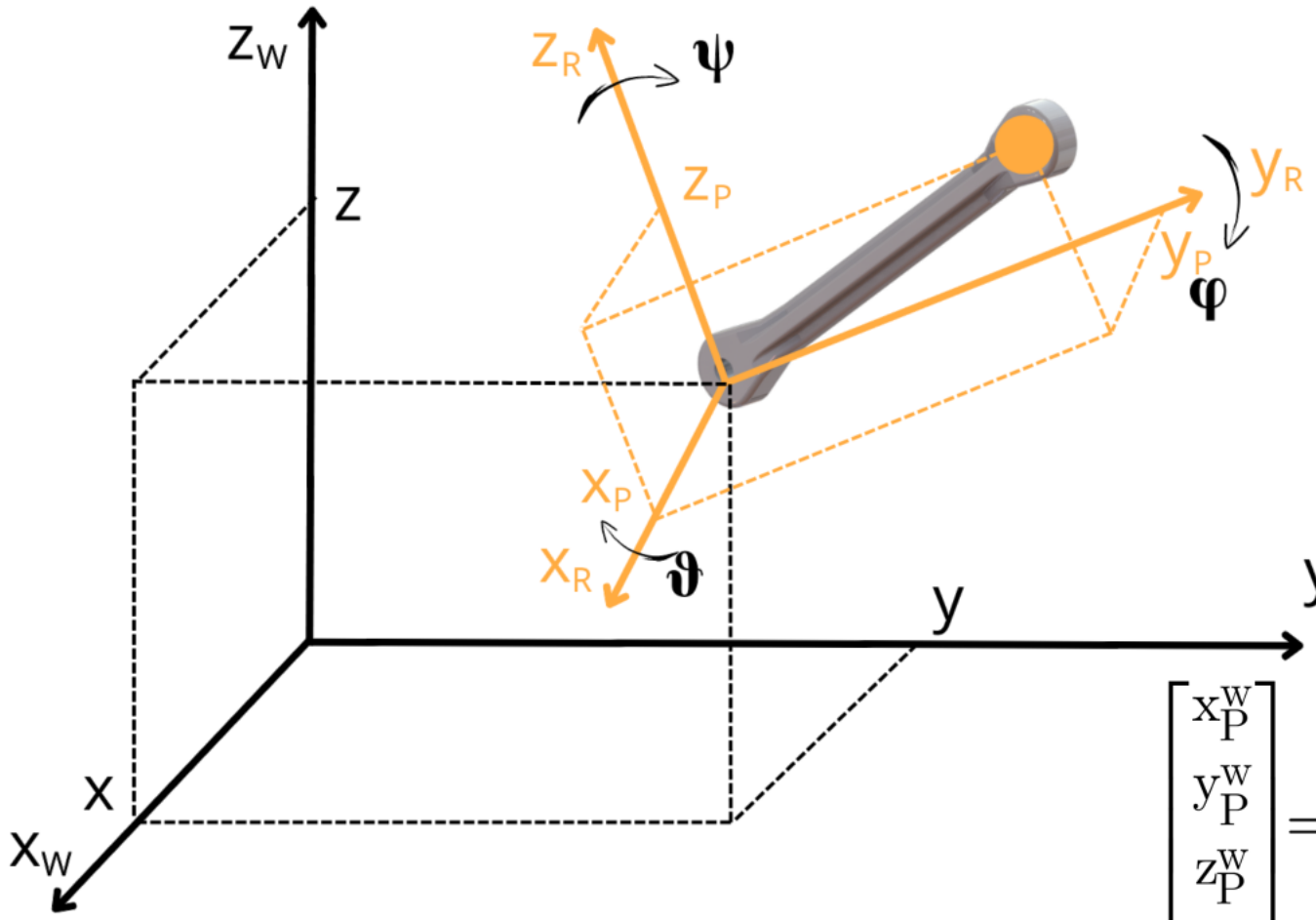
$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = R \cdot \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}$$



$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}$$

$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = R \cdot \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}$$

$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix}$$

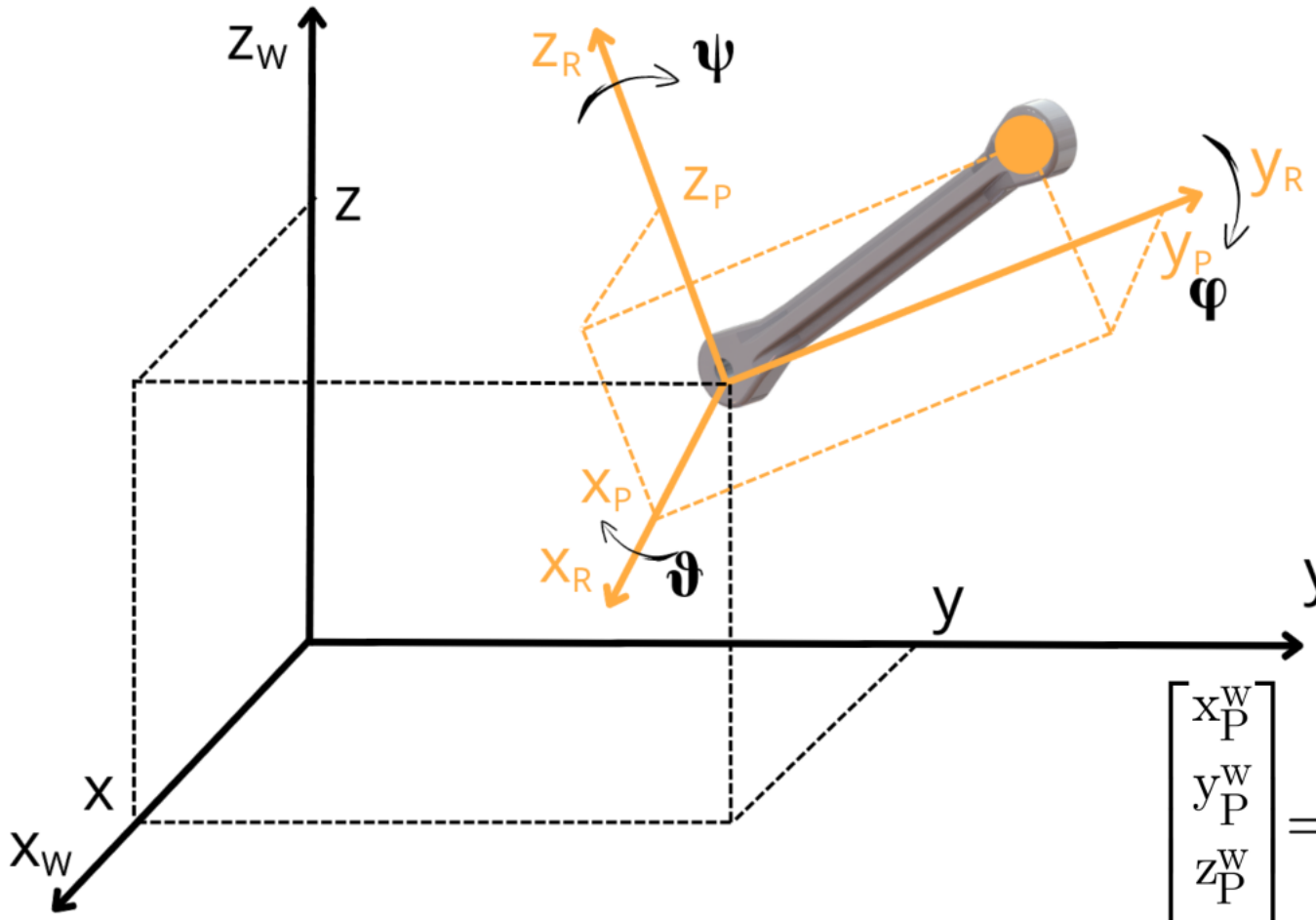


$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}$$

$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = R \cdot \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}$$

Rotation Matrix

$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix}$$



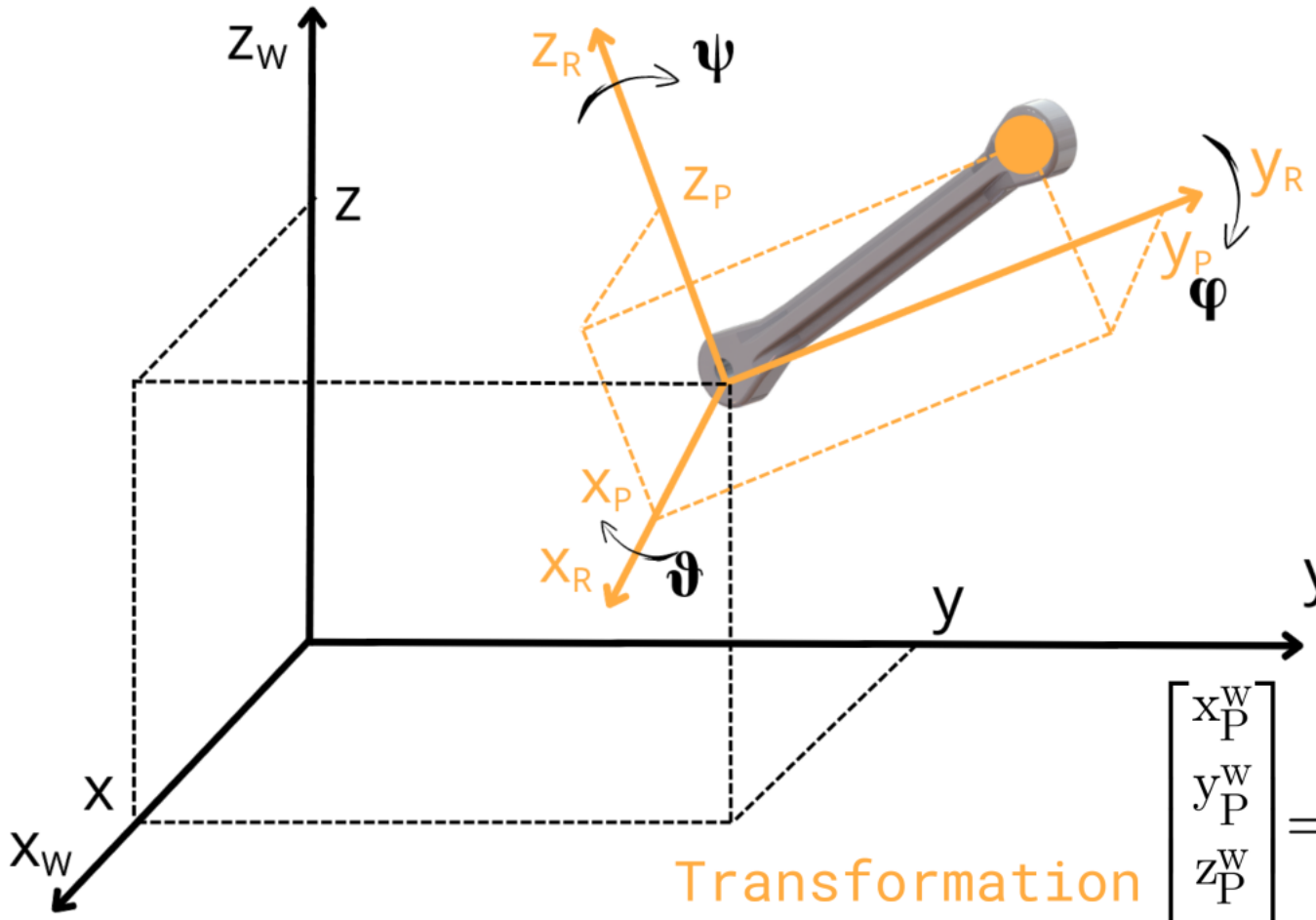
$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}$$

$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = R \cdot \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}$$

Rotation Matrix

$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix}$$

Translation Vector



$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}$$

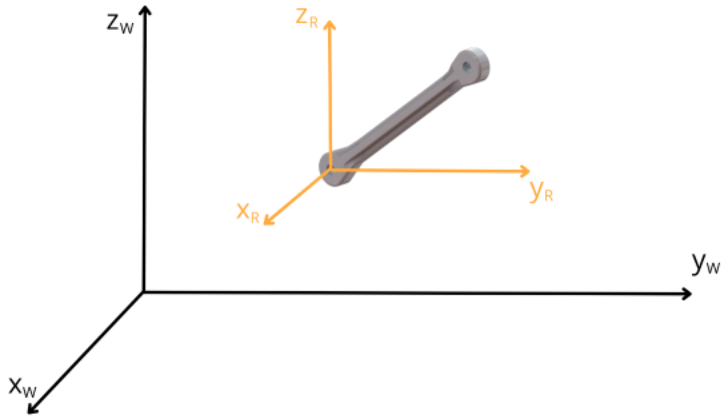
$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \end{bmatrix} = R \cdot \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}$$

Rotation Matrix

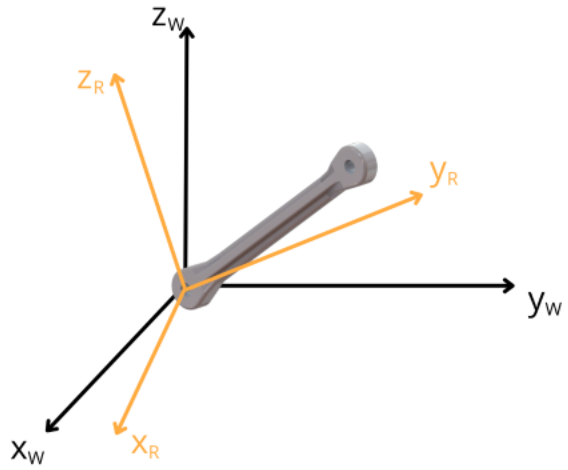
Transformation Matrix

$$\begin{bmatrix} x_P^W \\ y_P^W \\ z_P^W \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix}$$

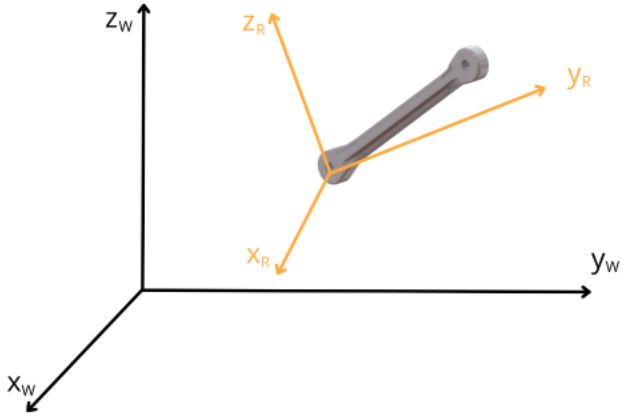
Translation Vector



Translation

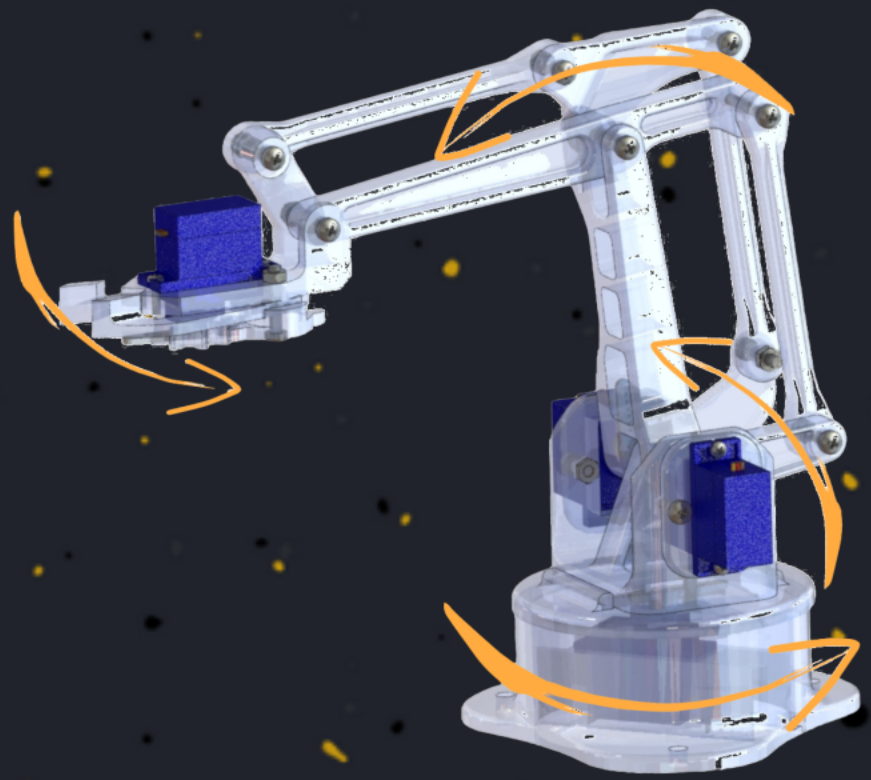


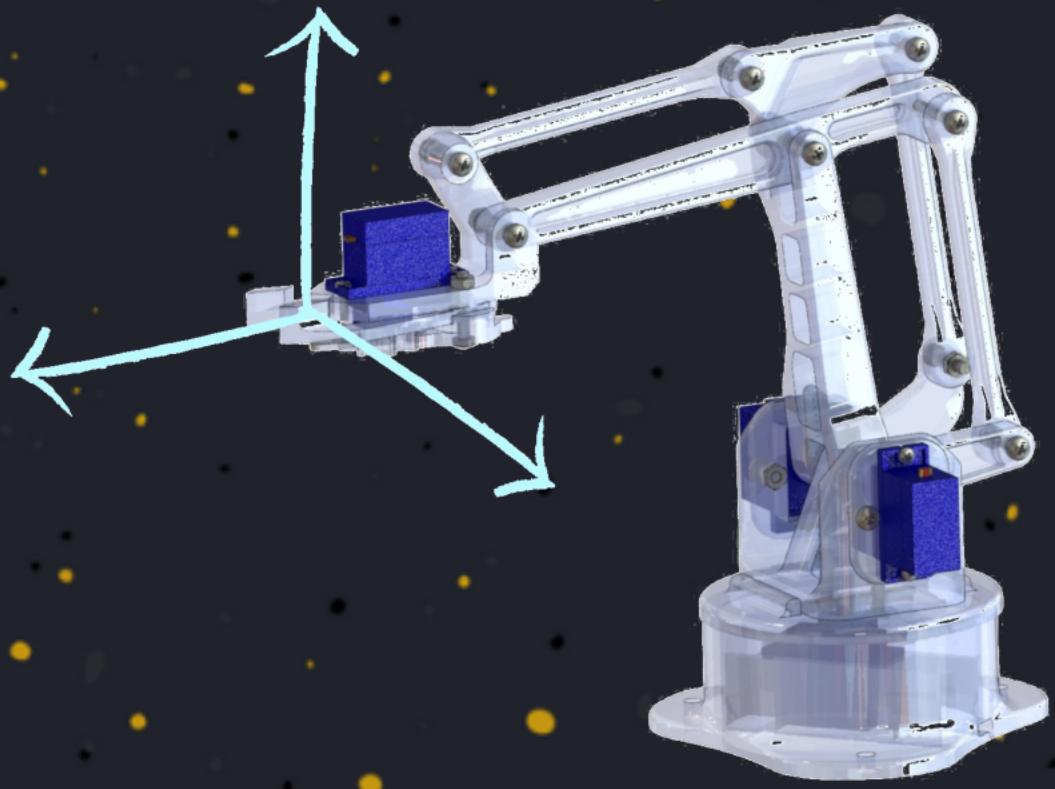
Rotation



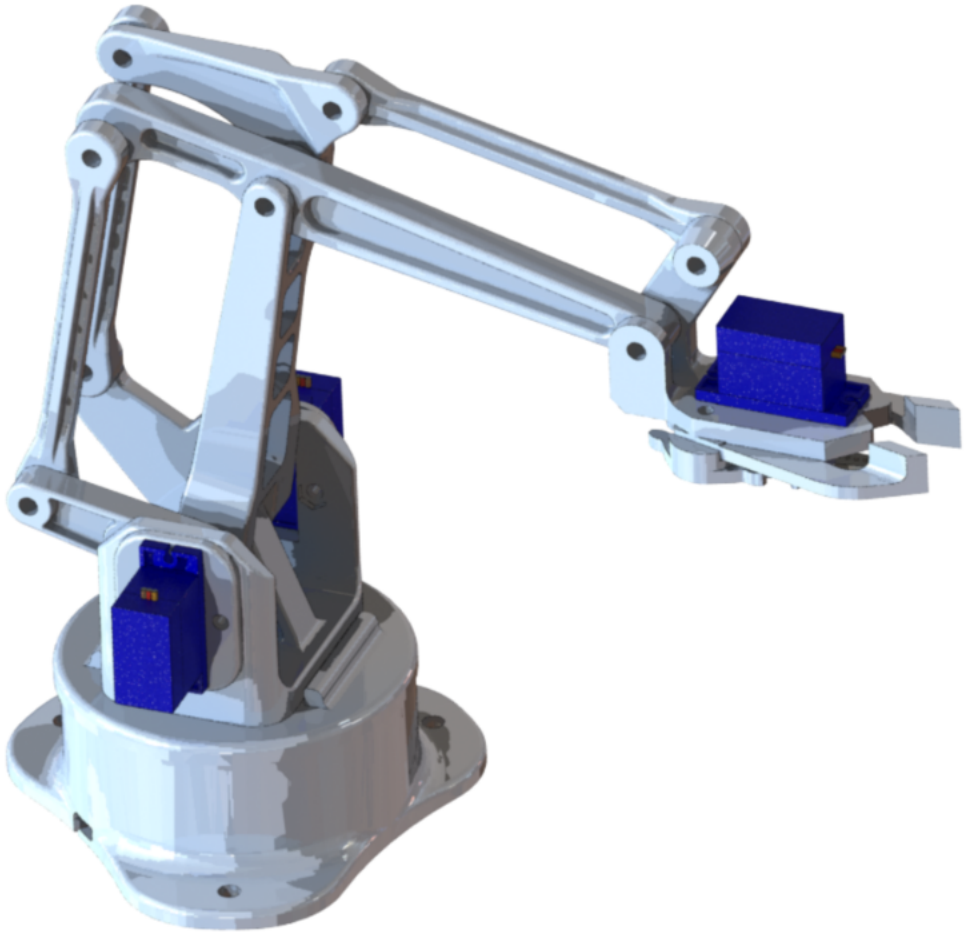
Roto-Translation

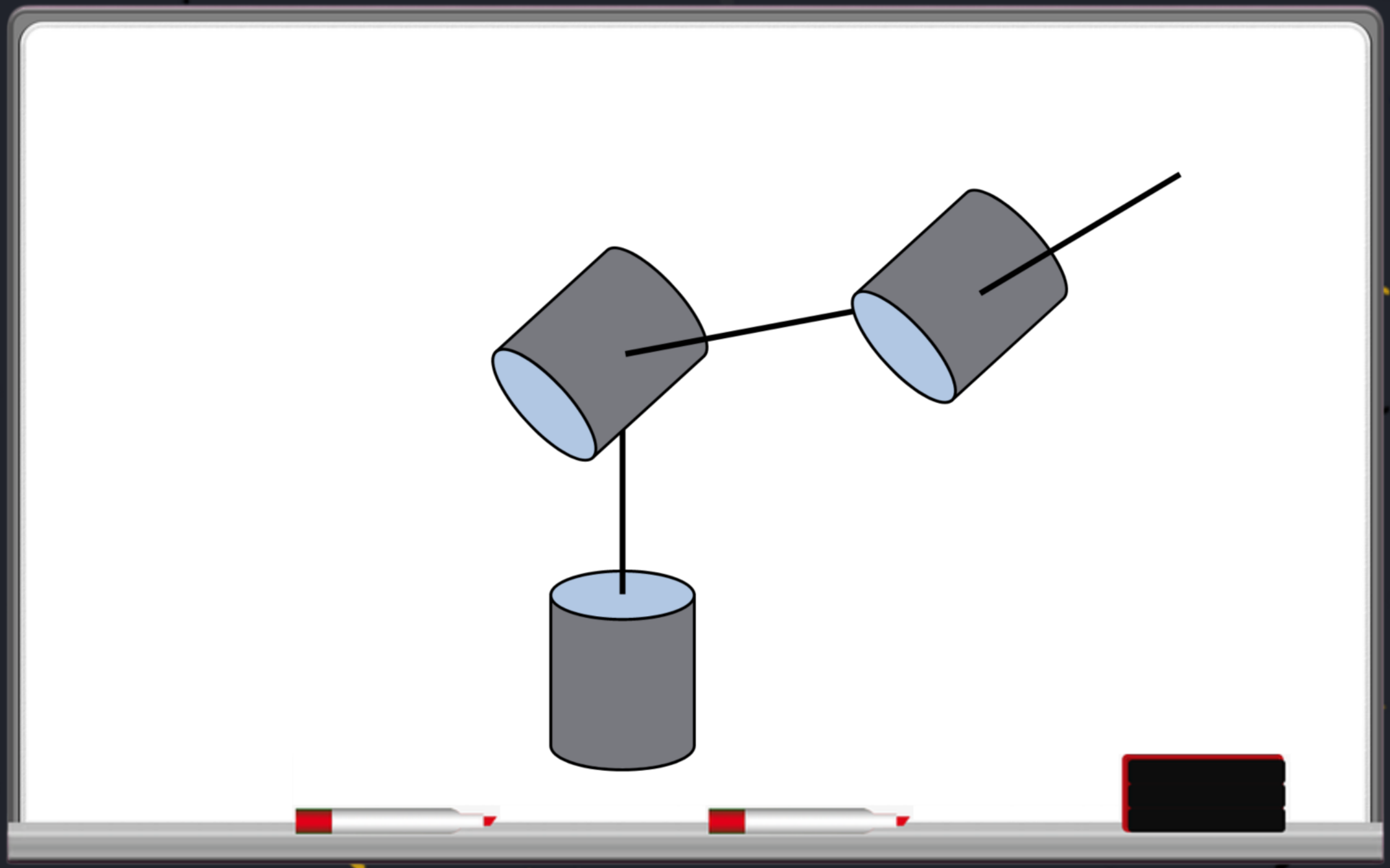
Forward

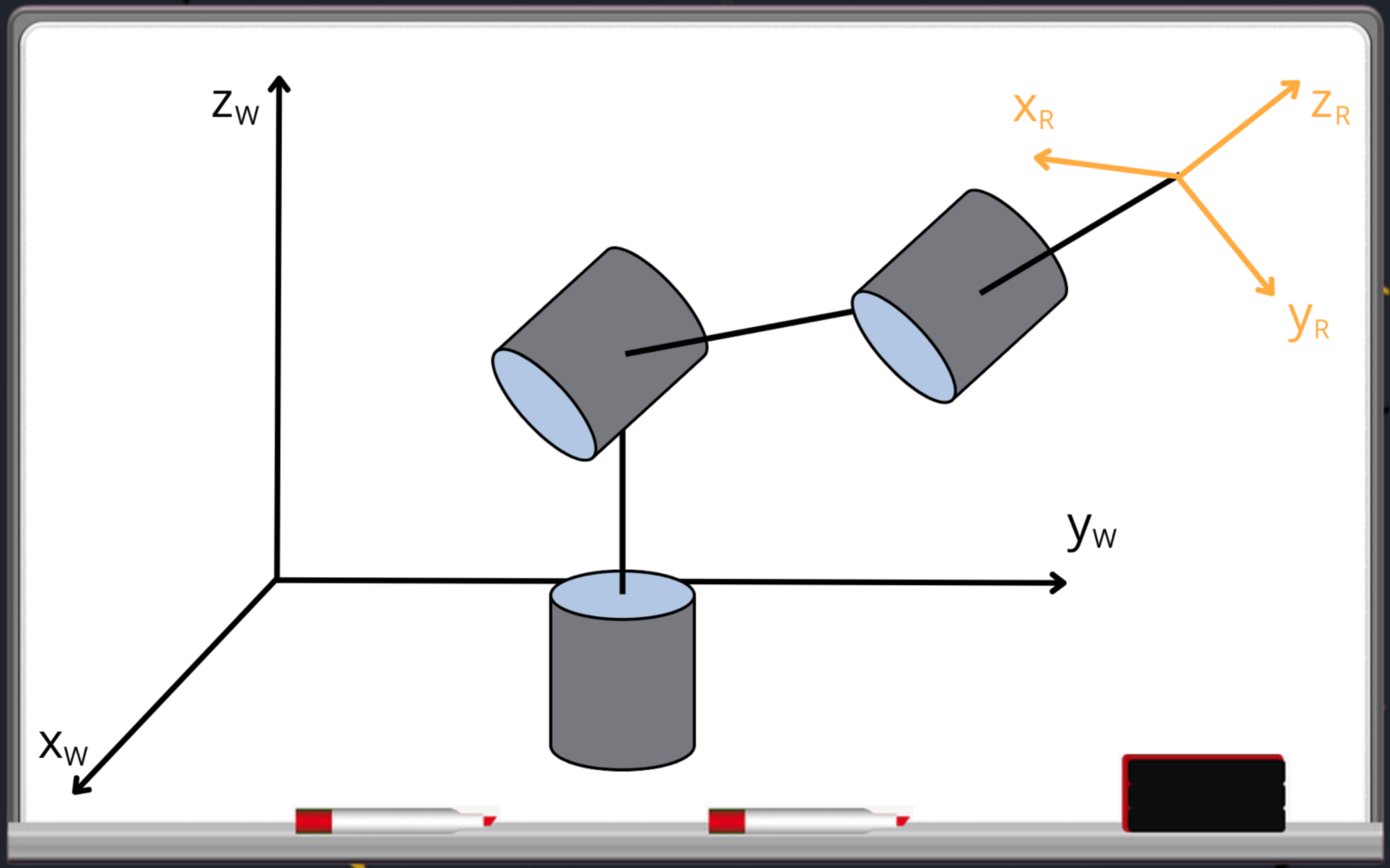


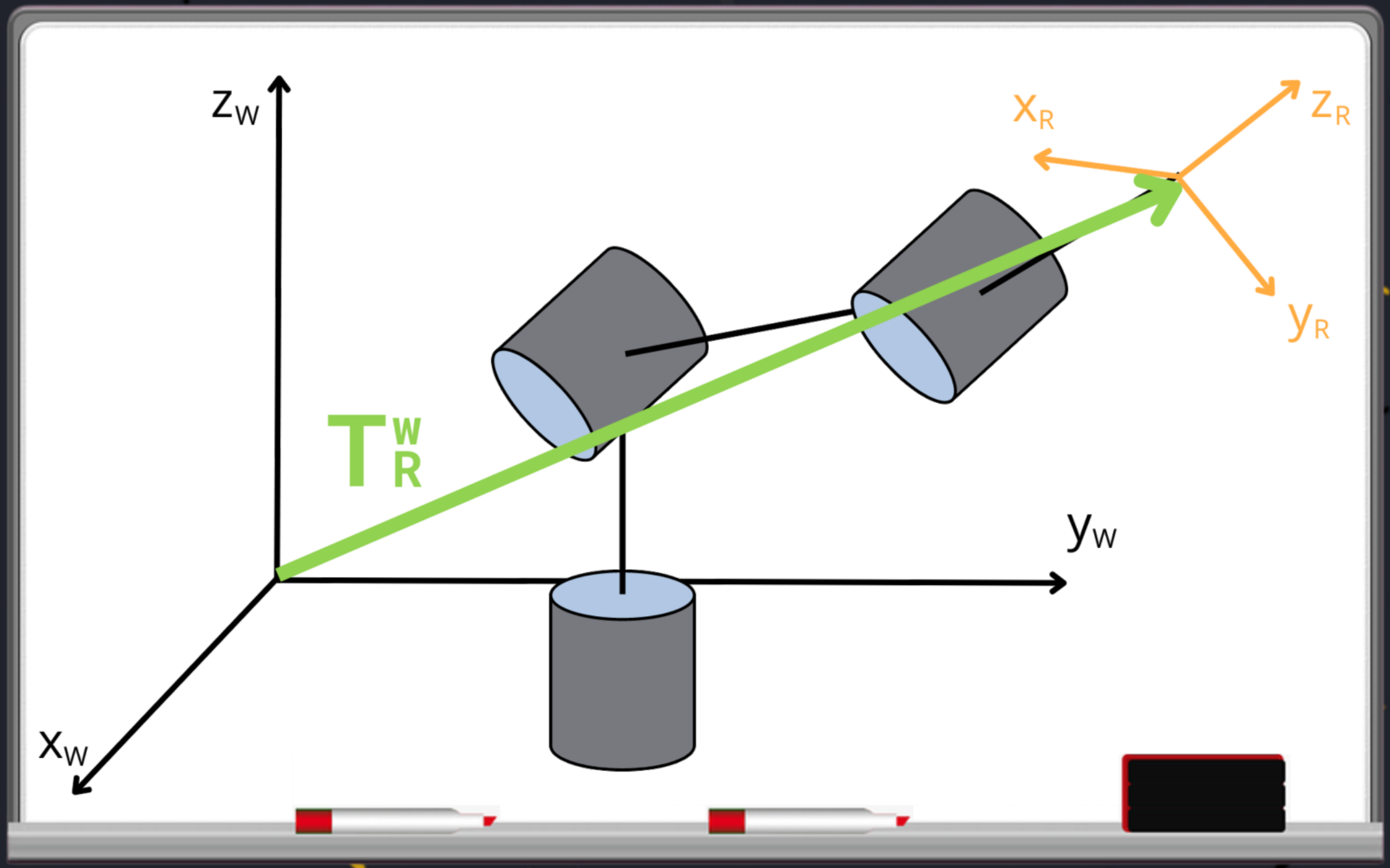


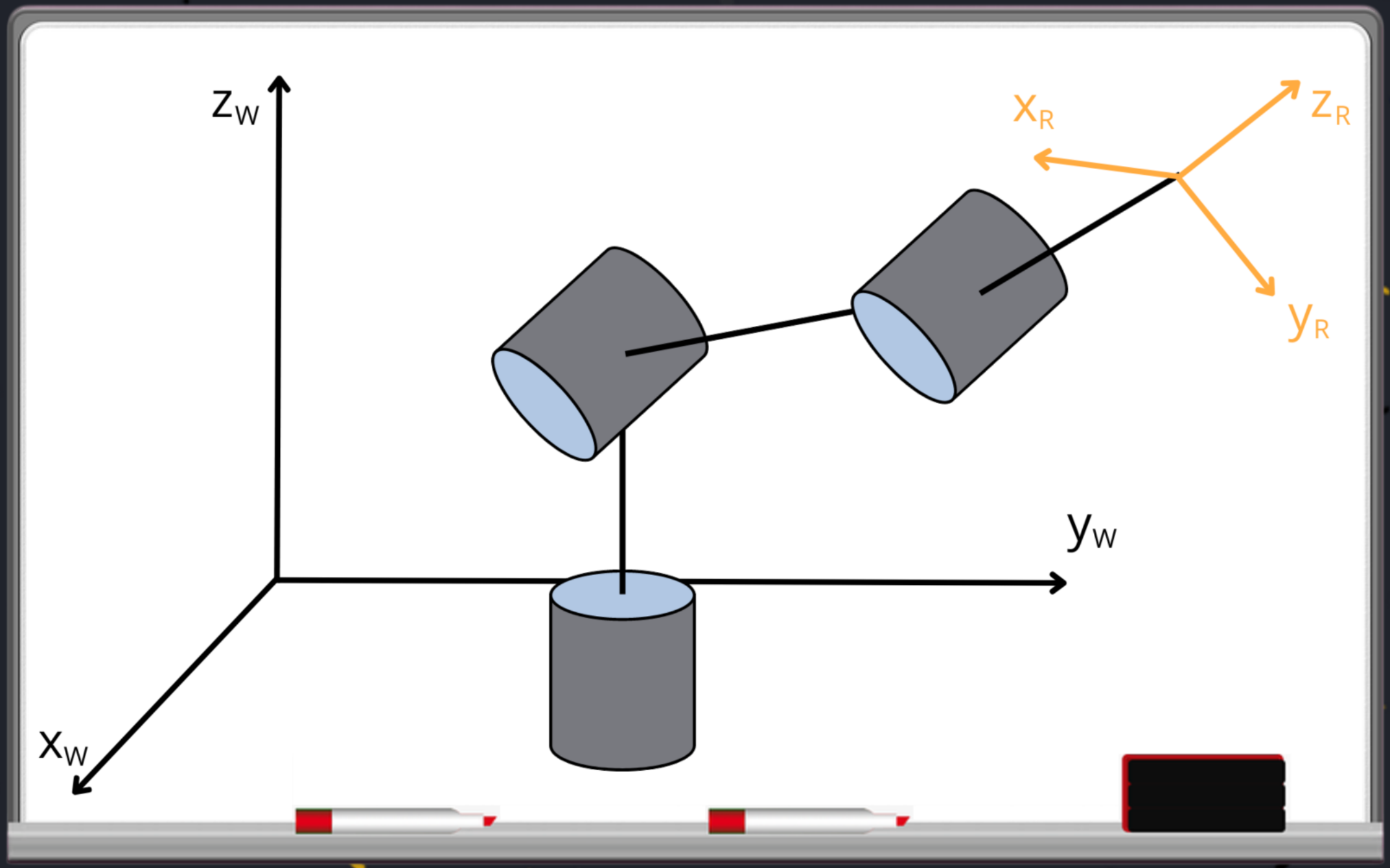
Inverse

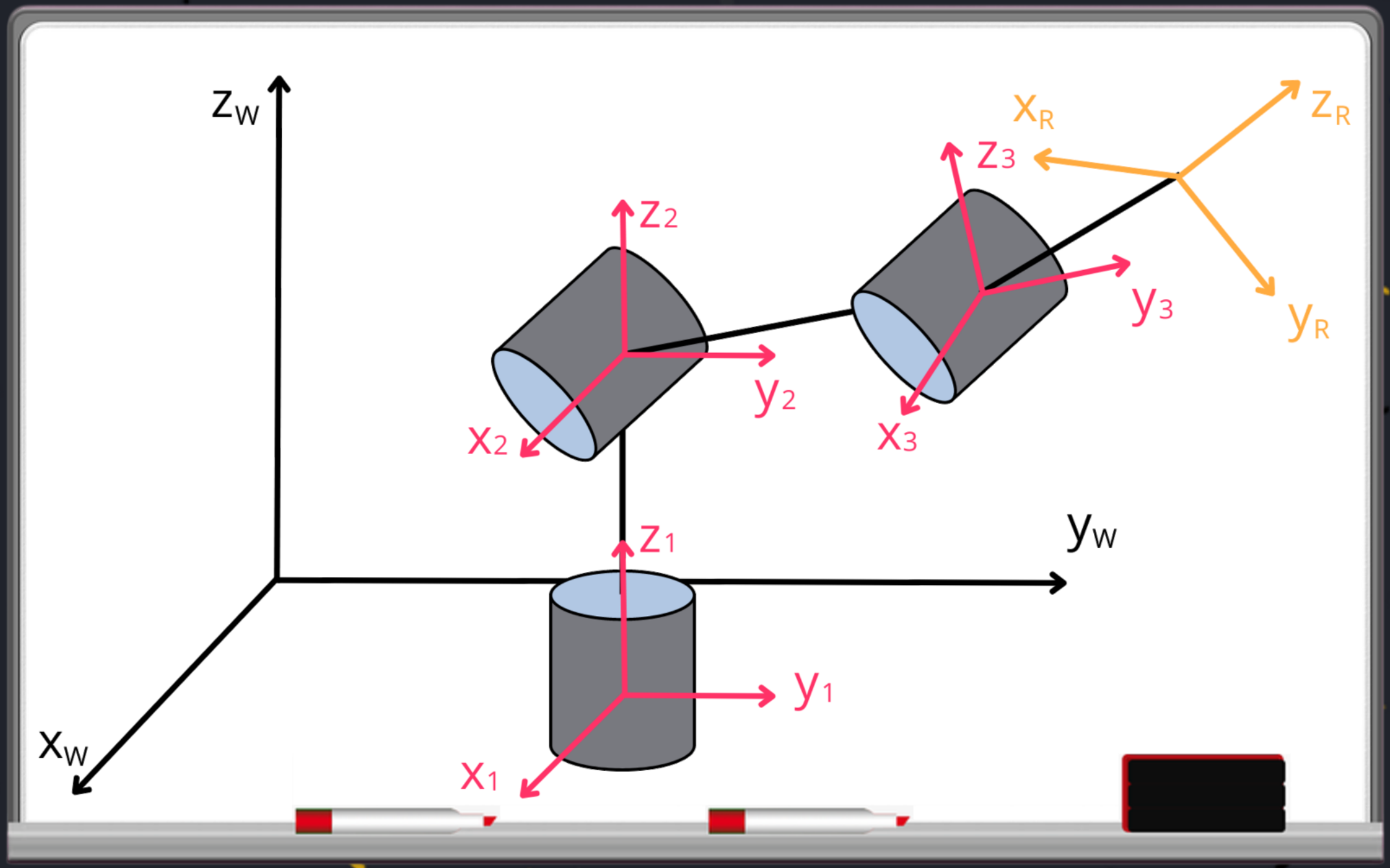


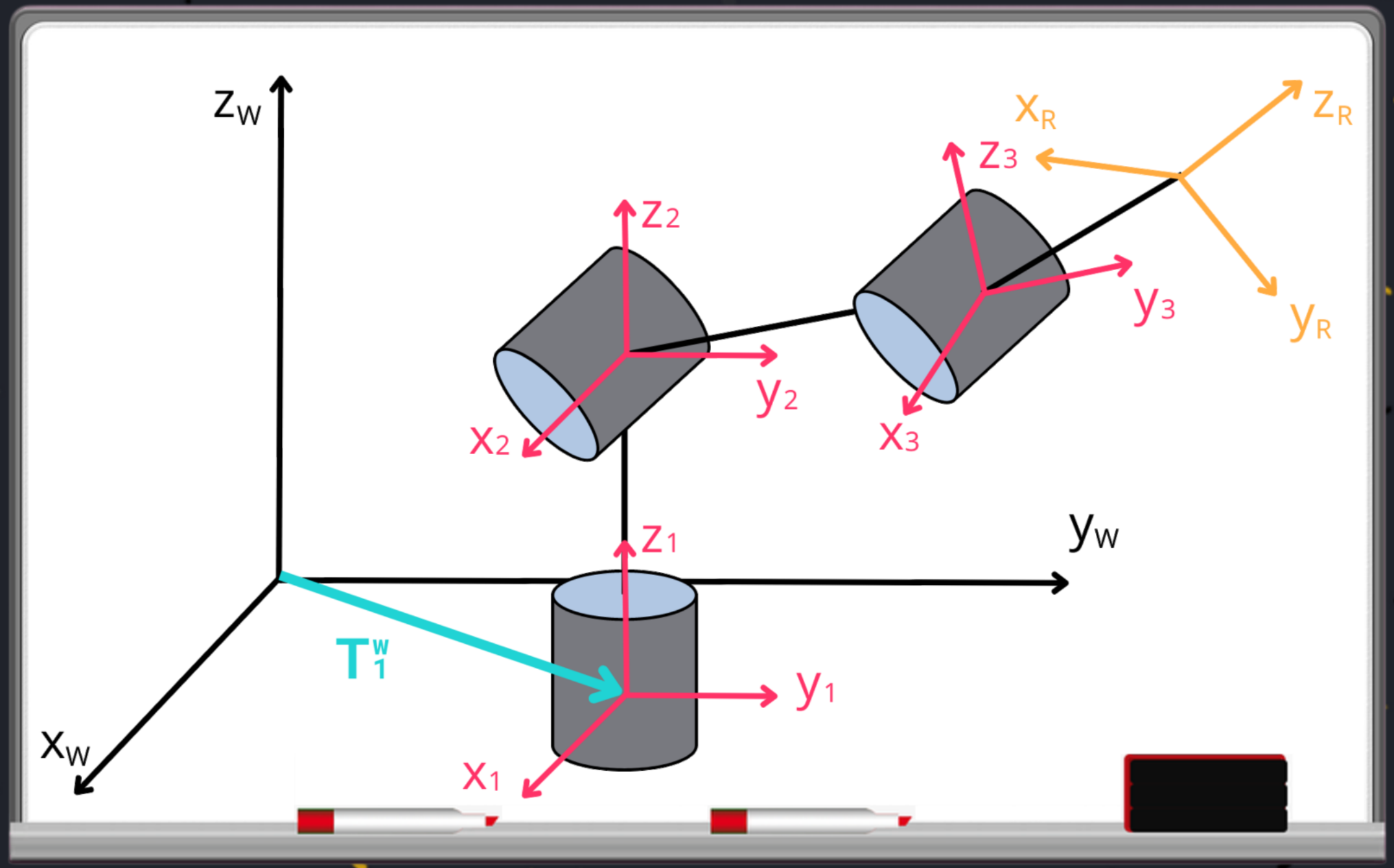


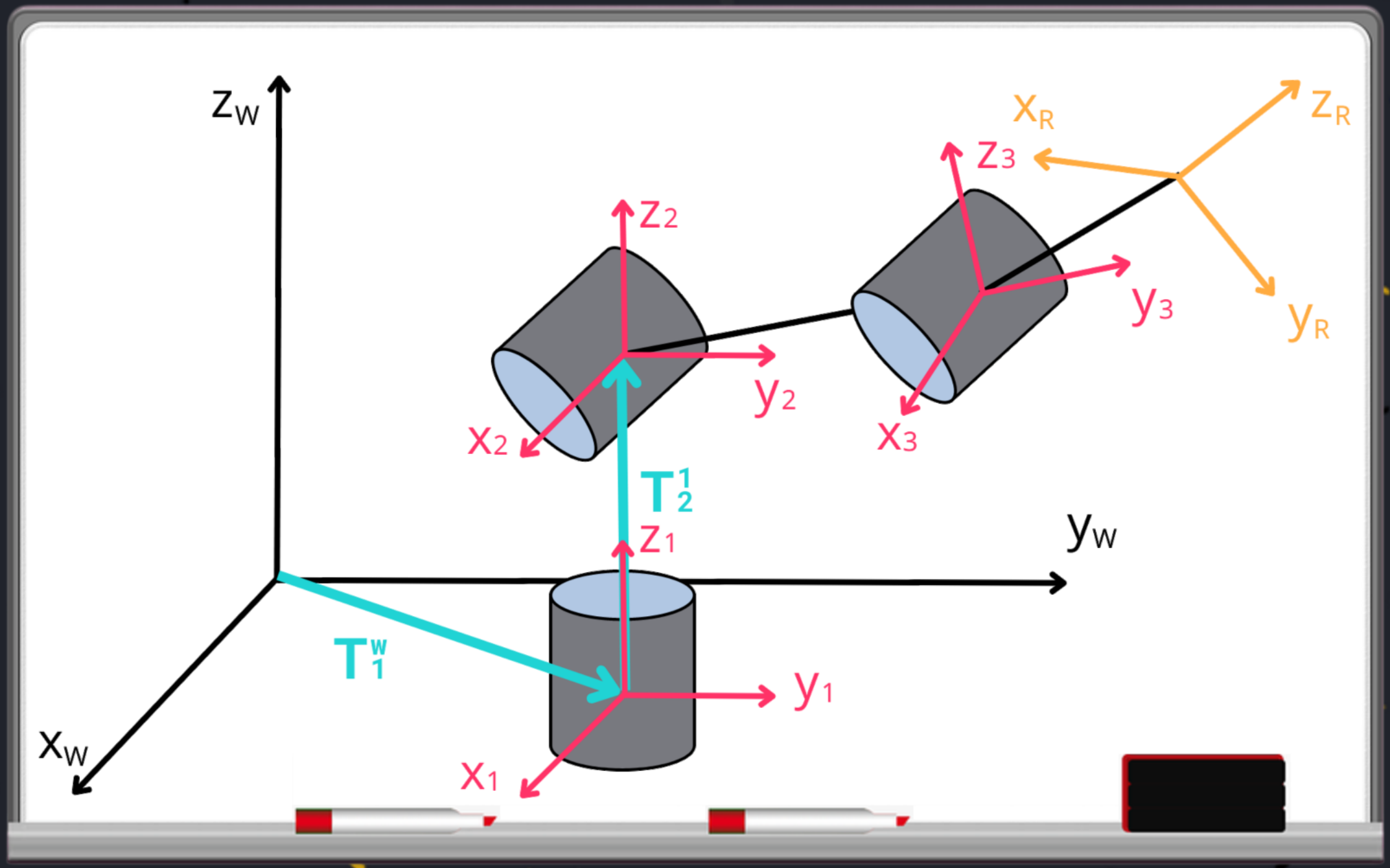

z_W
x_R
z_R
y_R
y_W
x_W

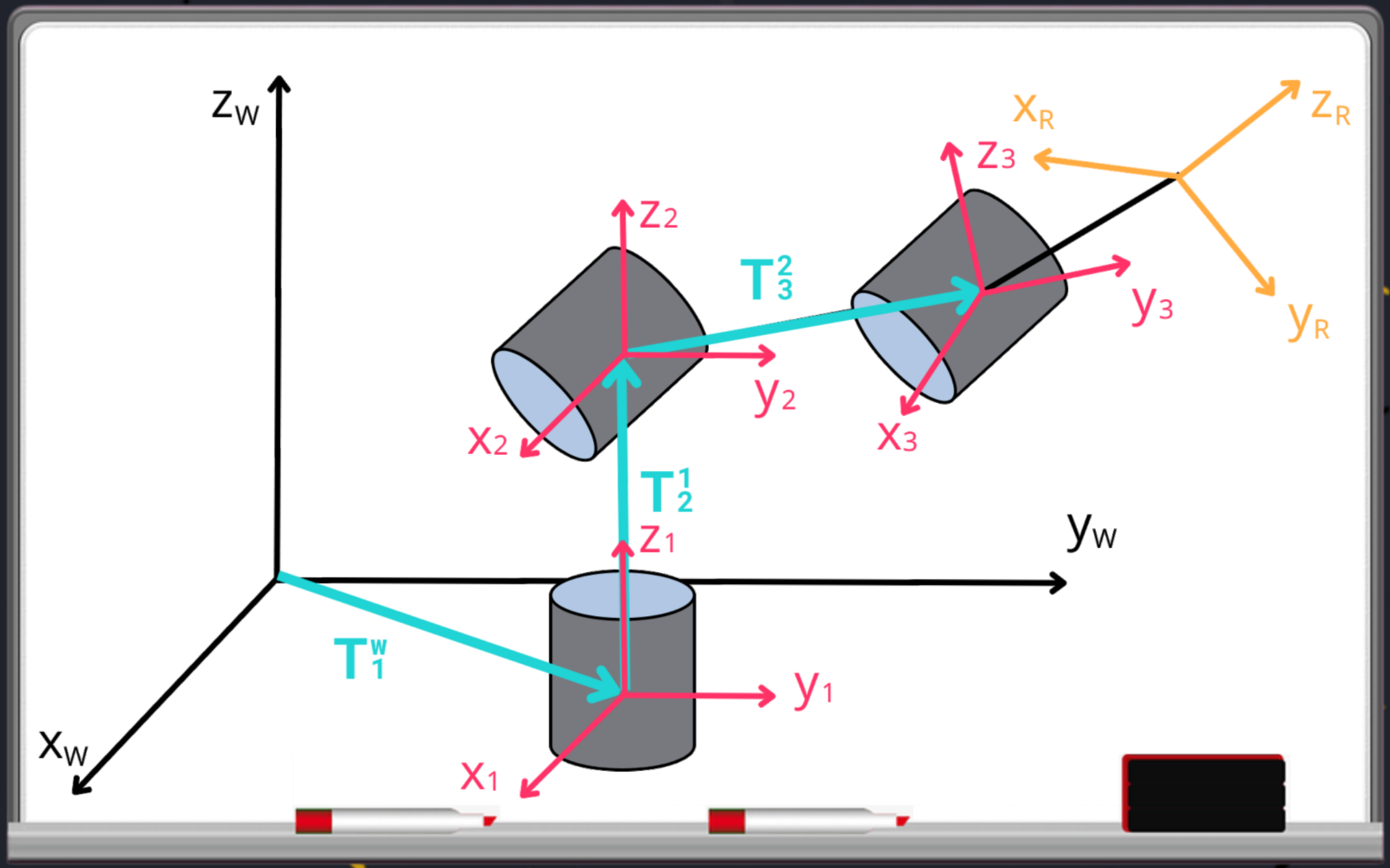

z_W
x_R
z_R
y_R
T_R^W
y_W
x_W

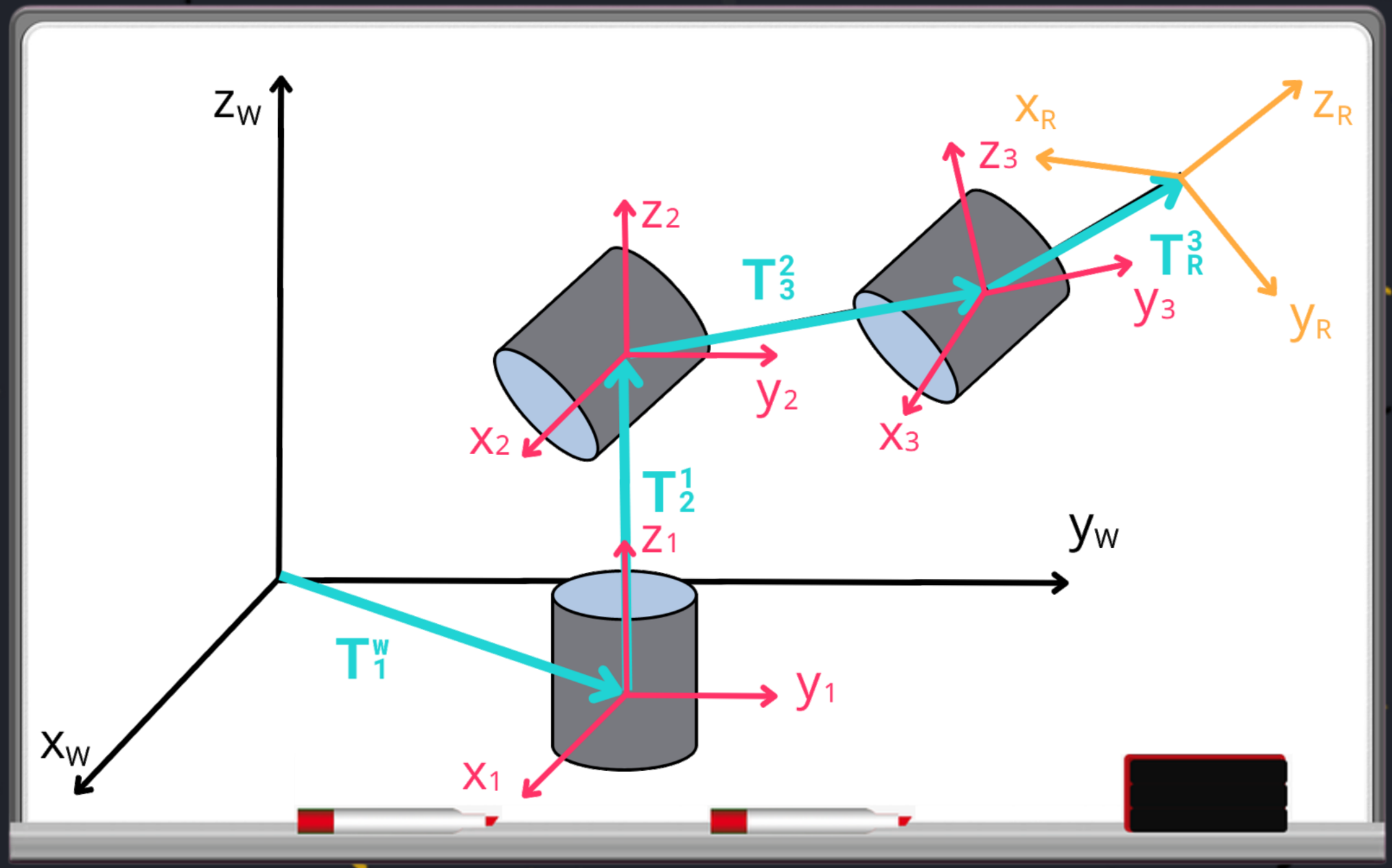

z_W
x_R
z_R
y_R
y_W
x_W

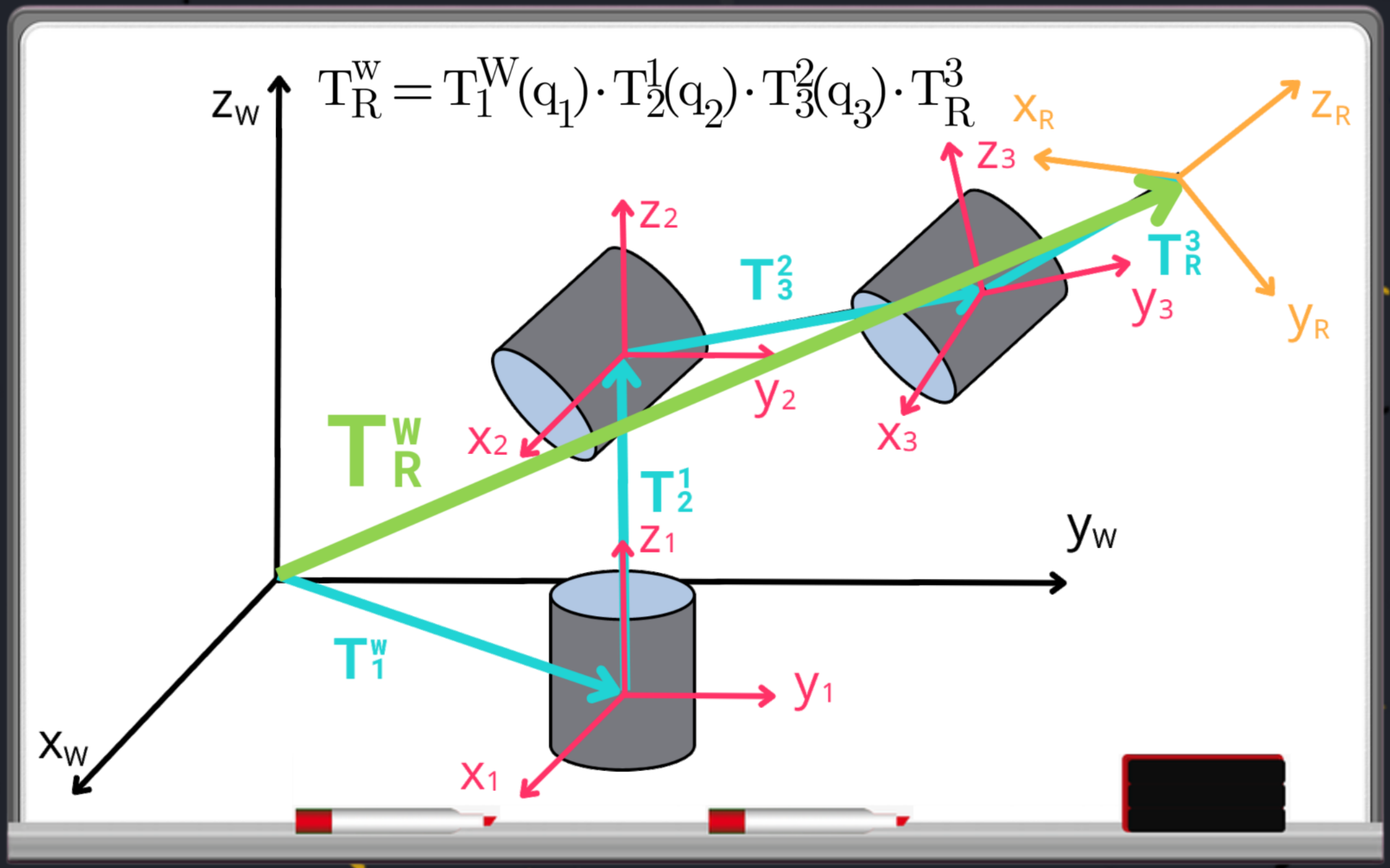

z_W
x_W
y_W
z_1
x_1
y_1
z_2
x_2
y_2
z_3
x_3
y_3
x_R
z_R
y_R

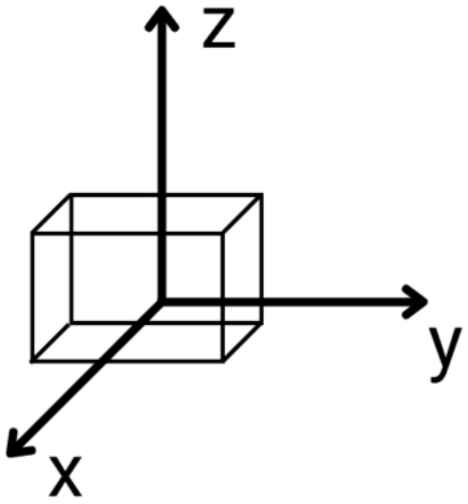

z_W
x_W
y_W
z_1
x_1
y_1
z_2
x_2
y_2
z_3
x_3
y_3
x_R
z_R
y_R
T_1^W

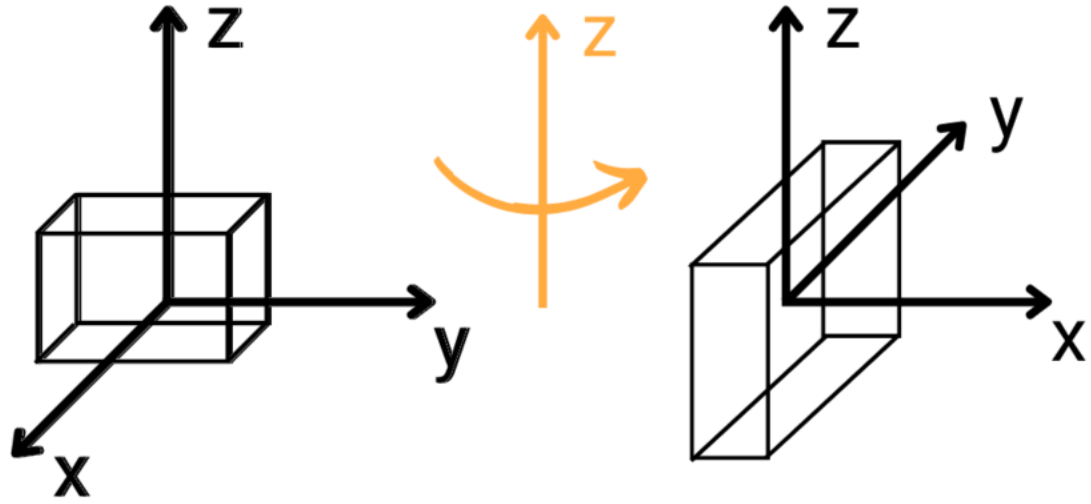

z_W
x_W
y_W
z_1
x_1
y_1
z_2
x_2
y_2
z_3
x_3
y_3
x_R
z_R
y_R
T_1^W
T_2^1

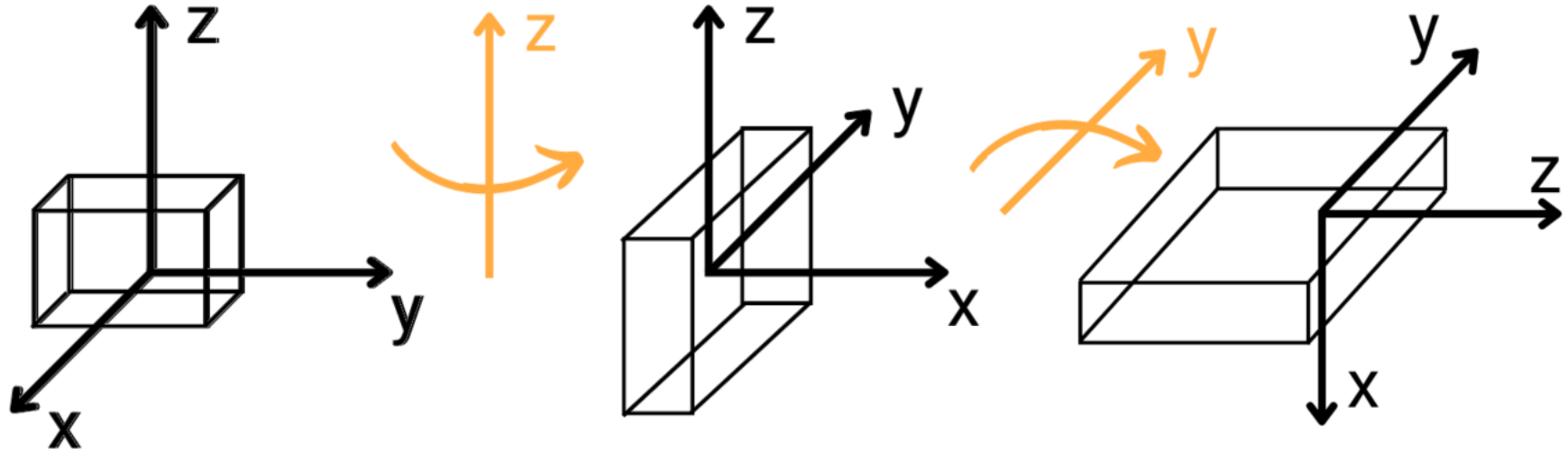

z_W
x_W
y_W
z_2
y_2
x_2
T_3^2
z_3
x_R
z_R
y_3
y_R
x_3
T_2^1
z_1
T_1^w
y_1
x_1

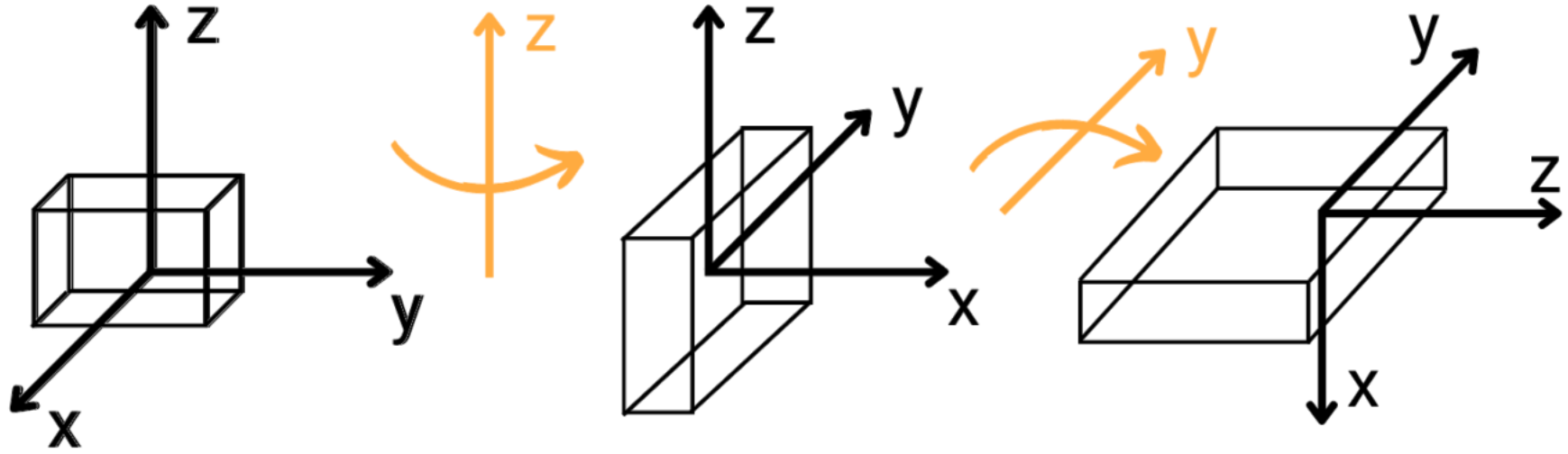

z_W
x_W
y_W
z_1
x_1
y_1
z_2
x_2
y_2
z_3
x_3
y_3
x_R
z_R
y_R
T_1^W
T_2^1
T_3^2
T_R^3

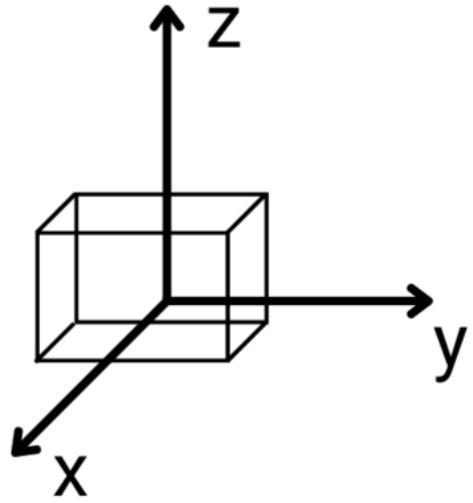

$T_R^w = T_1^W(q_1) \cdot T_2^1(q_2) \cdot T_3^2(q_3) \cdot T_R^3$
z_W
y_W
x_W
z_1
y_1
x_1
z_2
y_2
x_2
z_3
y_3
x_3
z_R
y_R
x_R
T_1^w
T_2^1
T_3^2
T_R^3
T_R^w

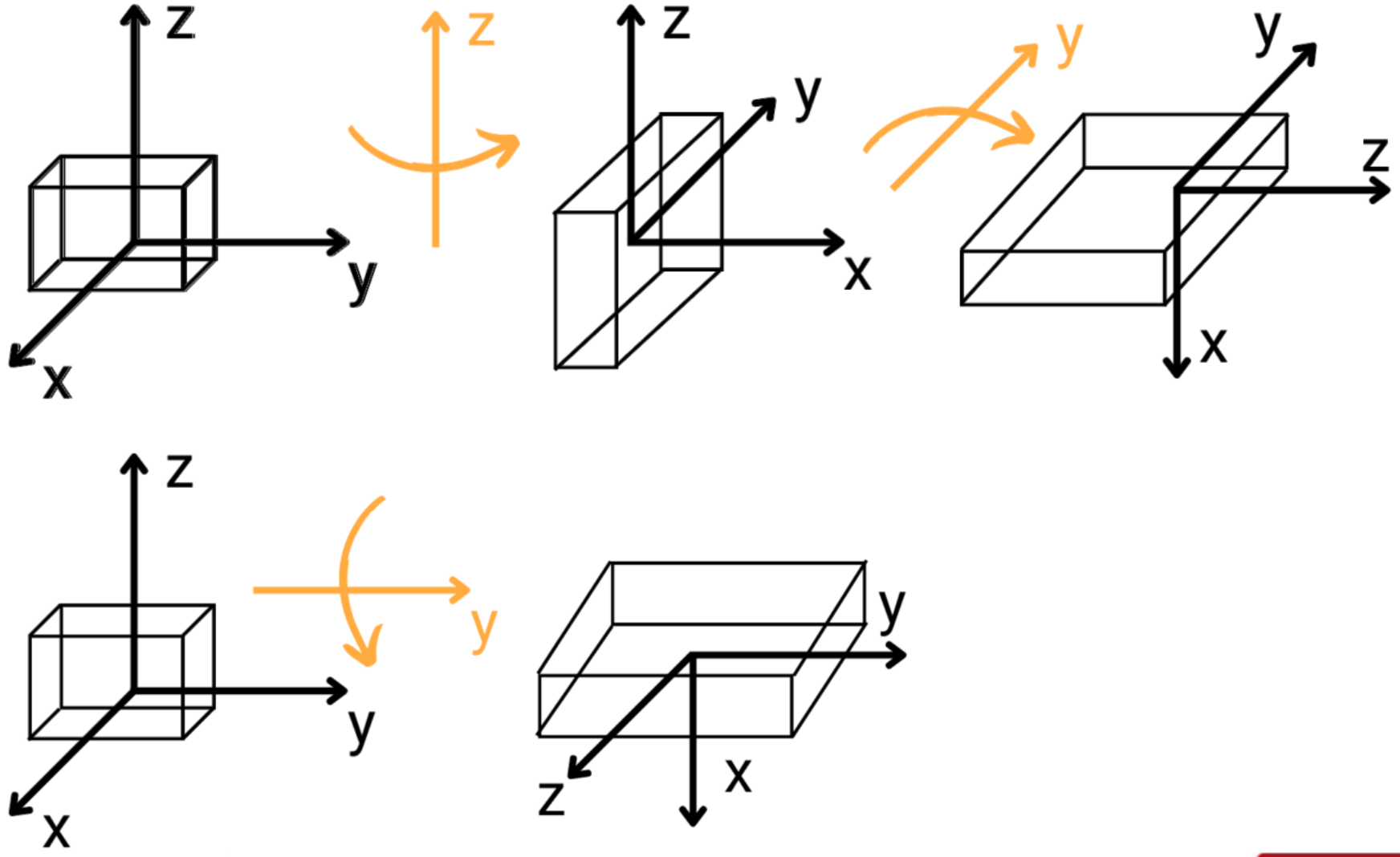

z
y
x

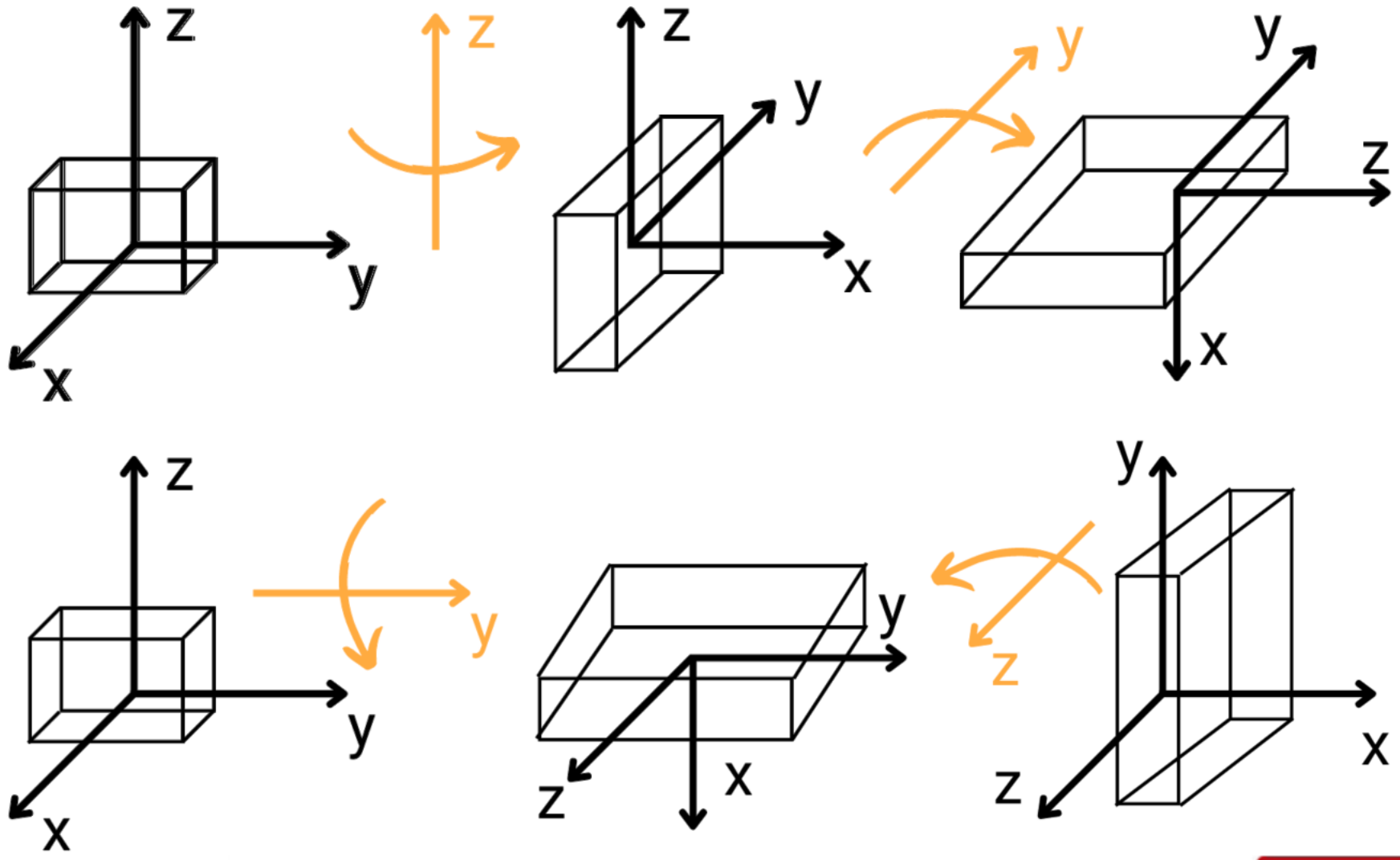

z
y
x
z
z
y
x

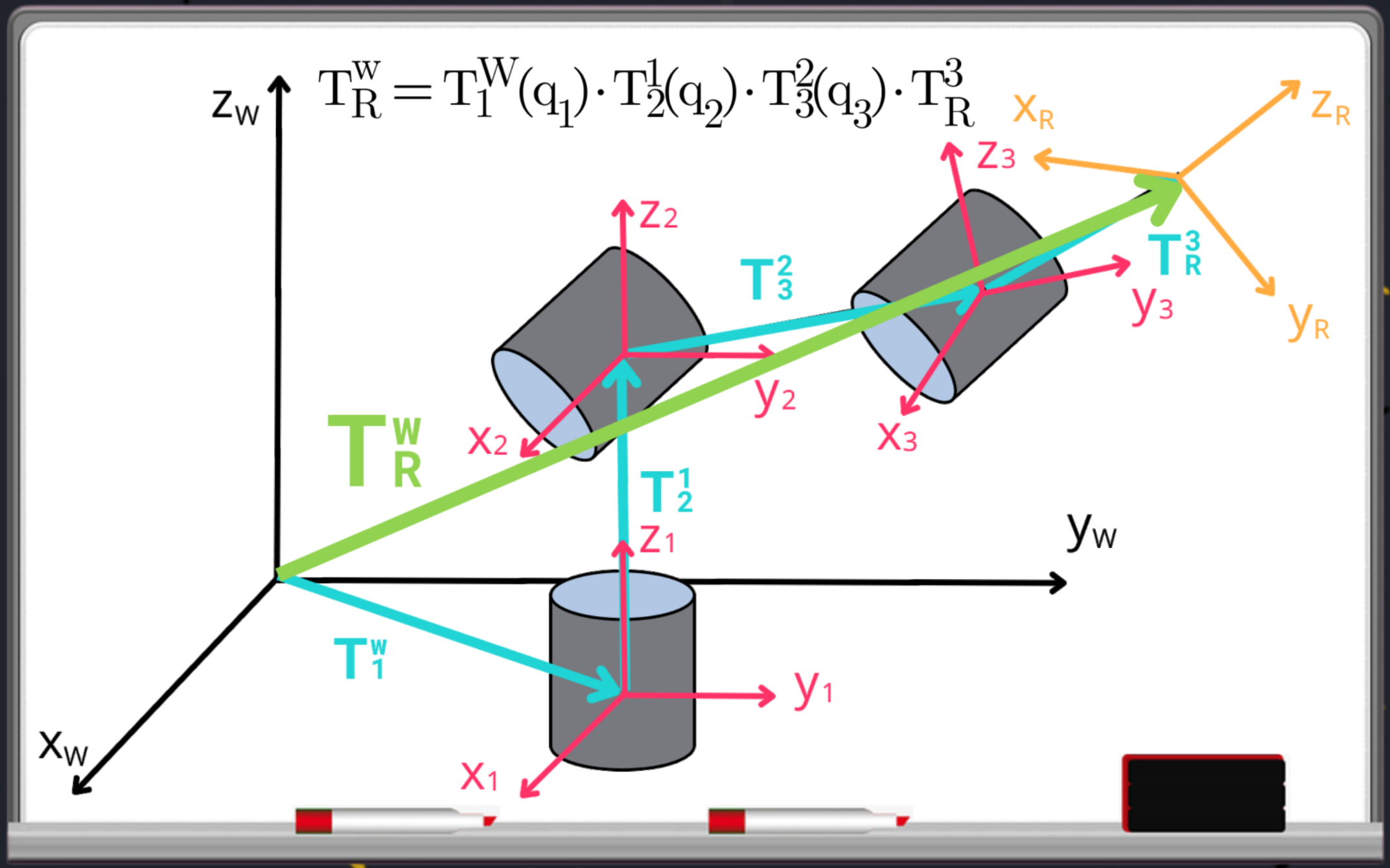

z
y
x
z
z
y
x
y
y
z
x


z
y
x
z
z
y
x
y
y
z
x

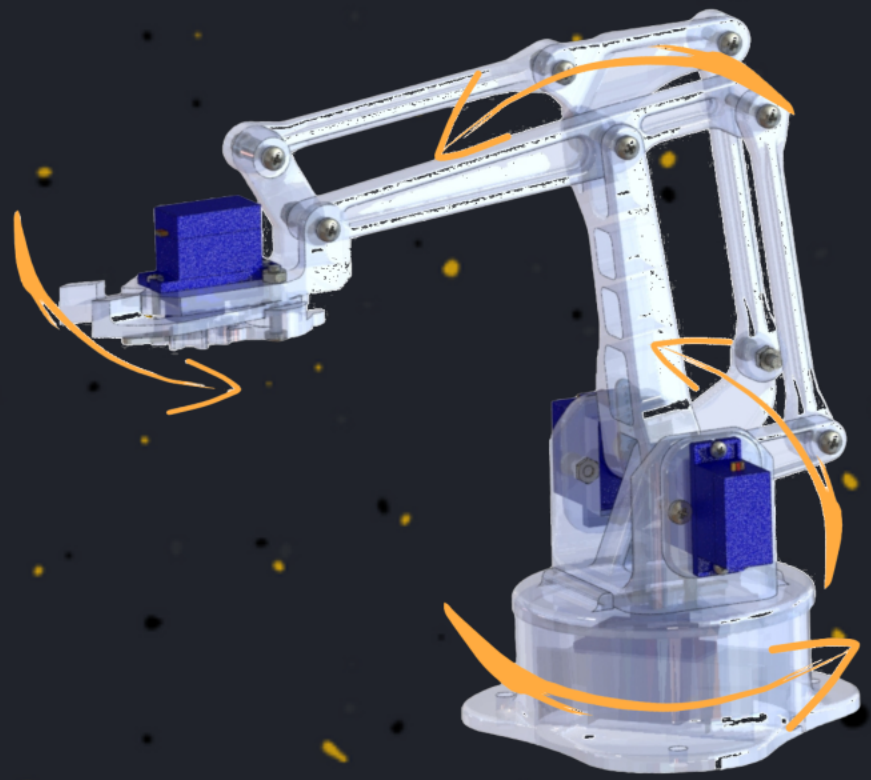

z
y
x

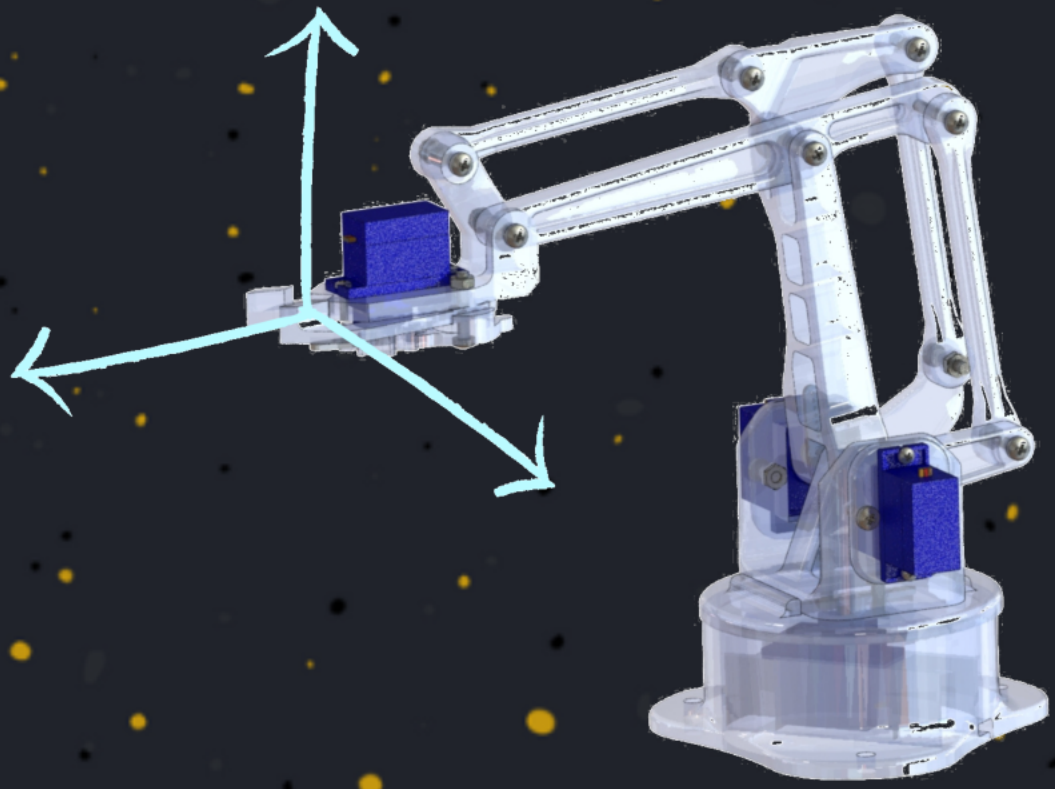

z
y
x
z
z
y
x
y
y
z
x
z
y
x
y
z
x


z
y
x
z
z
y
x
y
y
z
x
z
y
x
y
z
x
y
z
z
x
y
x
z


$T_R^w = T_1^W(q_1) \cdot T_2^1(q_2) \cdot T_3^2(q_3) \cdot T_R^3$
z_W
y_W
x_W
z_1
y_1
x_1
z_2
y_2
x_2
z_3
y_3
x_3
z_R
y_R
x_R
T_1^w
T_2^1
T_3^2
T_R^3
T_R^w

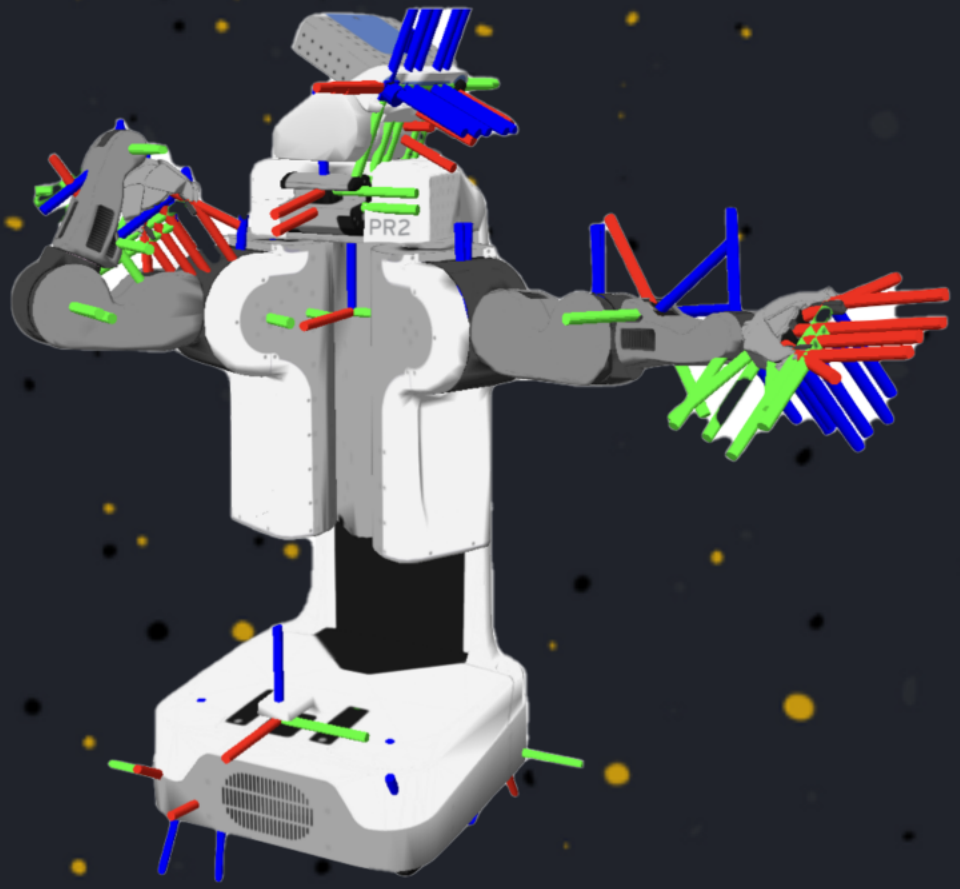
Forward





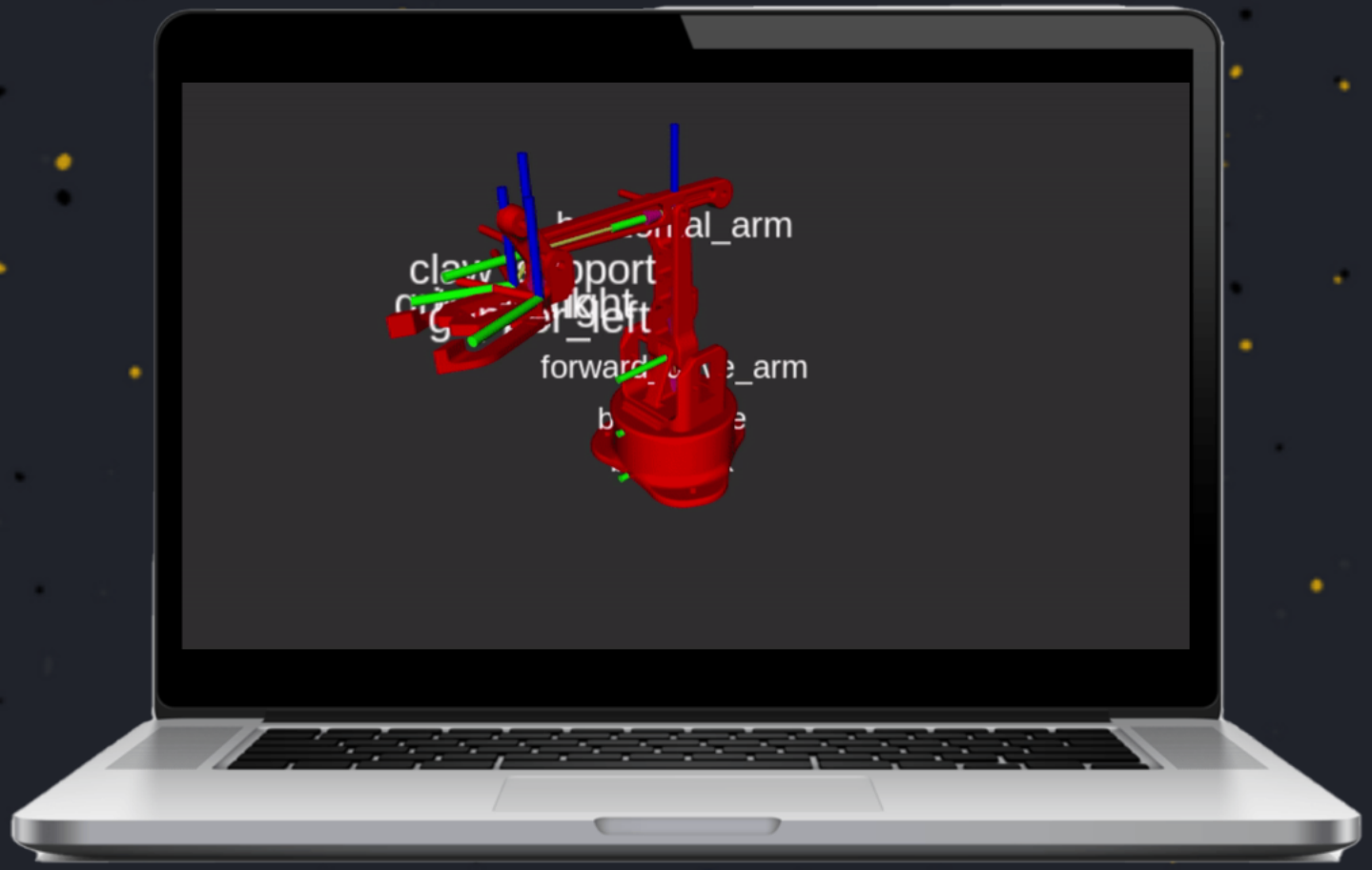
Inverse

TF2 Library

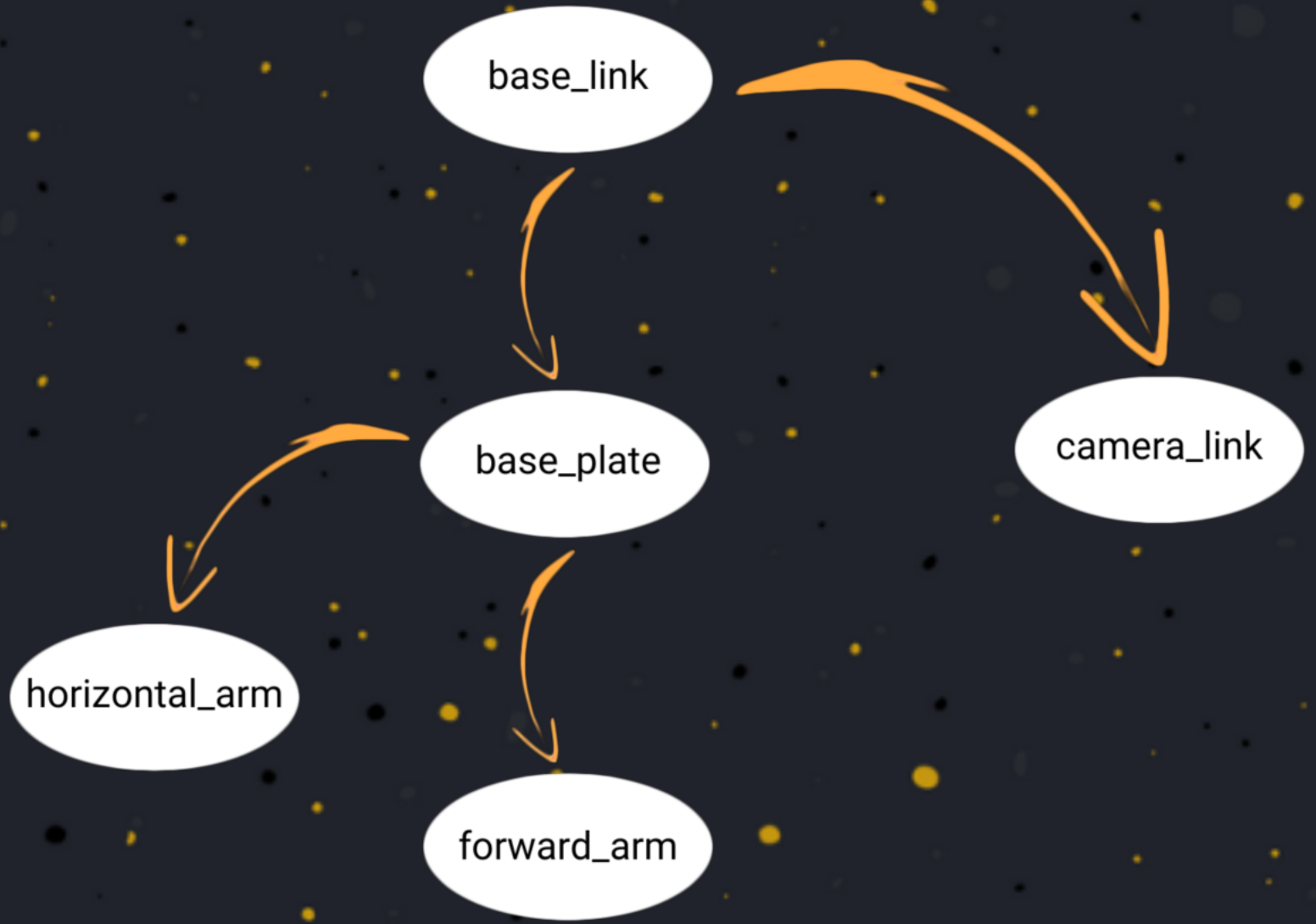


TF2 Library

TF2 Library



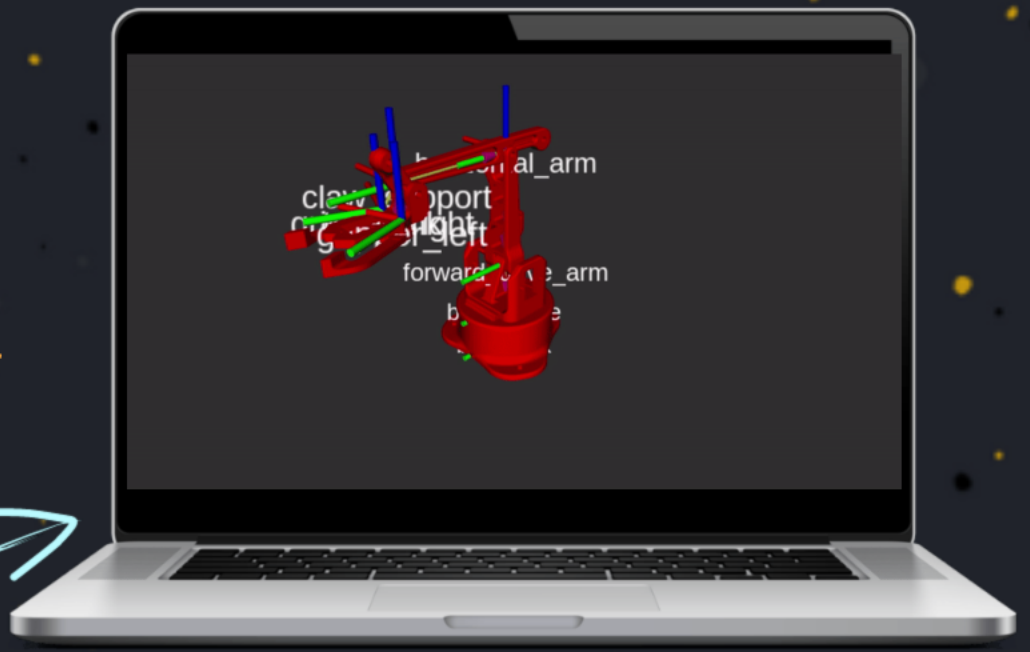

horizontal_arm
forward_drive_arm
base_plate


base_link
base_plate
camera_link
horizontal_arm
forward_arm

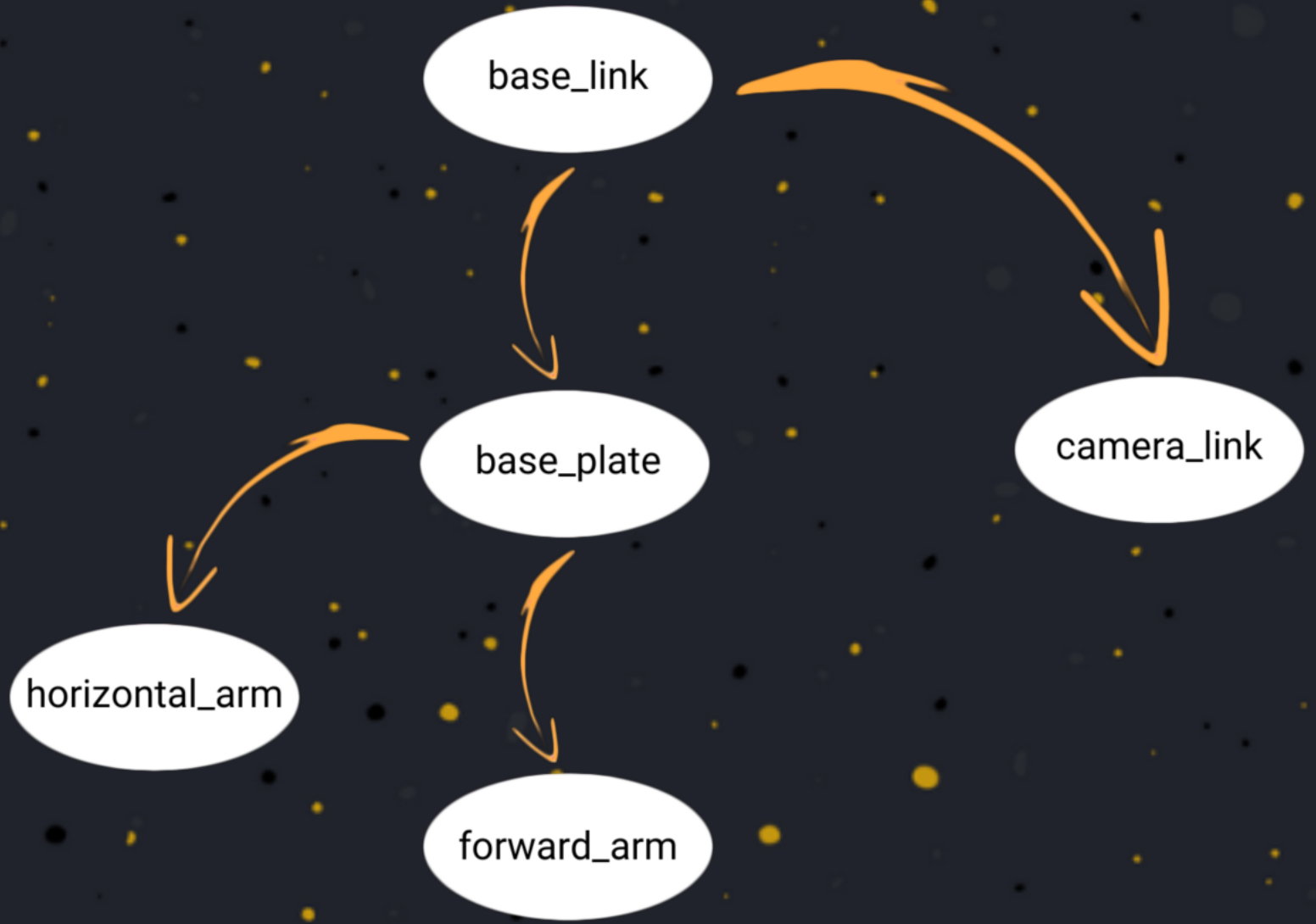
```xml
<robot name="arduinobot">

    <link name="world"/>

    <link name="base_link">
        <visual>
            <origin rpy="0 0 0" xyz="-0.5 -0.5 0"/>
            <geometry>
                <mesh filename="basement.STL"/>
            </geometry>
        </visual>
        <collision>
            <origin rpy="0 0 0" xyz="-0.5 -0.5 0"/>
            <geometry>
                <mesh filename="basement.STL"/>
            </geometry>
        </collision>
    </link>

    <joint name="virtual_joint" type="fixed">
        <parent link="world"/>
        <child link="base_link"/>
        <origin xyz="0 0 0" rpy="0 0 0"/>
    </joint>

</robot>
```

robot_state_publisher

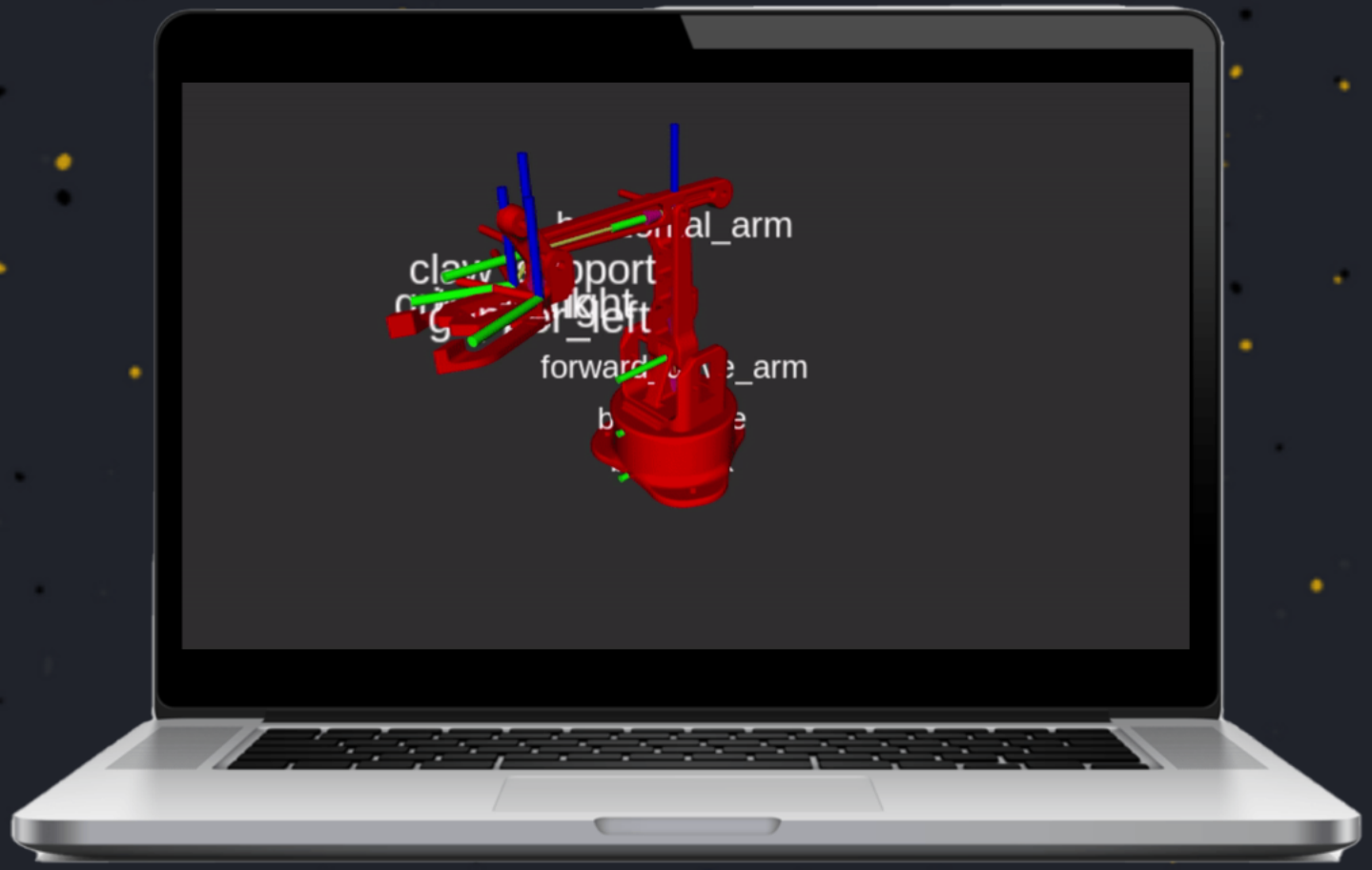
/tf



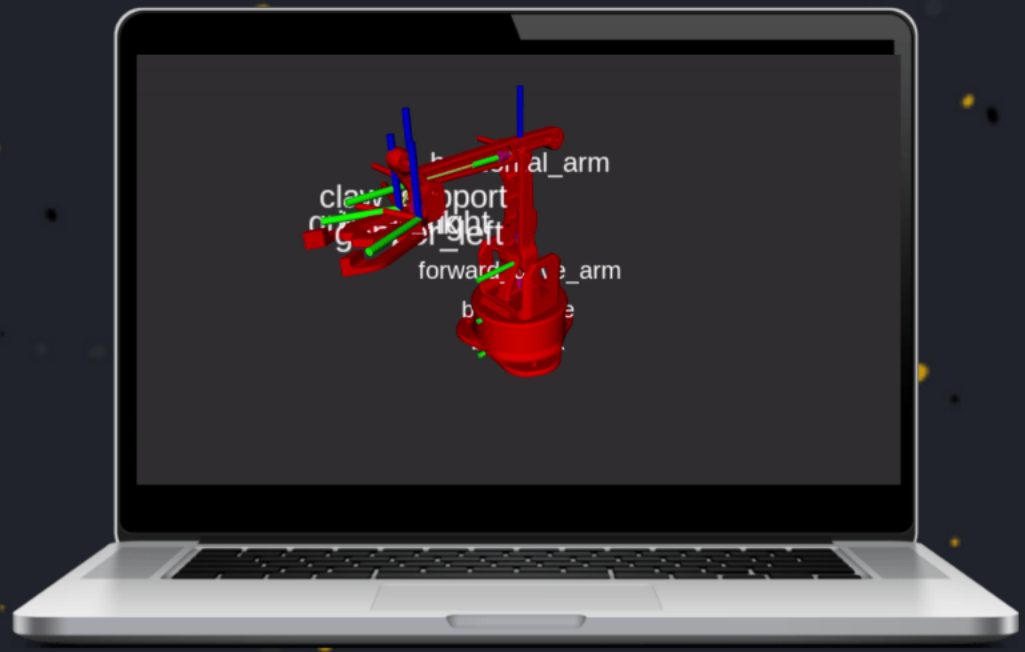

horizontal_arm
forward_drive_arm
base_plate


base_link
base_plate
camera_link
horizontal_arm
forward_arm

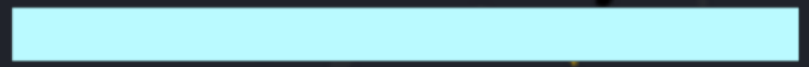
TF2 Library



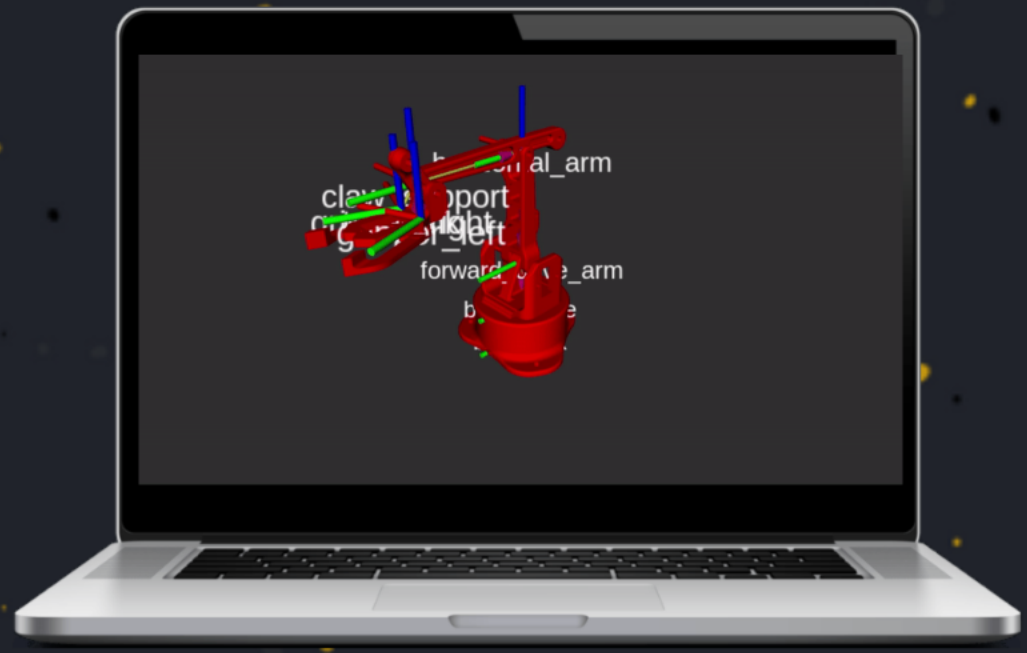
```xml
<robot name="arduinobot">

    <link name="world"/>

    <link name="base_link">
        <visual>
            <origin rpy="0 0 0" xyz="-0.5 -0.5 0"/>
            <geometry>
                <mesh filename="basement.STL"/>
            </geometry>
        </visual>
        <collision>
            <origin rpy="0 0 0" xyz="-0.5 -0.5 0"/>
            <geometry>
                <mesh filename="basement.STL"/>
            </geometry>
        </collision>
    </link>

    <joint name="virtual_joint" type="fixed">
        <parent link="world"/>
        <child link="base_link"/>
        <origin xyz="0 0 0" rpy="0 0 0"/>
    </joint>

    <joint name ="joint_1" type="revolute">
        <parent link="base_link"/>
        <child link = "base_plate"/>
        <origin xyz="0 0 0.307"/>
        <axis xyz="0 0 1"/>
    </joint>

</robot>
```



robot_state_publisher



```
<robot name="arduinobot">

    <link name="world"/>

    <link name="base_link">
        <visual>
            <origin rpy="0 0 0" xyz="-0.5 -0.5 0"/>
            <geometry>
                <mesh filename="basement.STL"/>
            </geometry>
        </visual>
        <collision>
            <origin rpy="0 0 0" xyz="-0.5 -0.5 0"/>
            <geometry>
                <mesh filename="basement.STL"/>
            </geometry>
        </collision>
    </link>

    <joint name="virtual_joint" type="fixed">
        <parent link="world"/>
        <child link="base_link"/>
        <origin xyz="0 0 0" rpy="0 0 0"/>
    </joint>

    <joint name ="joint_1" type="revolute">
        <parent link="base_link"/>
        <child link = "base_plate"/>
        <origin xyz="0 0 0.307"/>
        <axis xyz="0 0 1"/>
    </joint>

</robot>
```



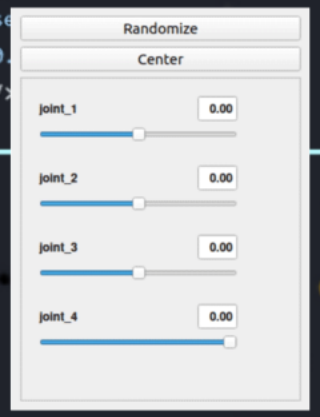
robot_state_publisher

/tf_static

```
<robot name="arduinobot">

    <link name="world"/>

    <link name="base_link">
        <visual>
            <origin rpy="0 0 0" xyz="-0.5 -0.5 0"/>
            <geometry>
                <mesh filename="basement.STL"/>
            </geometry>
        </visual>
        <collision>
            <origin rpy="0 0 0" xyz="-0.5 -0.5 0"/>
            <geometry>
                <mesh filename="basement.STL"/>
            </geometry>
        </collision>
    </link>
```

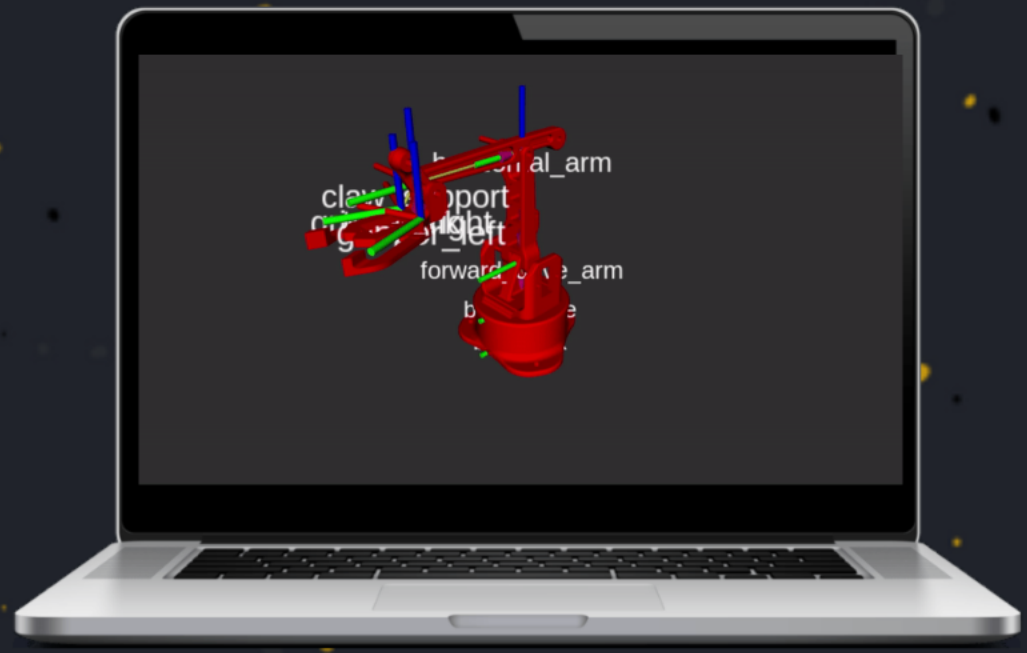
```
    <joint name="virtual_joint" type="fixed">
        <parent link="world"/>
        <child link="base_link"/>
        <origin xyz="0 0 0" rpy="0 0 0"/>
    </joint>
```

```
    <joint name ="joint_1" type="revolute">
        <parent link="base_link"/>
        <child link = "base
        <origin xyz="0 0 0.
        <axis xyz="0 0 1"/>
    </joint>

</robot>
```



robot_state_publisher

/tf_static

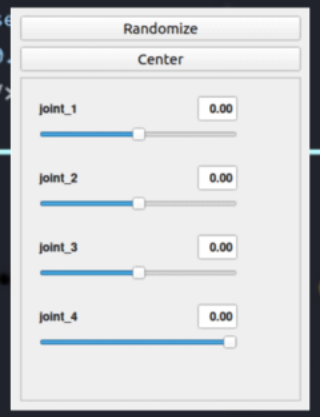


```
<robot name="arduinobot">

    <link name="world"/>

    <link name="base_link">
        <visual>
            <origin rpy="0 0 0" xyz="-0.5 -0.5 0"/>
            <geometry>
                <mesh filename="basement.STL"/>
            </geometry>
        </visual>
        <collision>
            <origin rpy="0 0 0" xyz="-0.5 -0.5 0"/>
            <geometry>
                <mesh filename="basement.STL"/>
            </geometry>
        </collision>
    </link>
```

```
<joint name="virtual_joint" type="fixed">
    <parent link="world"/>
    <child link="base_link"/>
    <origin xyz="0 0 0" rpy="0 0 0"/>
</joint>
```

```
<joint name ="joint_1" type="revolute">
    <parent link="base_link"/>
    <child link = "base
    <origin xyz="0 0 0.
    <axis xyz="0 0 1"/>
</joint>
```

```
</robot>
```

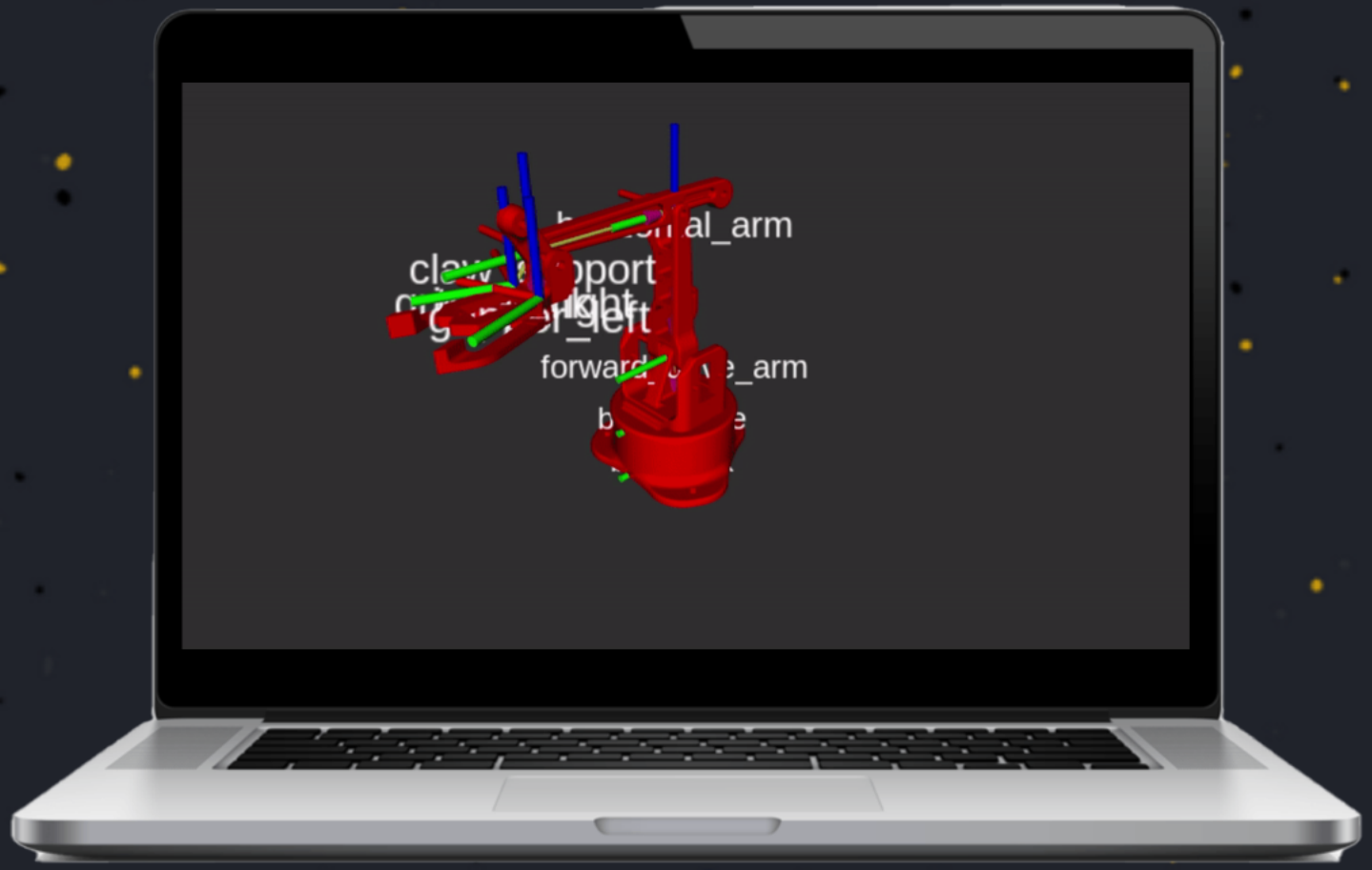


robot_state_publisher

/tf_static

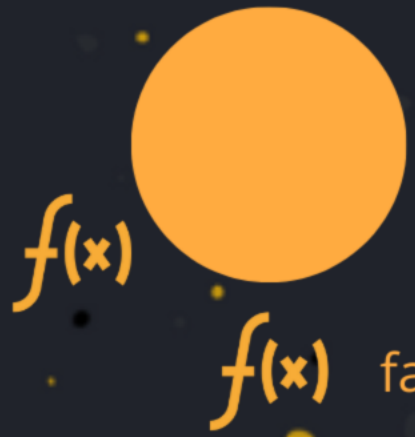
/tf

TF2 Library



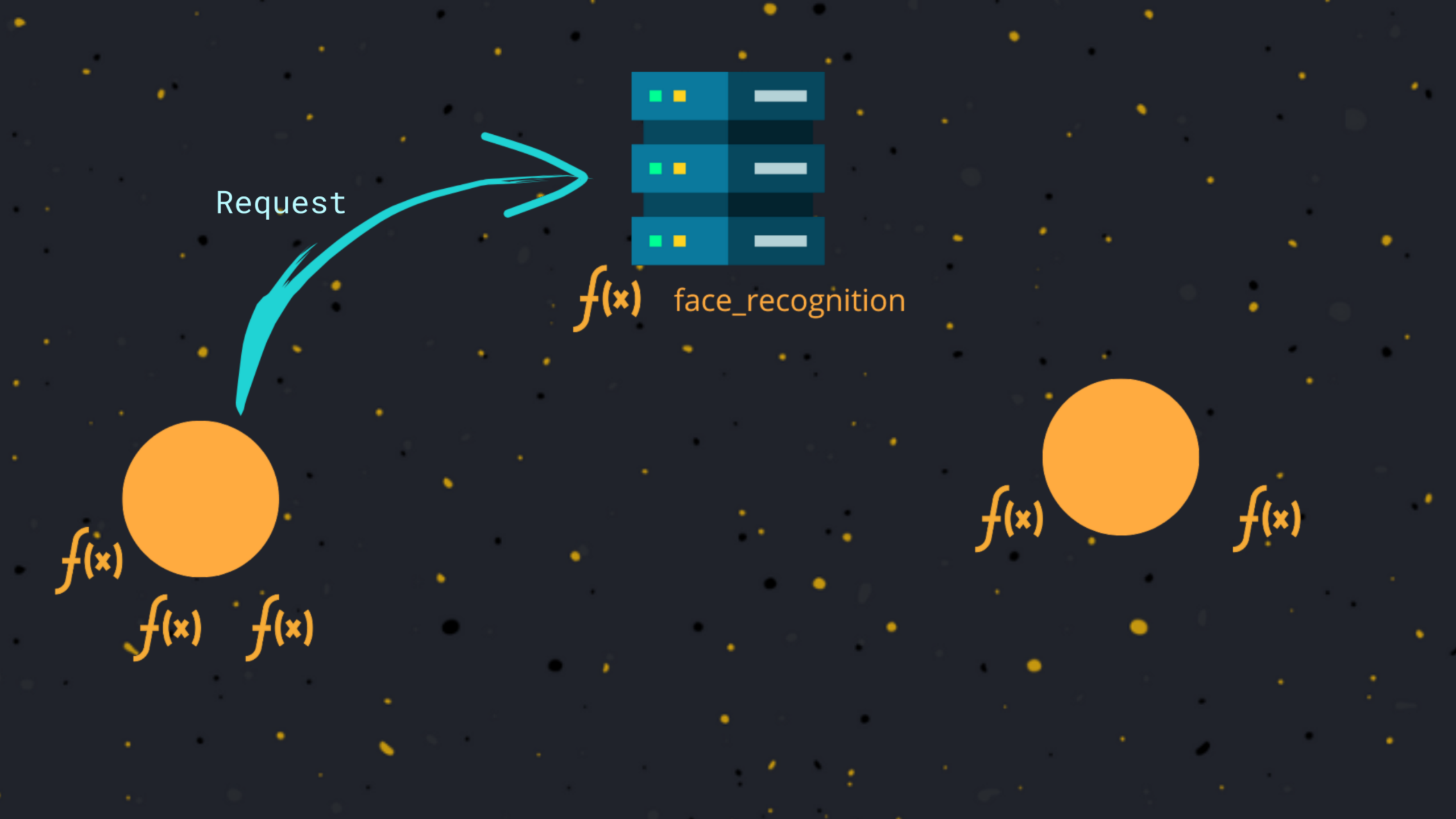
face_recognition

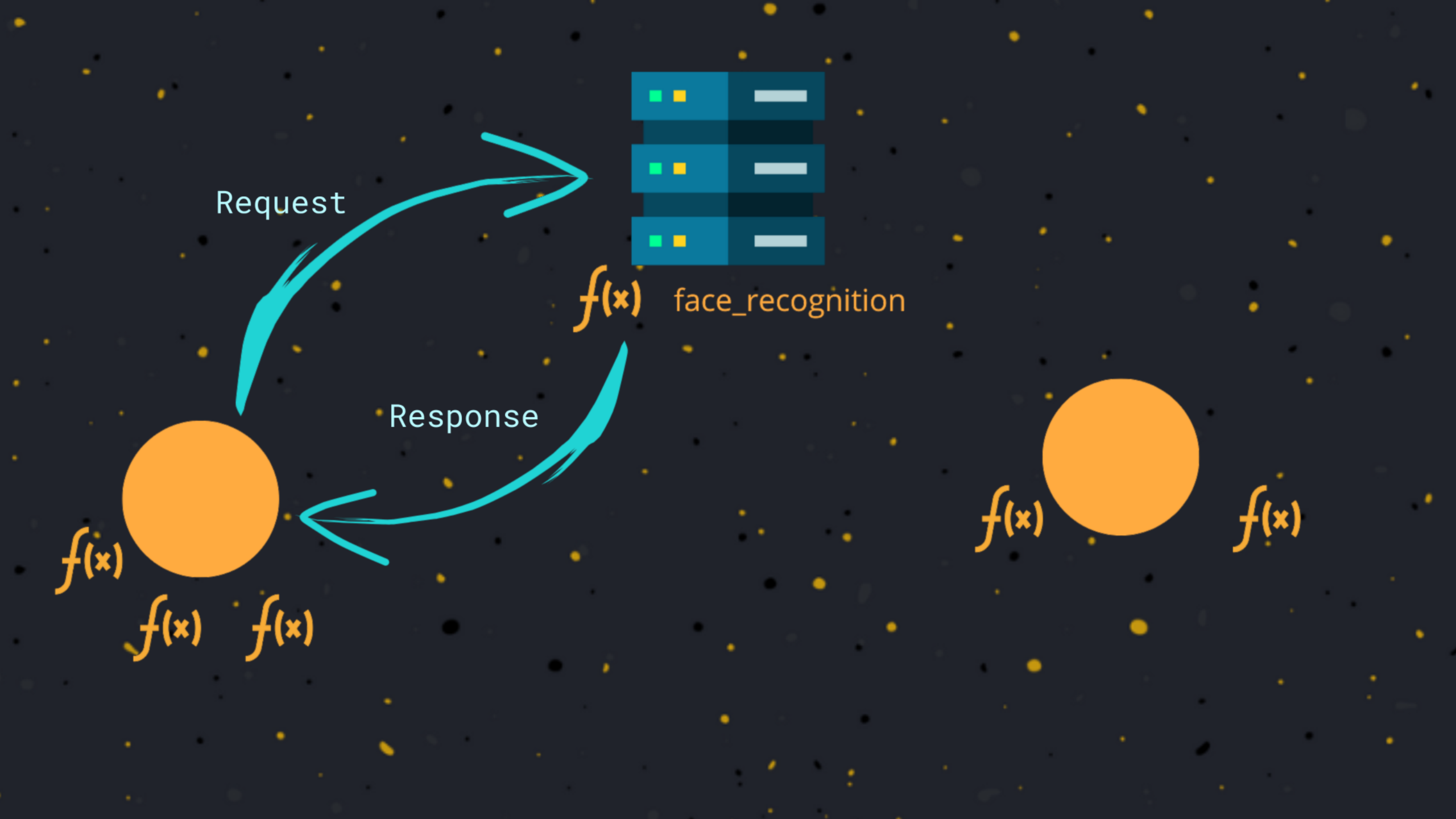
face_recognition

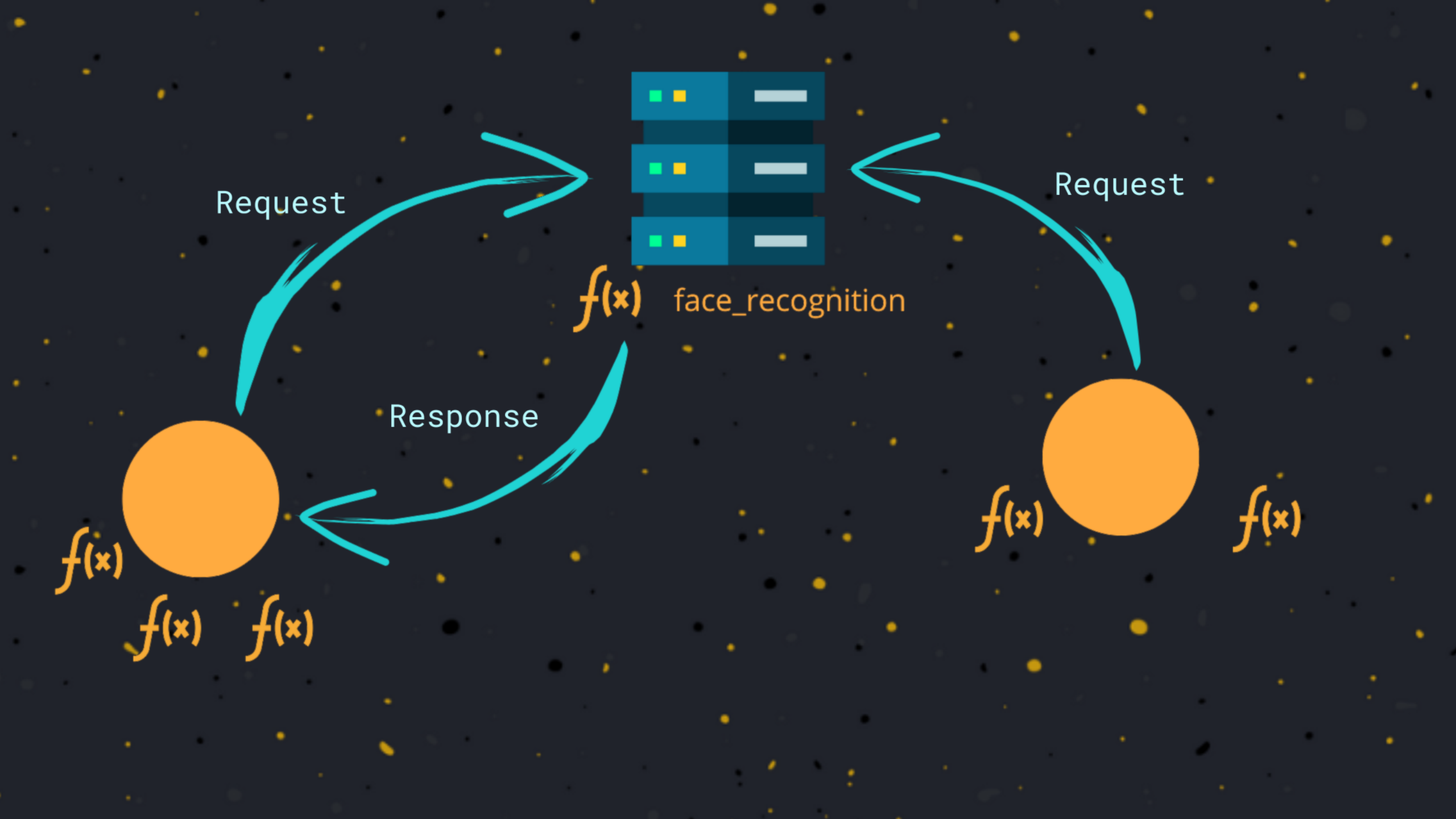


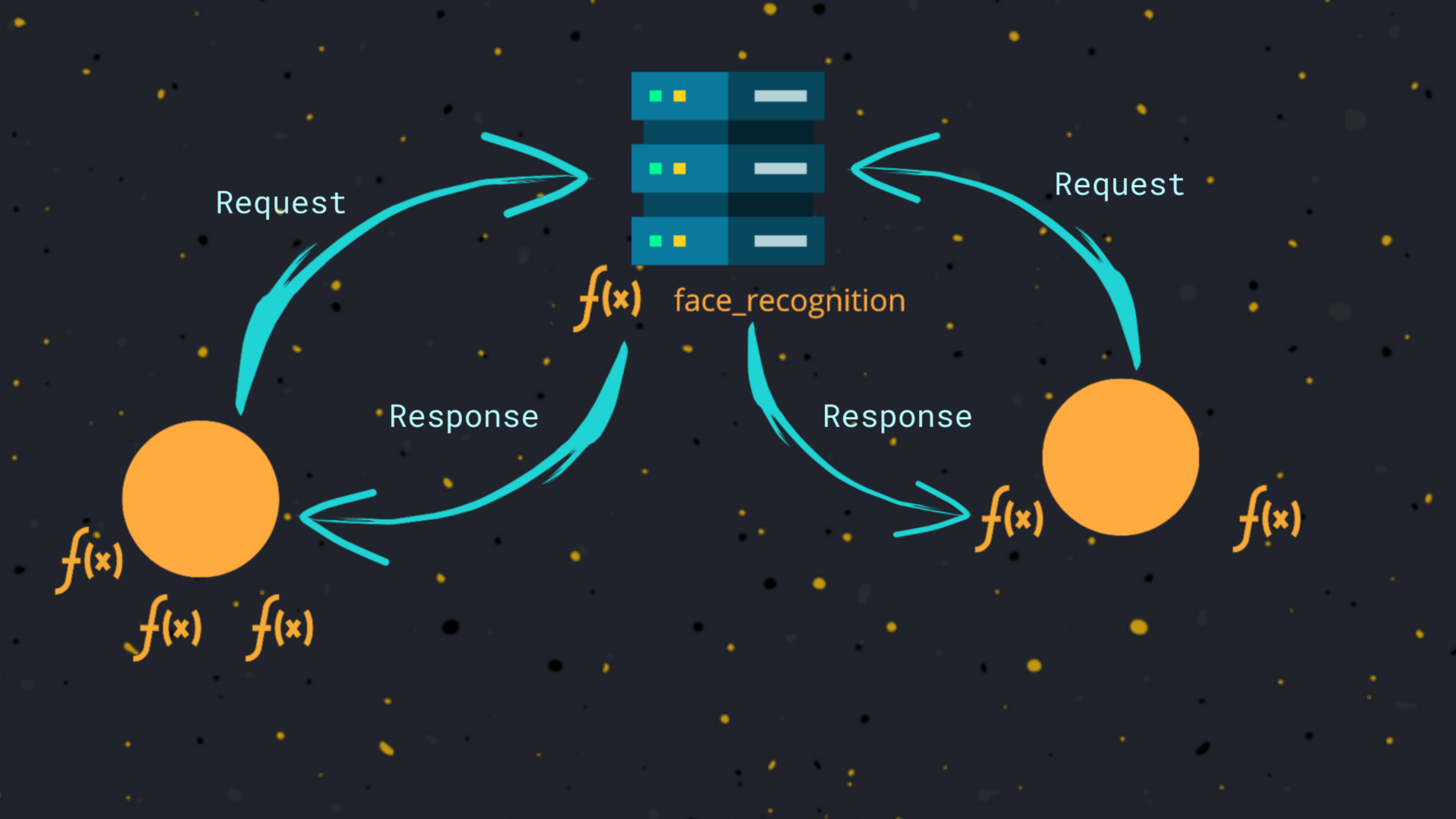
face_recognition

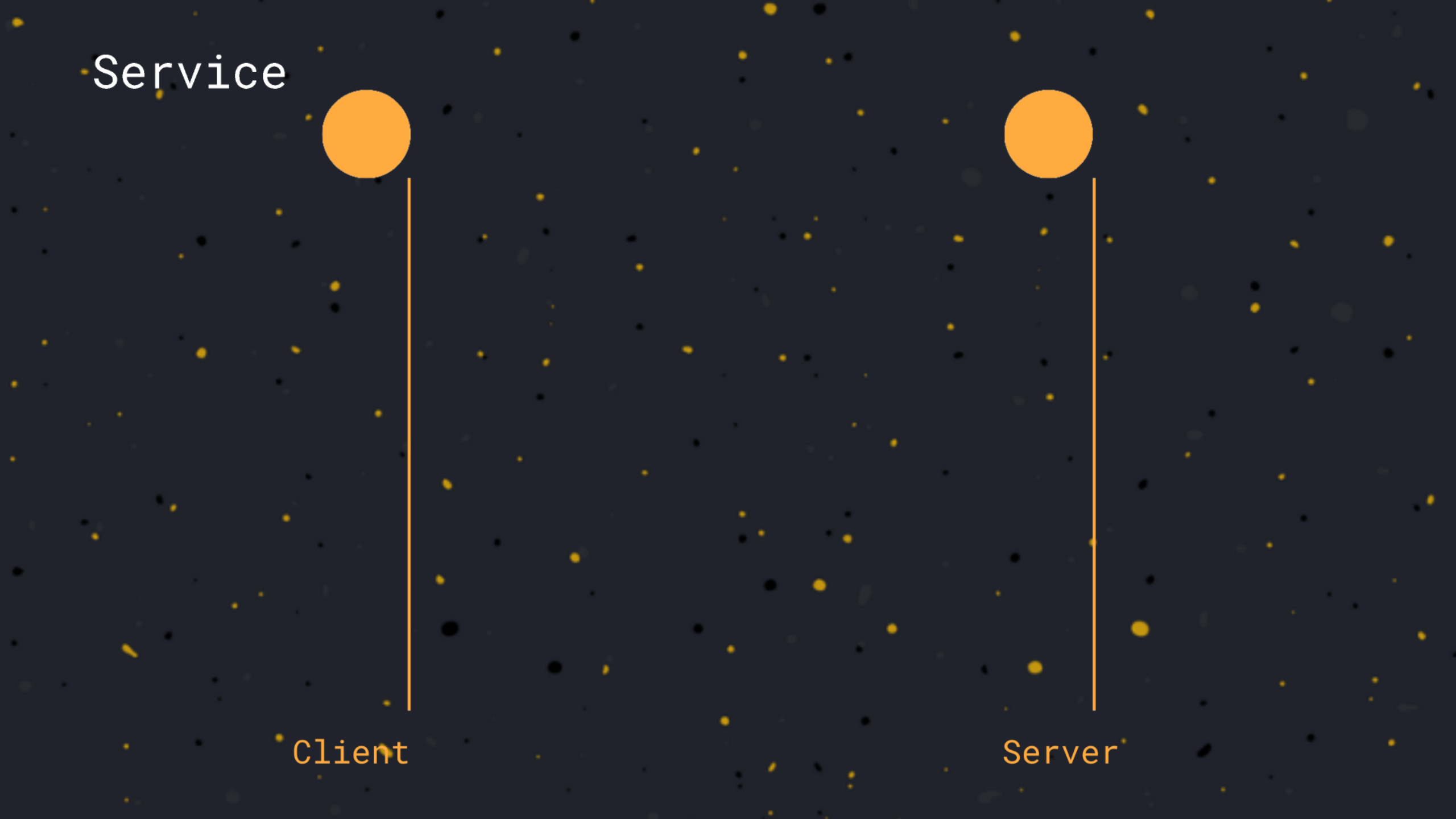

f(x)
face_recognition
f(x)
f(x)
f(x)
f(x)
f(x)

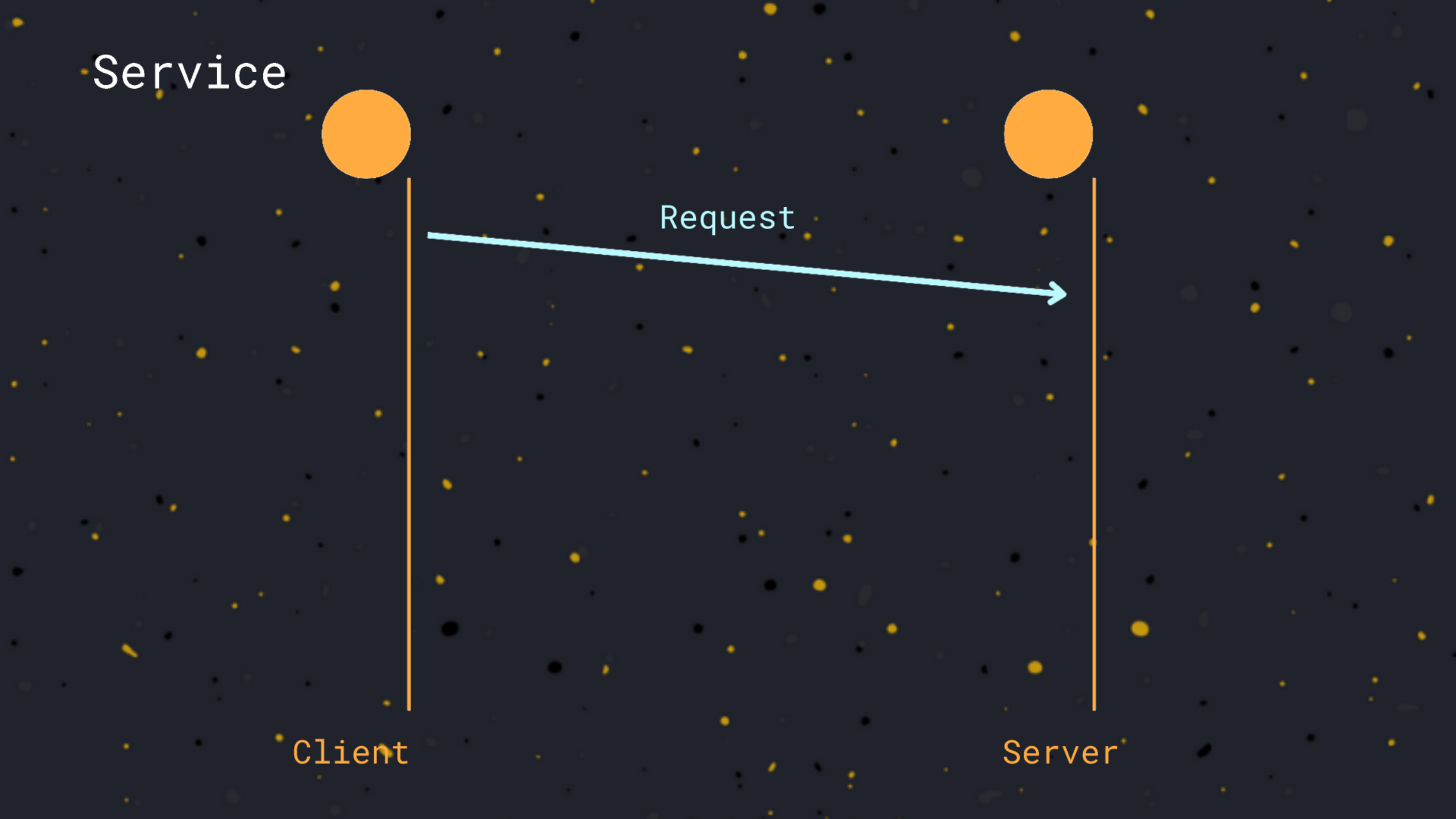

Request
f(x)
face_recognition
f(x)
f(x)
f(x)
f(x)
f(x)

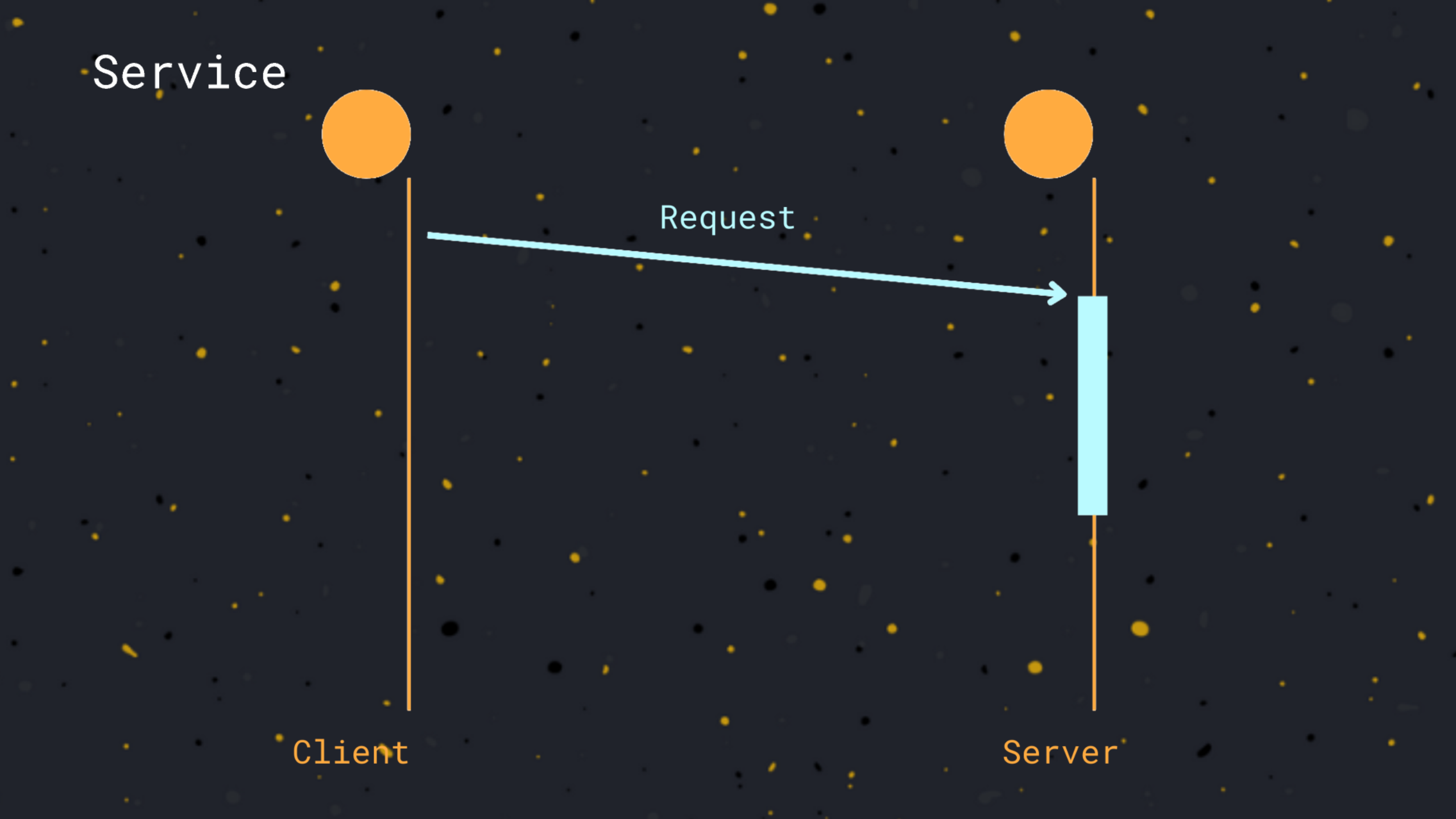

Request
Response
f(x)
face_recognition
f(x)
f(x)
f(x)
f(x)
f(x)


Request
Response
f(x)
face_recognition
Request
f(x)
f(x)
f(x)
f(x)
f(x)

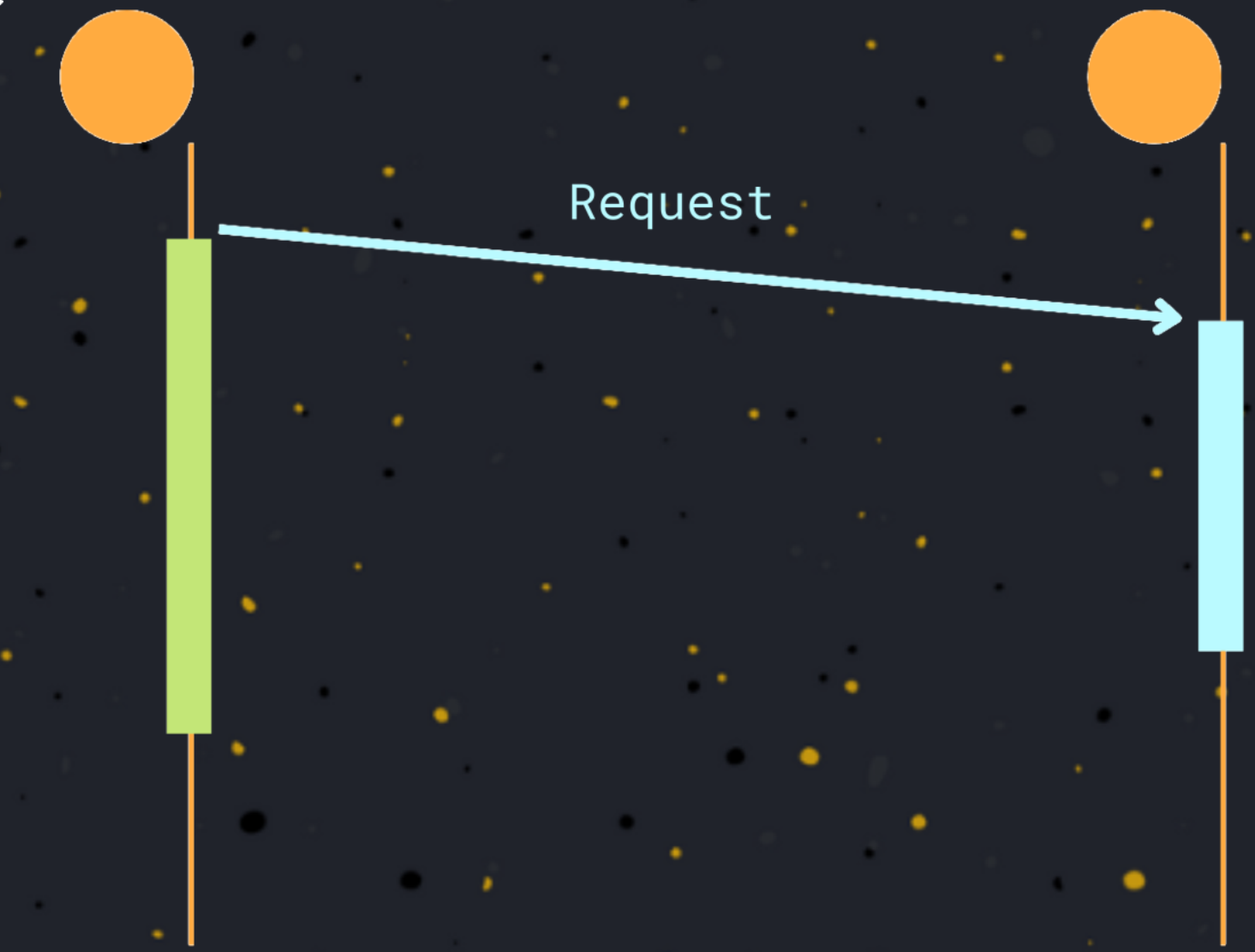

Request
Request
f(x)
face_recognition
Response
Response
f(x)
f(x)
f(x)
f(x)
f(x)


Service
Client
Server

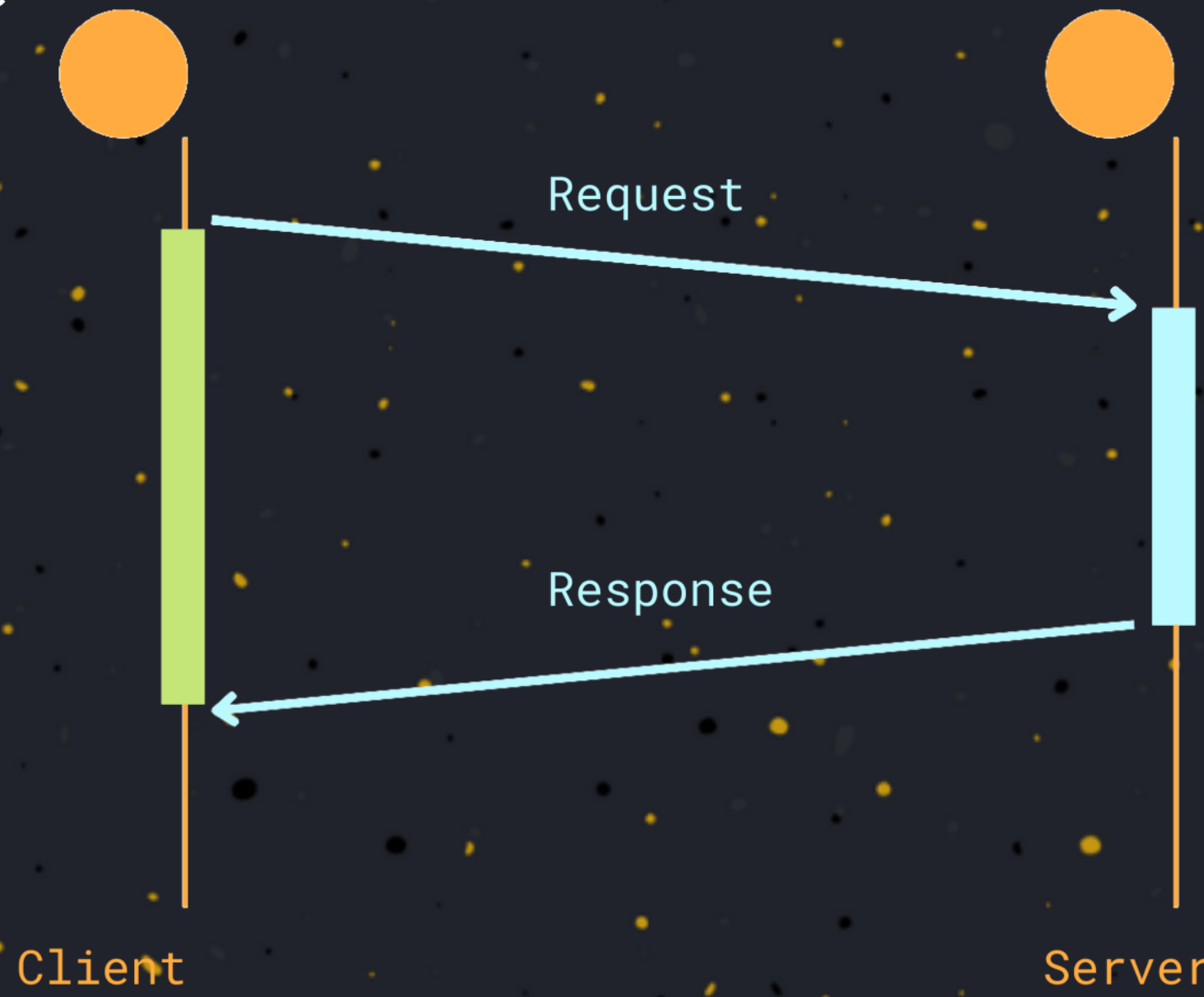

Service
Request
Client
Server

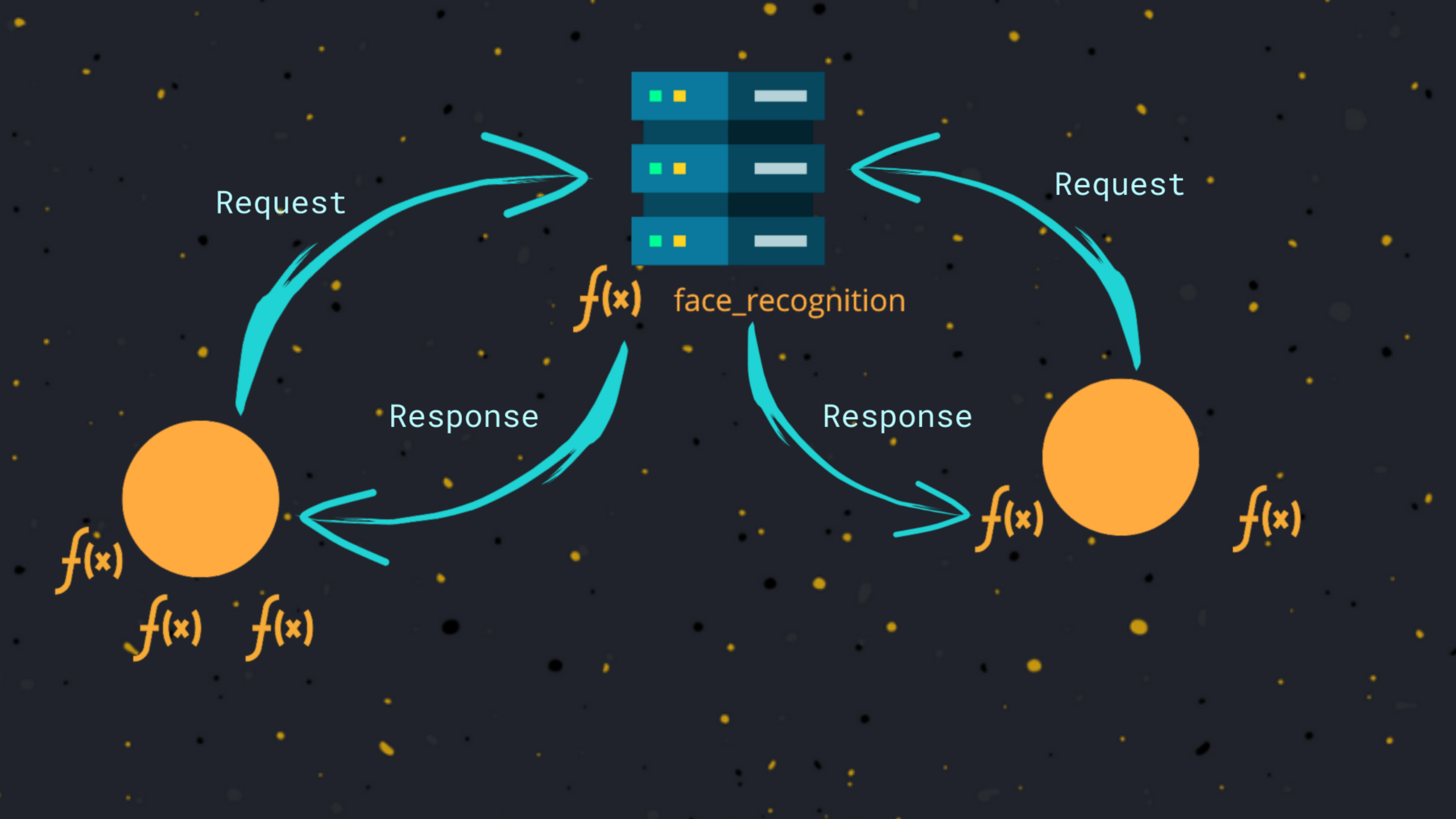

Service
Request
Client
Server

Service

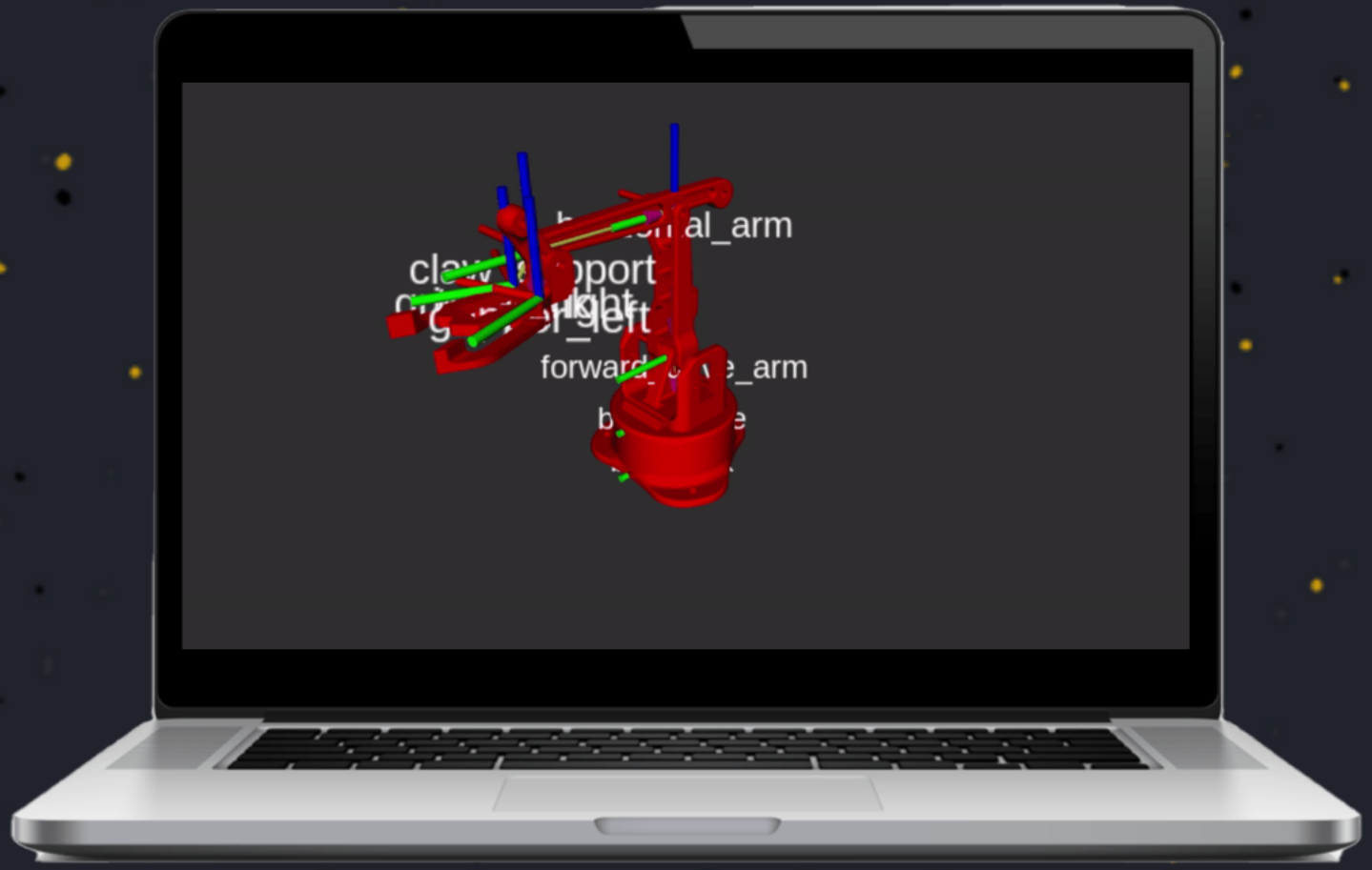


Service



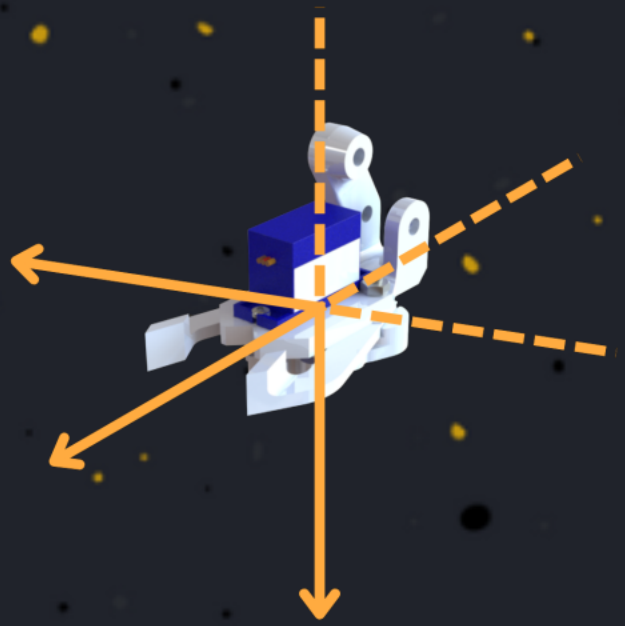

Request
Request
f(x)
face_recognition
Response
Response
f(x)
f(x)
f(x)
f(x)
f(x)

TF2 Library

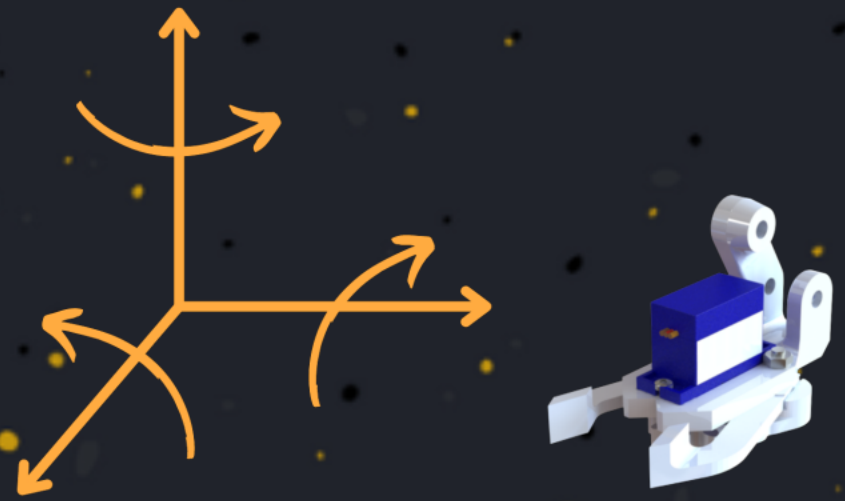


Angle Representations

Euler Angles



Quaternion



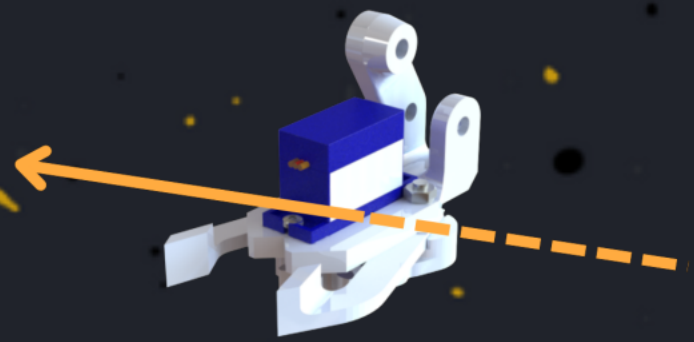
Euler Angles

Euler Angles

$$\begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Euler Angles

$$\begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Euler Angles

$$\begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\cdot \begin{bmatrix} \cos(\varphi) & 0 & \sin(\varphi) \\ 0 & 1 & 0 \\ -\sin(\varphi) & 0 & \cos(\varphi) \end{bmatrix}$$

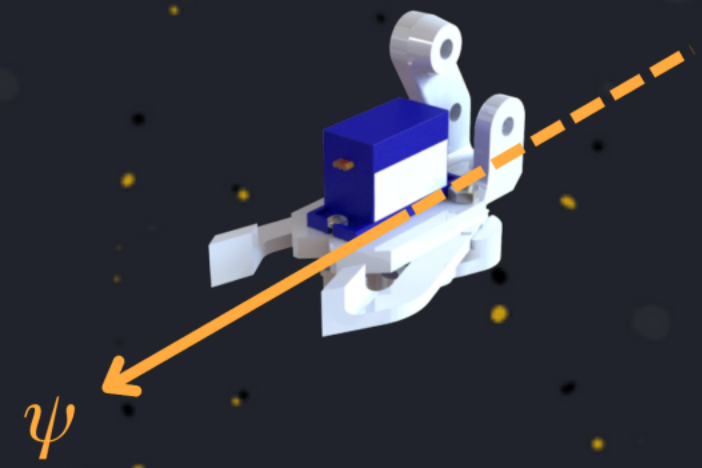
Euler Angles

$$\begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\cdot \begin{bmatrix} \cos(\varphi) & 0 & \sin(\varphi) \\ 0 & 1 & 0 \\ -\sin(\varphi) & 0 & \cos(\varphi) \end{bmatrix} \cdot$$

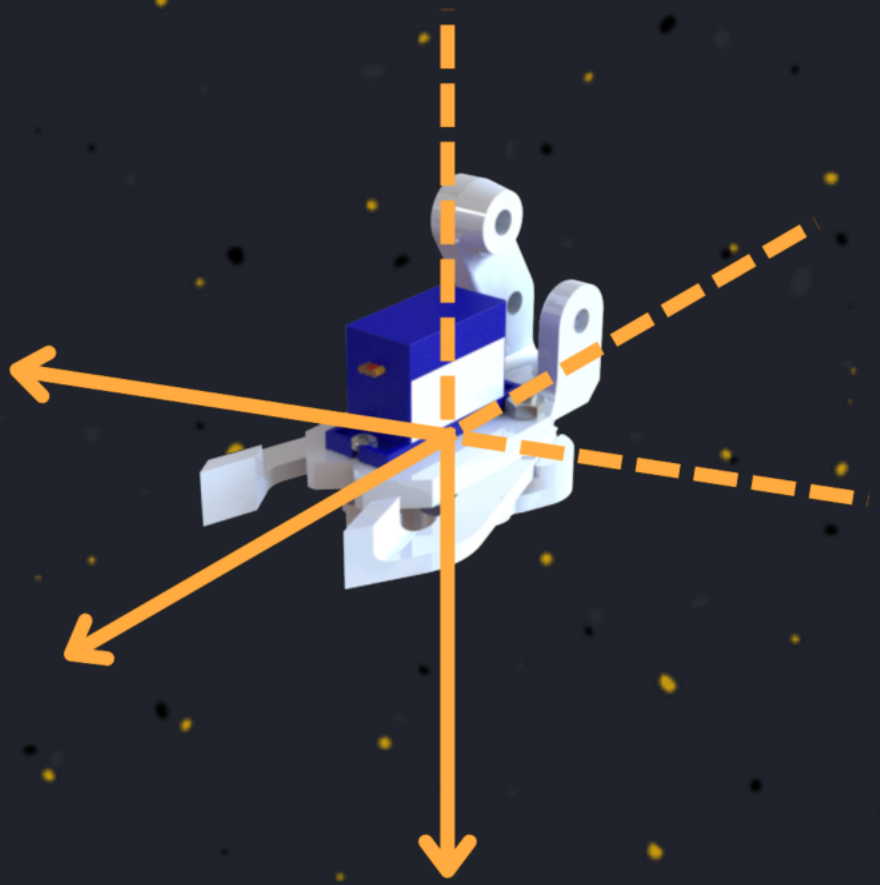
Euler Angles

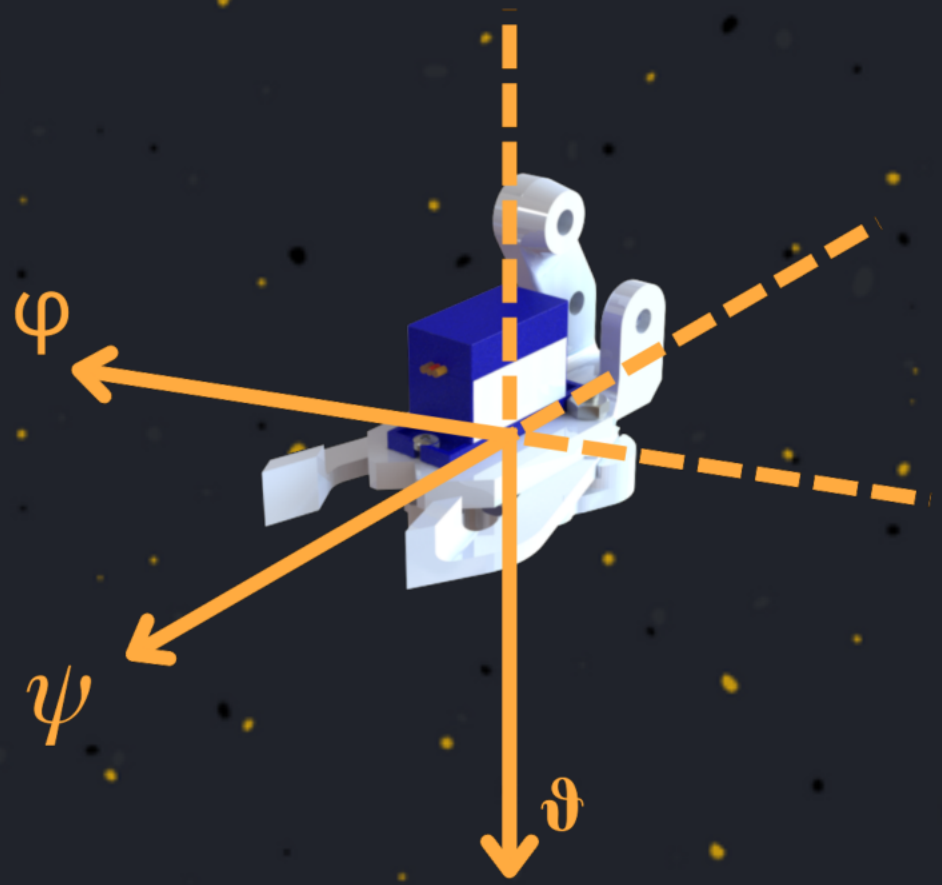
$$\begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

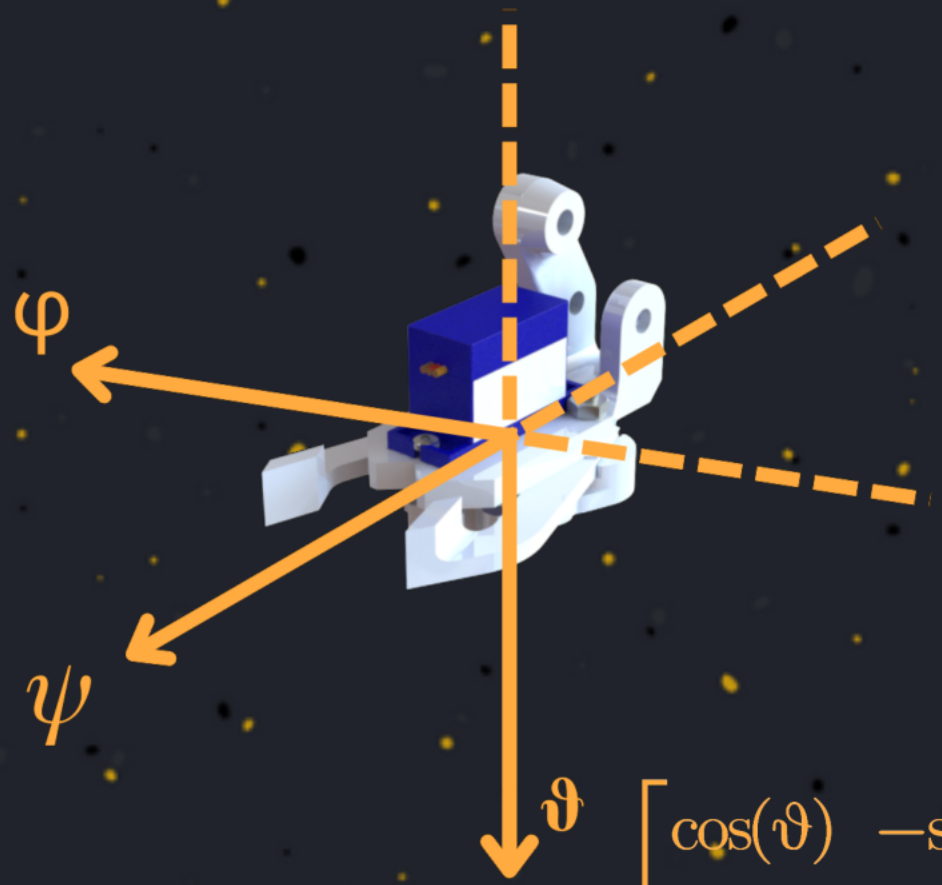


$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix}$$

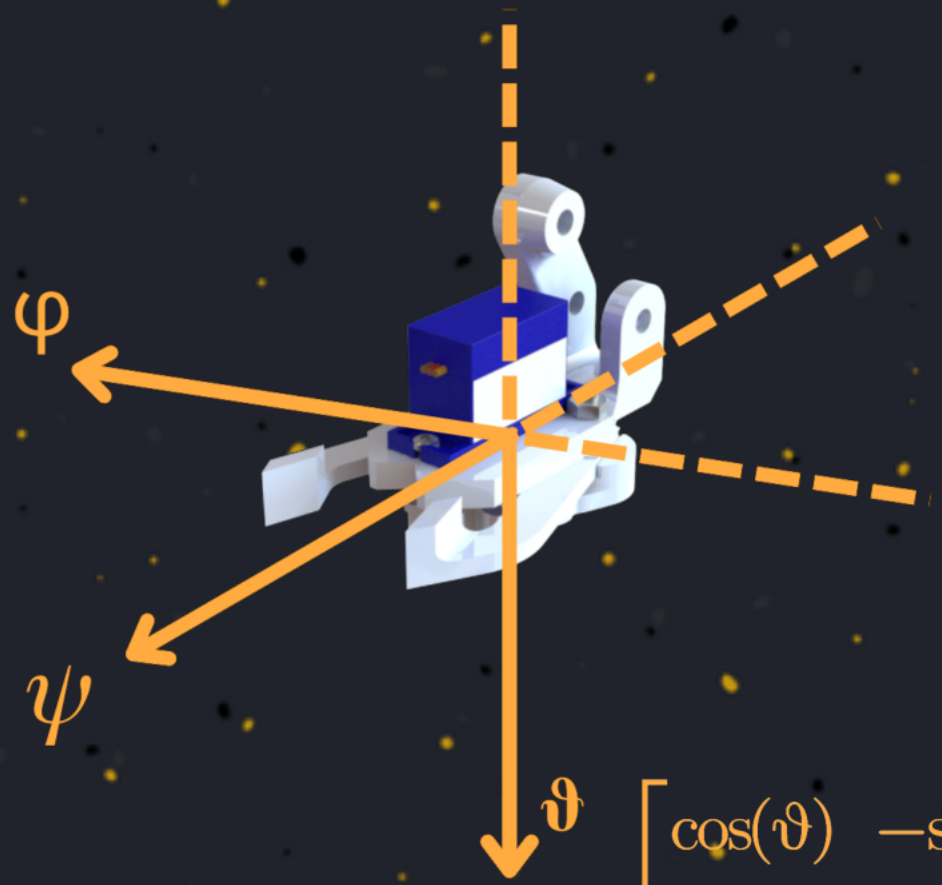
$$\begin{bmatrix} \cos(\varphi) & 0 & \sin(\varphi) \\ 0 & 1 & 0 \\ -\sin(\varphi) & 0 & \cos(\varphi) \end{bmatrix}$$




φ
ψ
ϑ

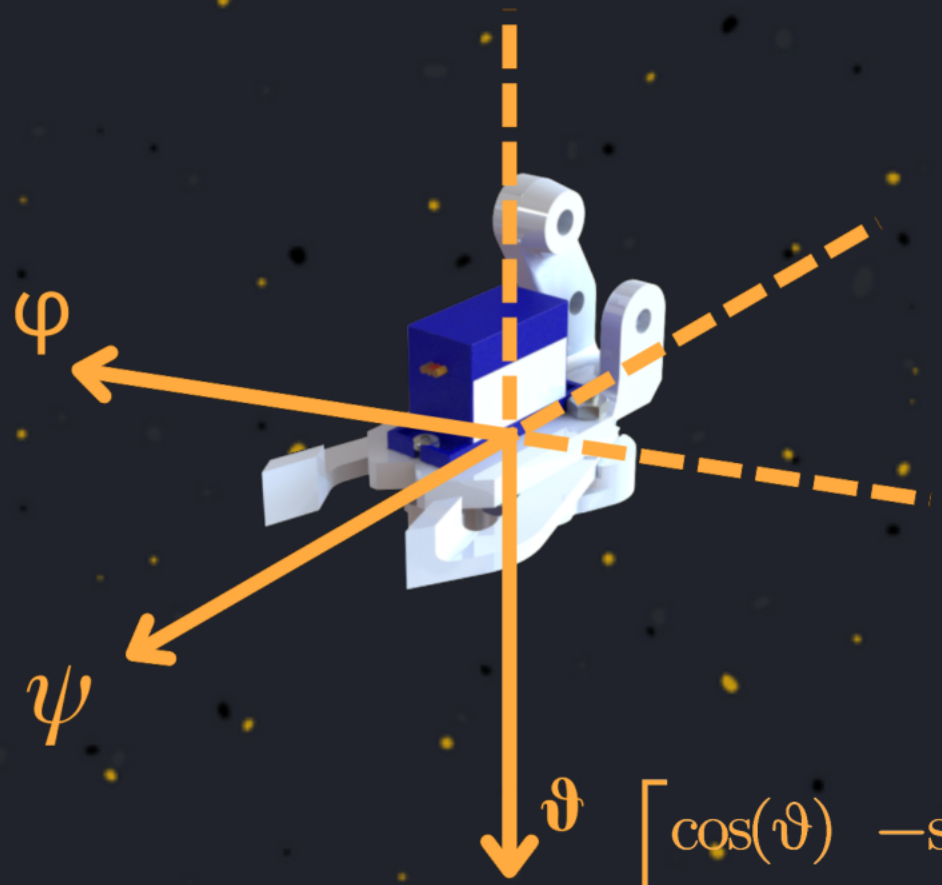


$$\begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\varphi) & 0 & \sin(\varphi) \\ 0 & 1 & 0 \\ -\sin(\varphi) & 0 & \cos(\varphi) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix}$$



$$\begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\varphi) & 0 & \sin(\varphi) \\ 0 & 1 & 0 \\ -\sin(\varphi) & 0 & \cos(\varphi) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix}$$

Yaw Pitch Roll



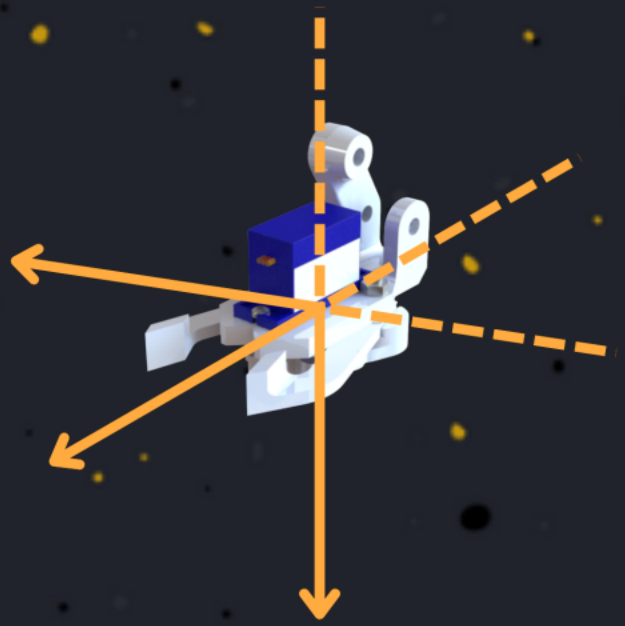
$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$\begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\varphi) & 0 & \sin(\varphi) \\ 0 & 1 & 0 \\ -\sin(\varphi) & 0 & \cos(\varphi) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix}$$

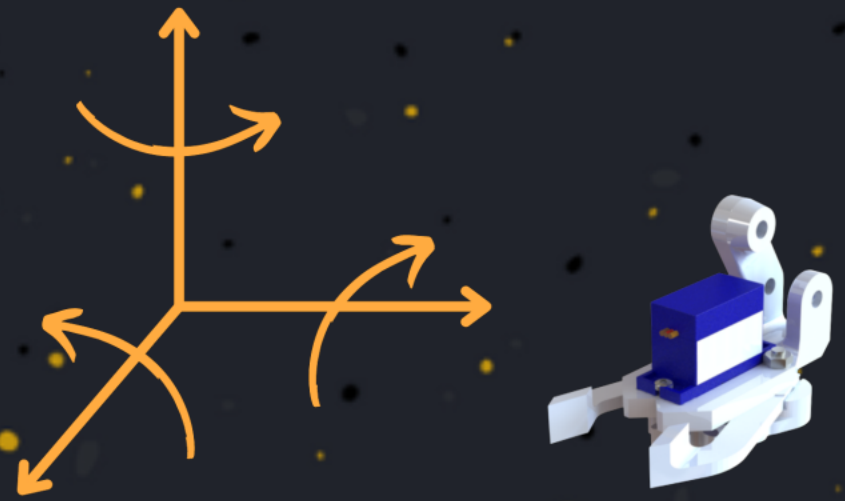
Yaw Pitch Roll

Angle Representations

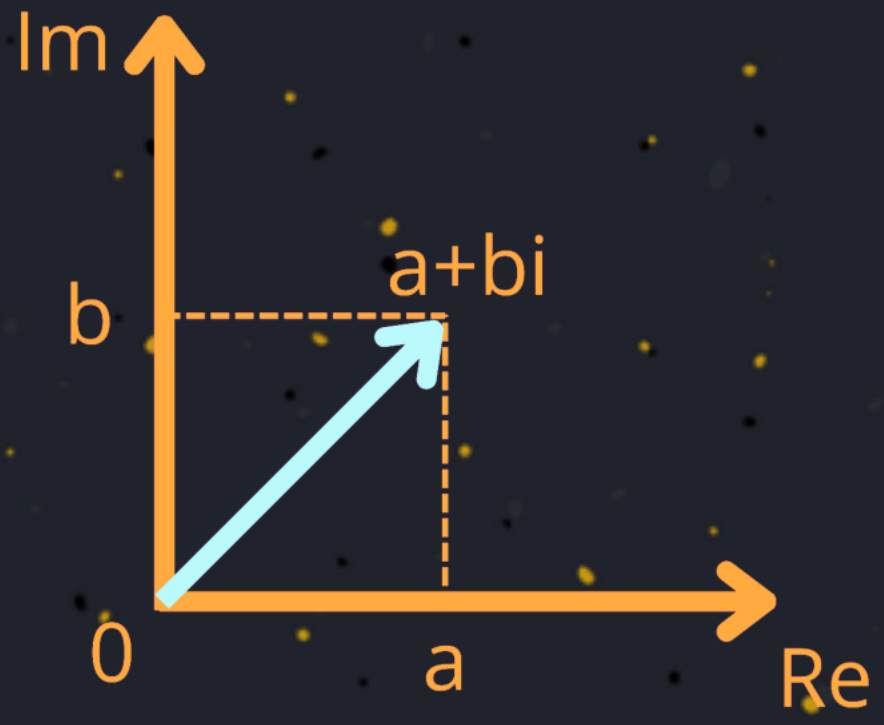
Euler Angles



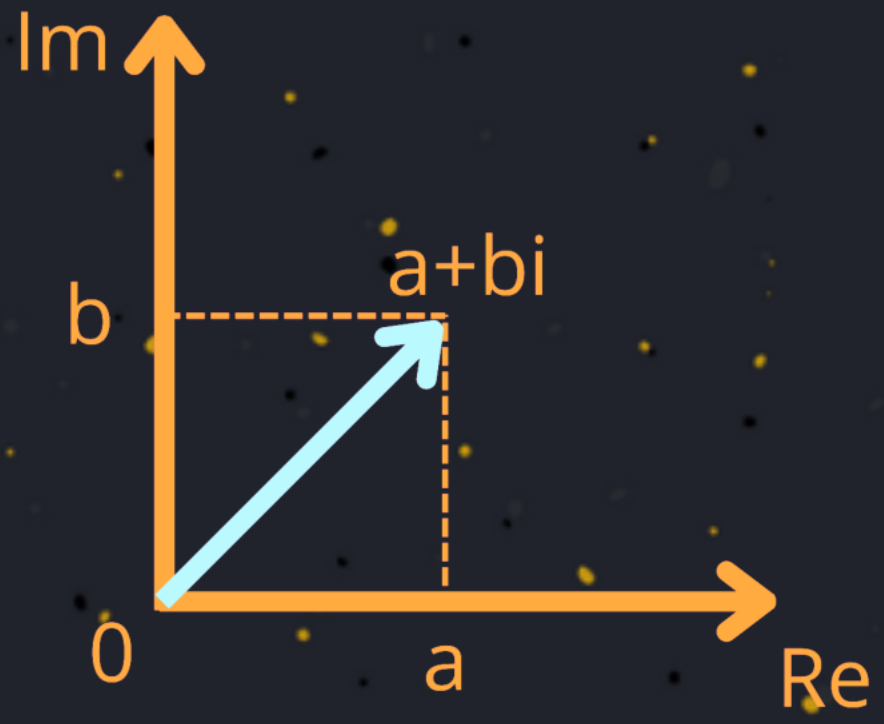
Quaternion



Quaternion

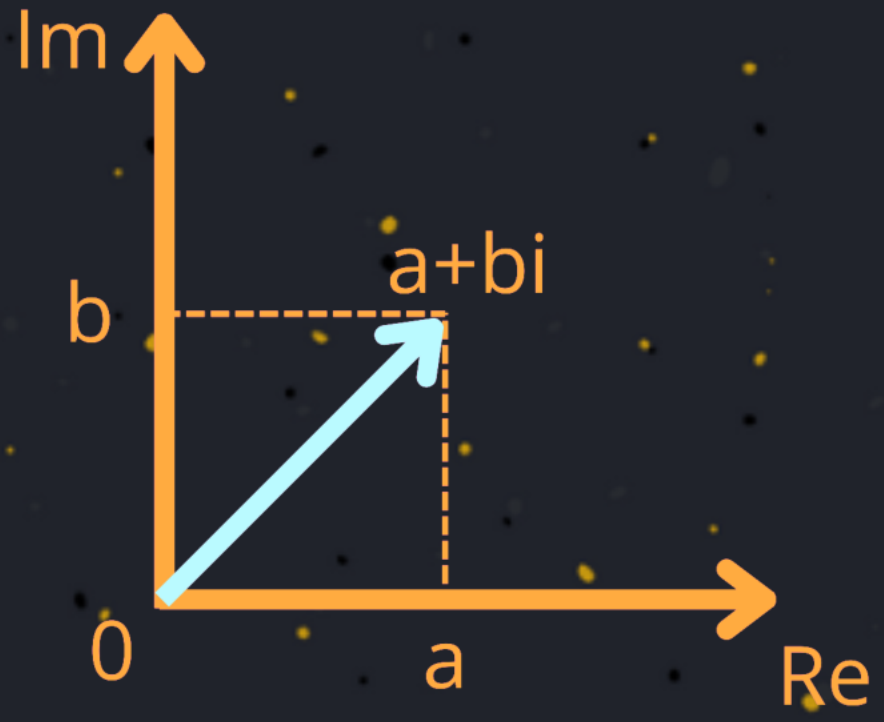


Quaternion



$$q = a + bi + cj + dk$$

Quaternion

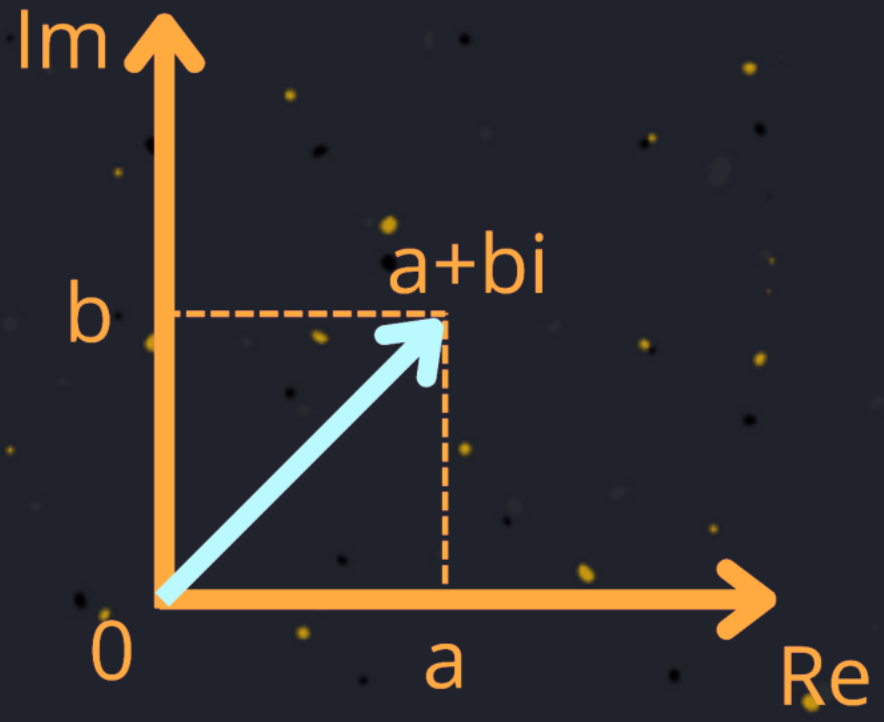


$$q = a + bi + cj + dk$$

Unitary

$$a^2 + b^2 + c^2 + d^2 = 1$$

Quaternion



$$q = a + bi + cj + dk$$

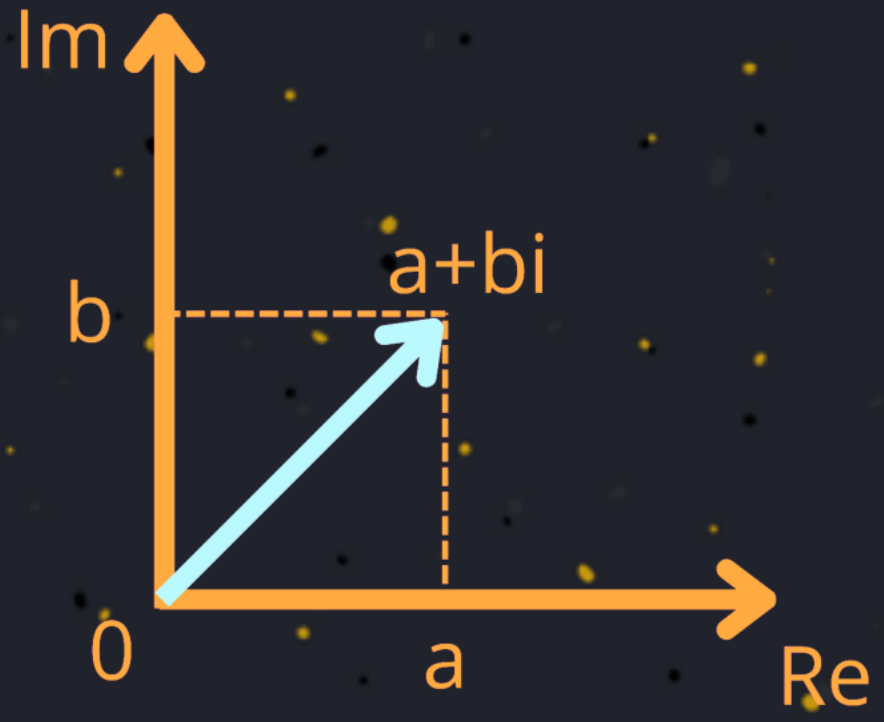
Unitary

$$a^2 + b^2 + c^2 + d^2 = 1$$

Rotation Composition

$$q_1 = \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \end{bmatrix} \qquad q_2 = \begin{bmatrix} a_2 \\ b_2 \\ c_2 \\ d_2 \end{bmatrix}$$

Quaternion



$$q = a + bi + cj + dk$$

Unitary

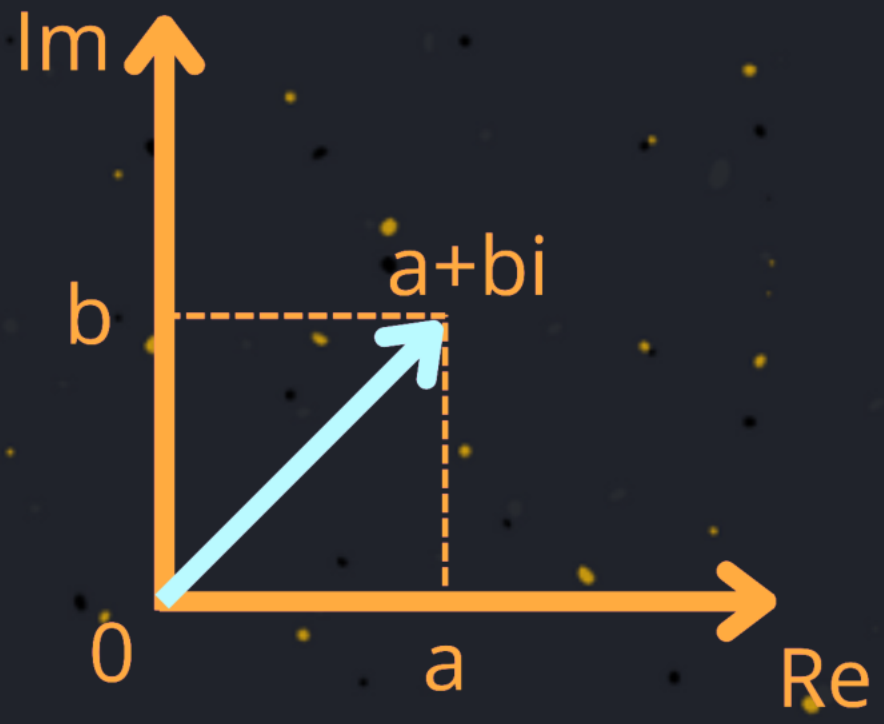
$$a^2 + b^2 + c^2 + d^2 = 1$$

Rotation Composition

$$q_1 = \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \end{bmatrix} \qquad q_2 = \begin{bmatrix} a_2 \\ b_2 \\ c_2 \\ d_2 \end{bmatrix}$$

$$q_1 \cdot q_2$$

Quaternion



$$q = a + bi + cj + dk$$

Unitary

$$a^2 + b^2 + c^2 + d^2 = 1$$

Rotation Composition

$$q_1 = \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \end{bmatrix} \qquad q_2 = \begin{bmatrix} a_2 \\ b_2 \\ c_2 \\ d_2 \end{bmatrix}$$

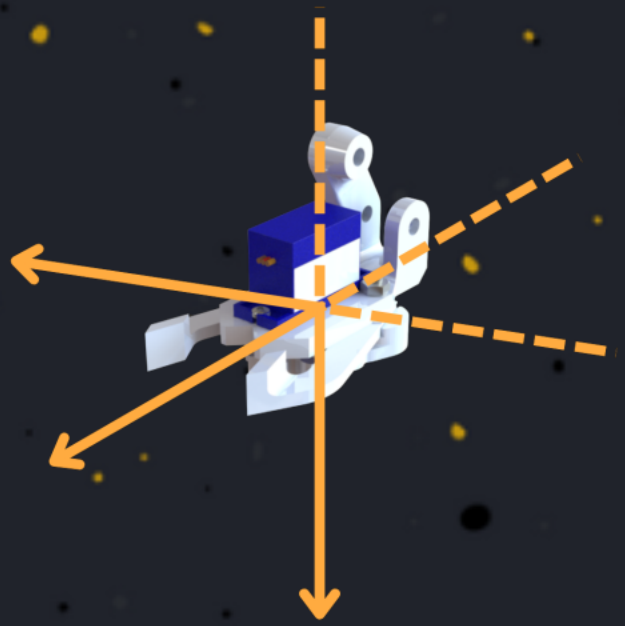
$$q_1 \cdot q_2$$

Inverse

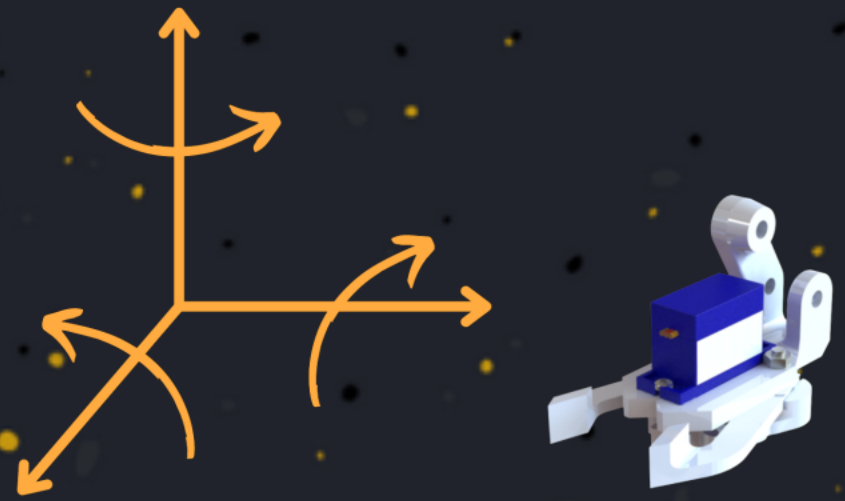
$$q^{-1} = a - bi - cj - dk$$

Angle Representations

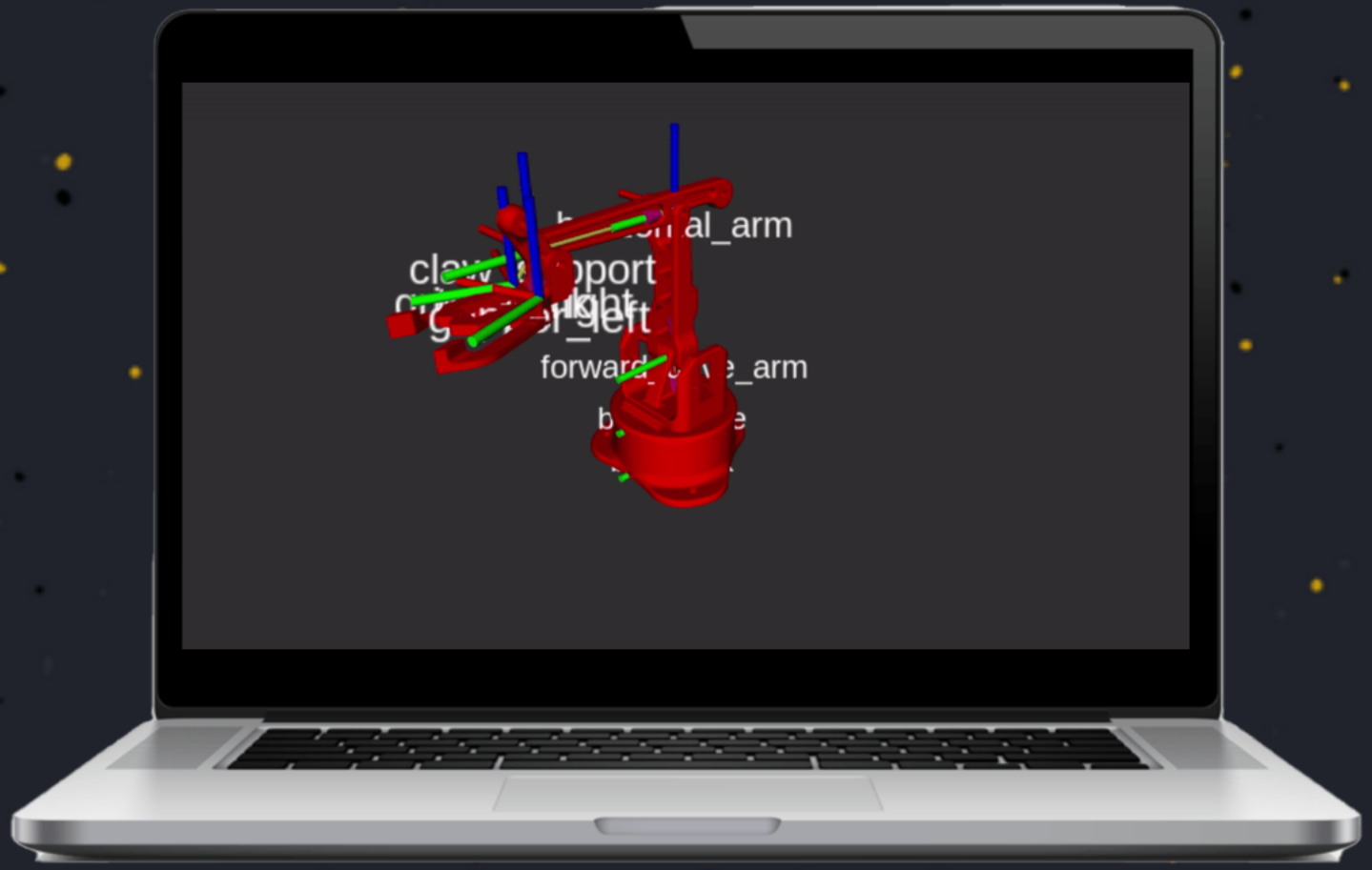
Euler Angles



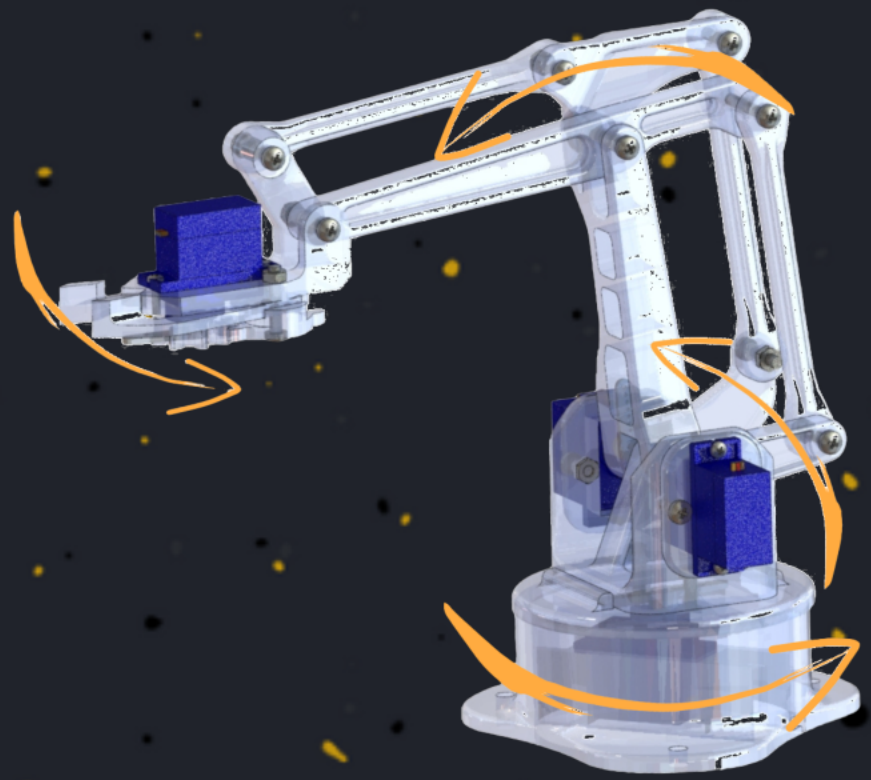
Quaternion

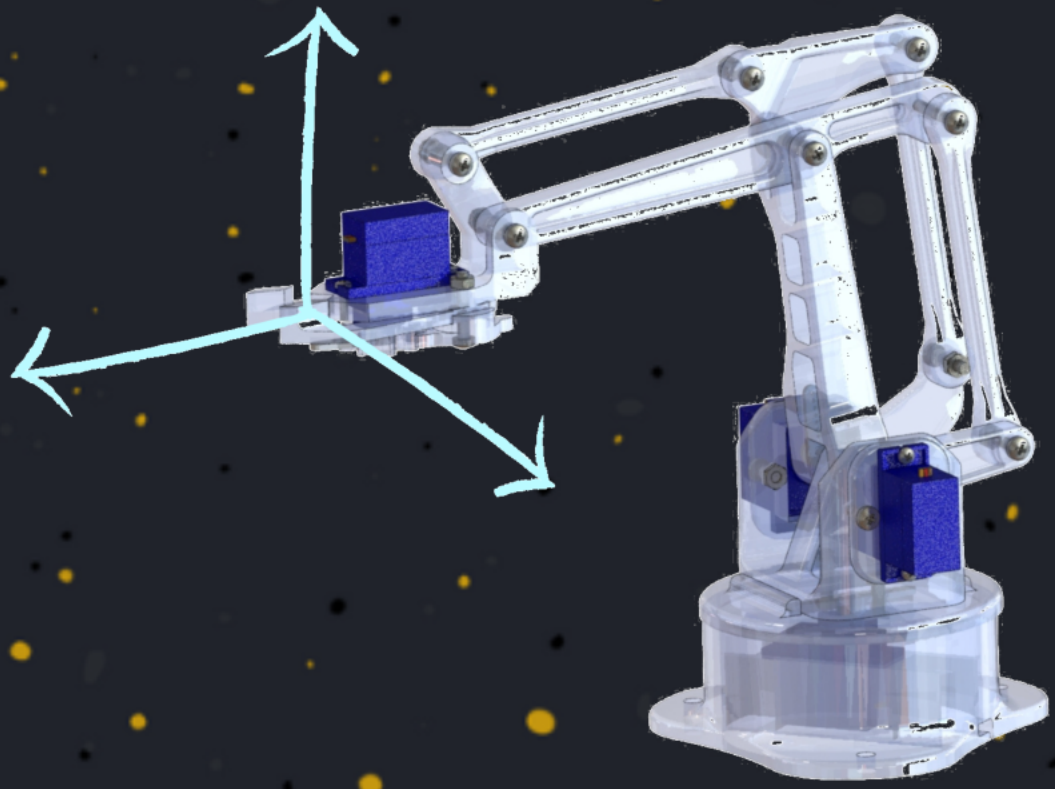


TF2 Library



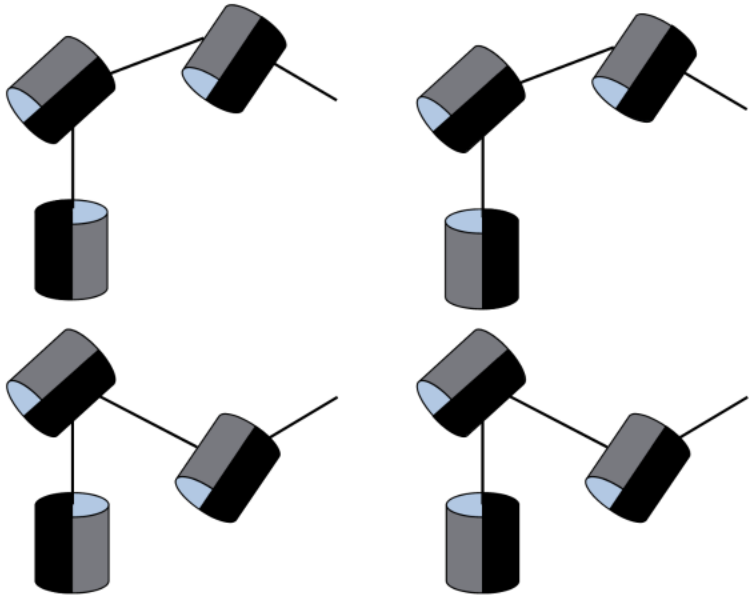
Forward





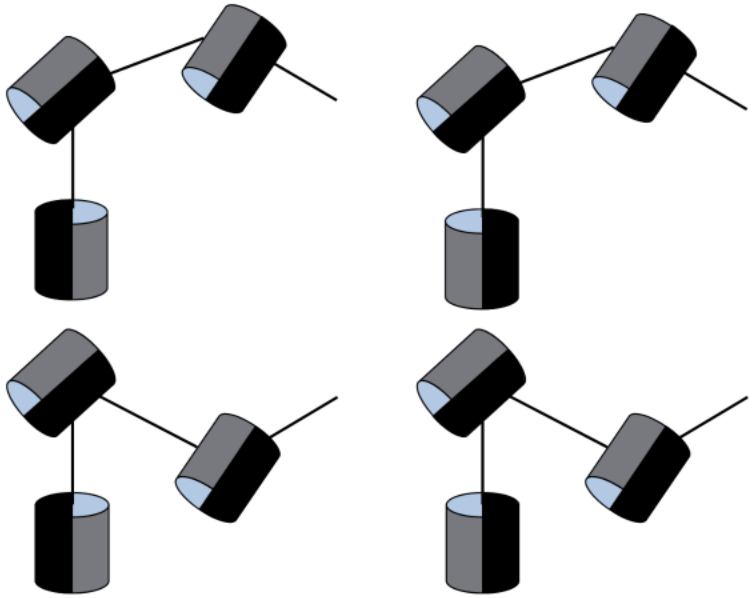
Inverse

Inverse Kinematics



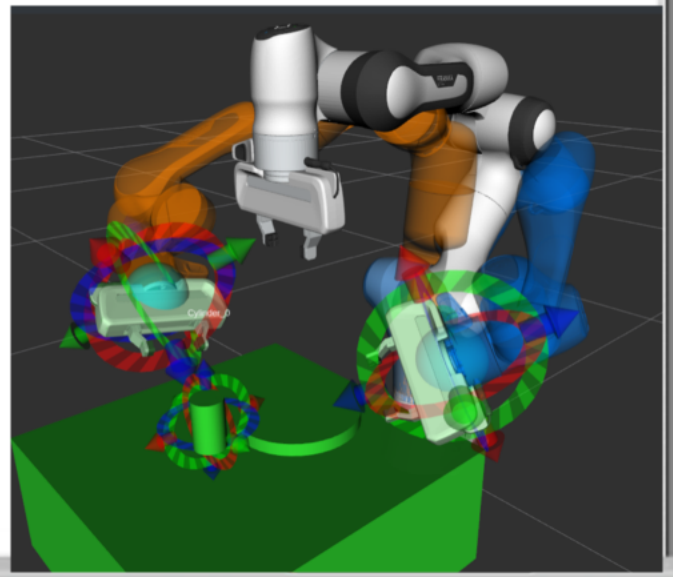
- Multiple Solutions
- Infinite Solutions
- No Solutions

Inverse Kinematics



- Multiple Solutions
- Infinite Solutions
- No Solutions

MoveIt2

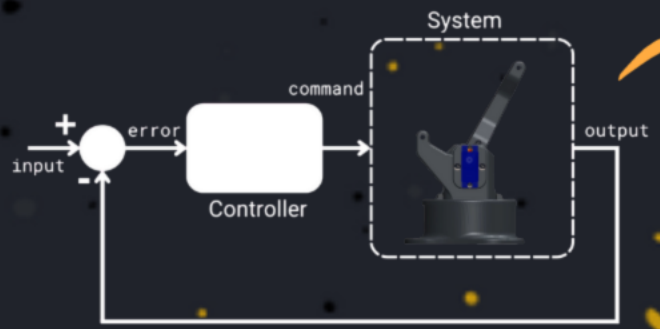


MoveIt2

Move Group

MoveIt2

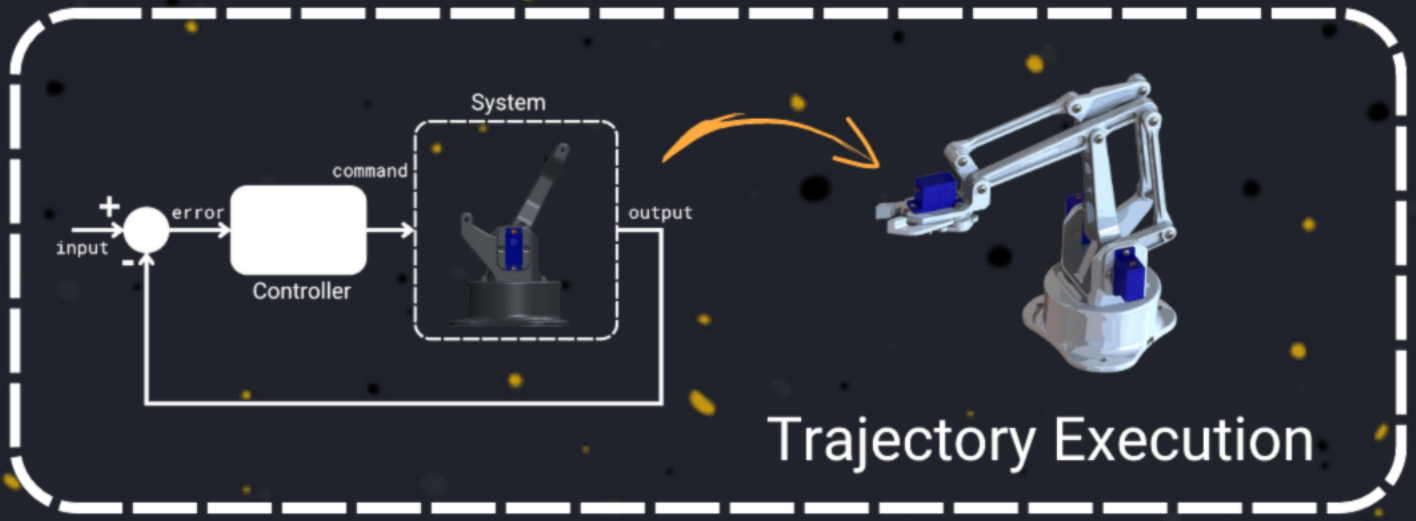
Move Group



Trajectory Execution

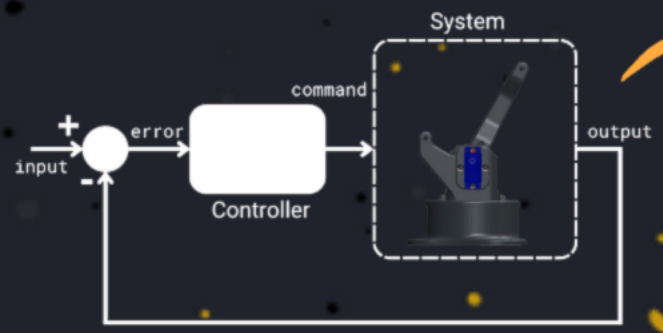
MoveIt2

Move Group

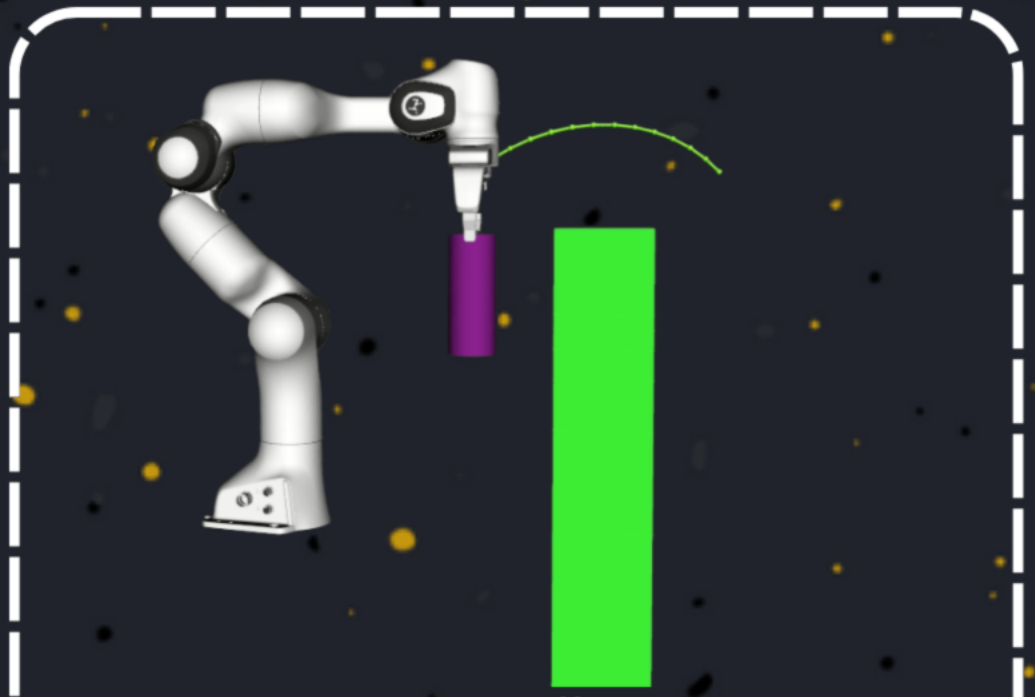


MoveIt2

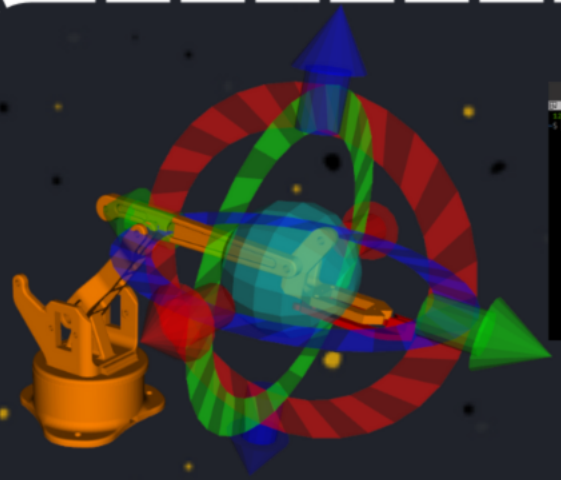
Move Group

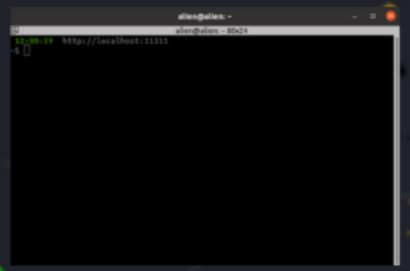


Trajectory Execution

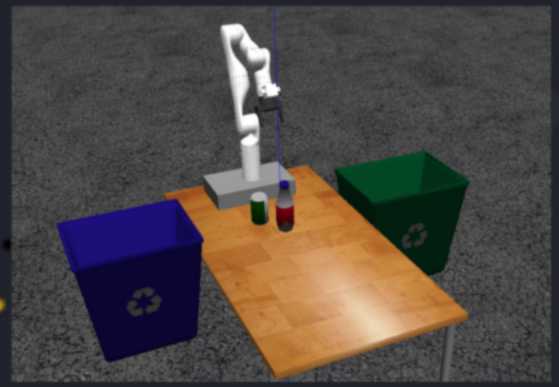


MoveIt2



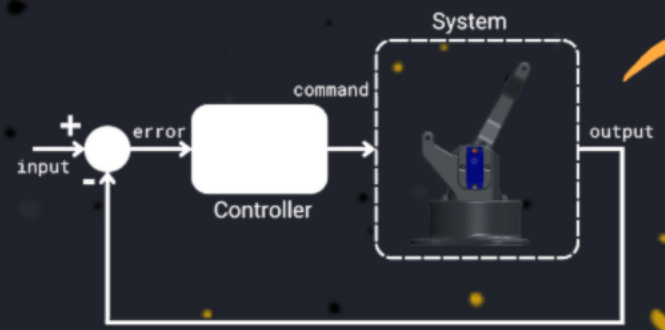


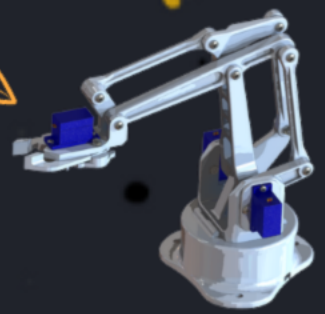
User Interface / API



Planning Scene

Move Group

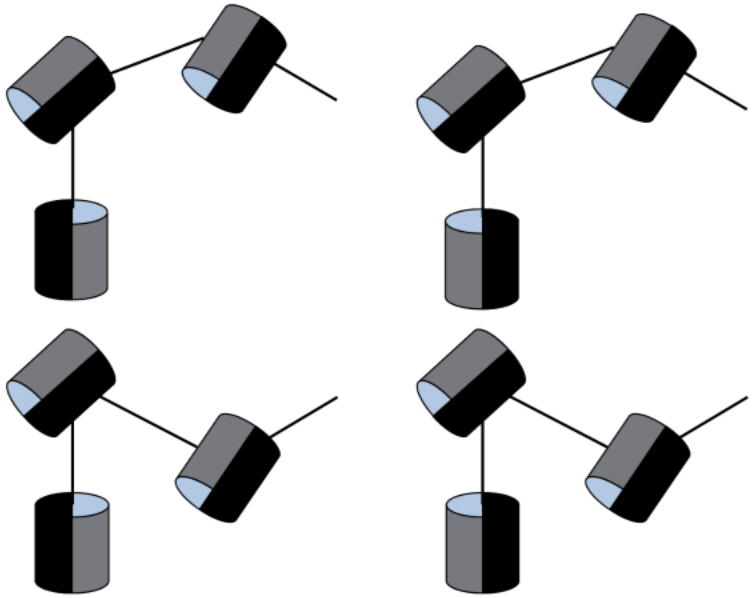




Trajectory Execution

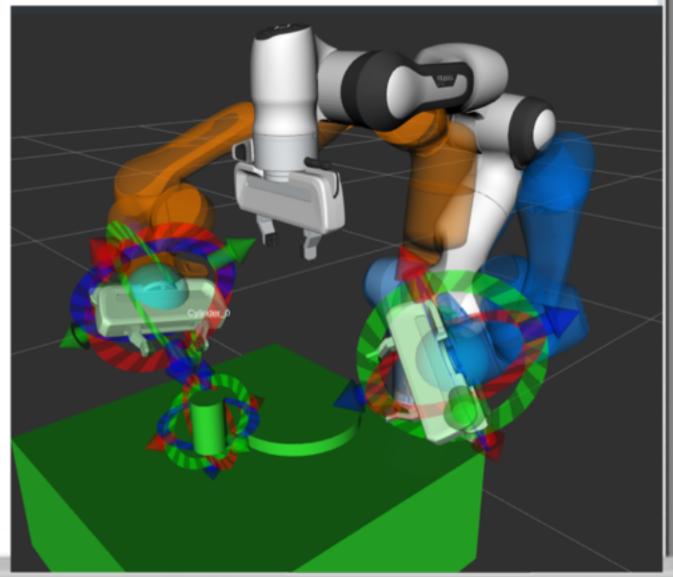
Collision Detection

Inverse Kinematics

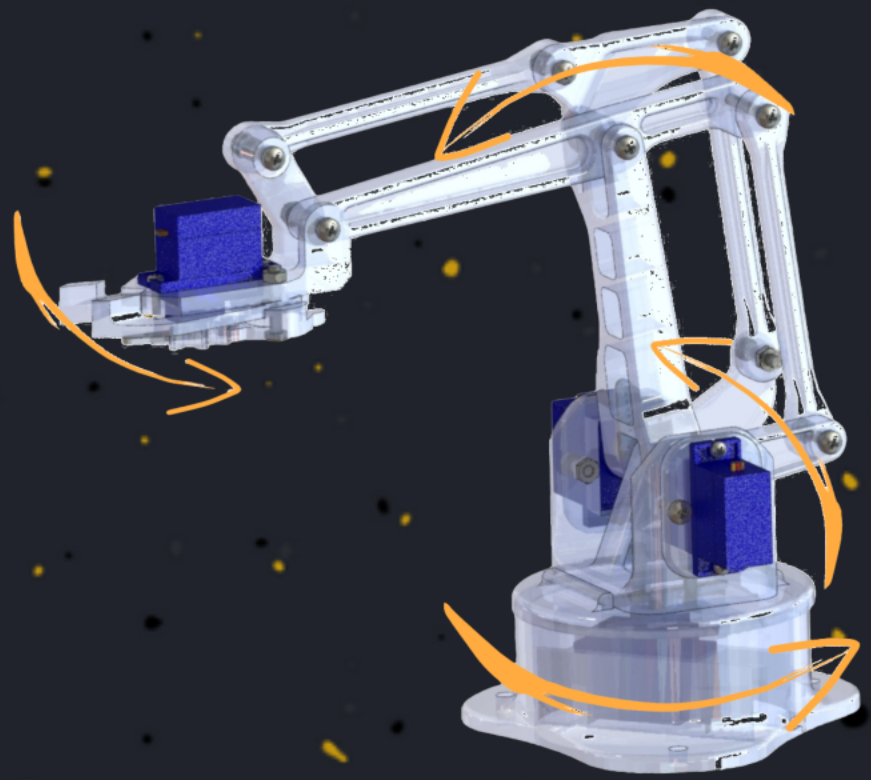


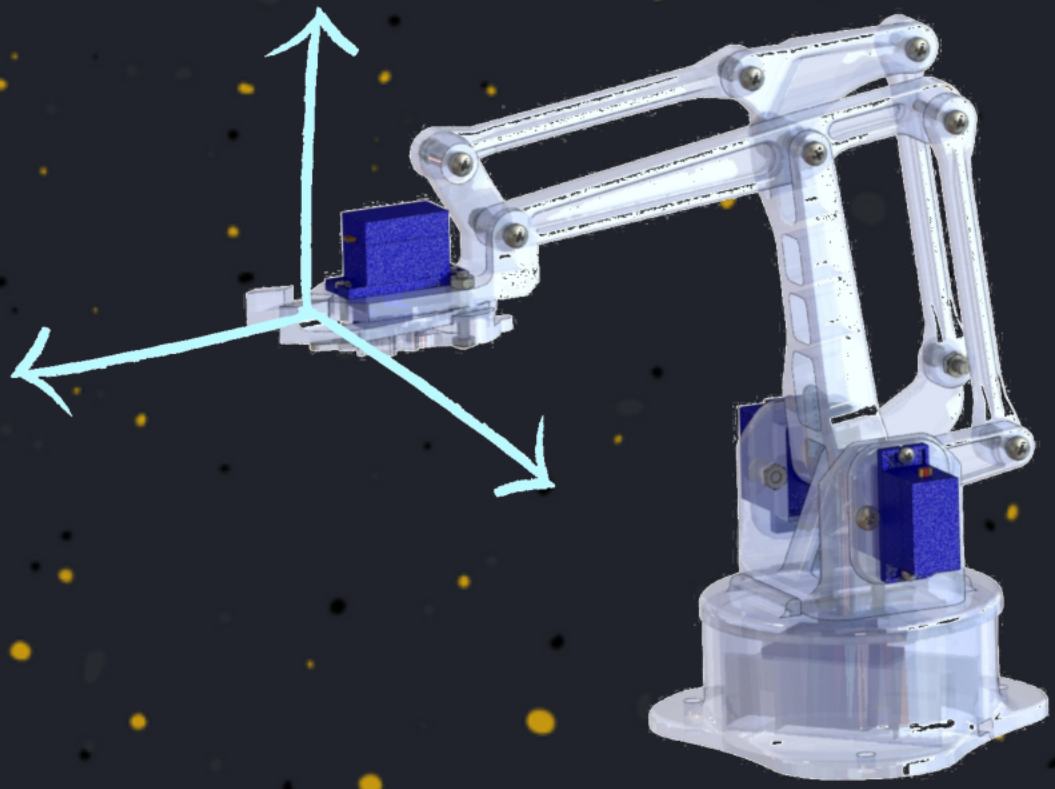
- Multiple Solutions
- Infinite Solutions
- No Solutions

MoveIt2



Forward





Inverse

Arduinobot

Introduction

Setup

Digital Twin

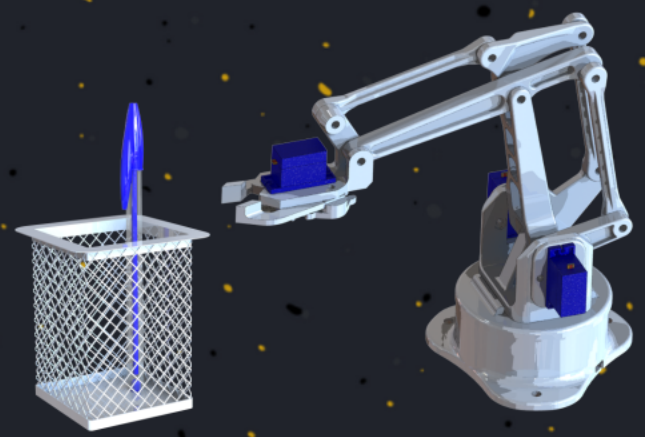
ROS 2

Control

Kinematics

Application

Alexa



Conclusions

Build