Introduction to Cognitive Robotics

Module 4: Robot Manipulators

Lecture 3: Robot programming by frame-based task specification

David Vernon
Carnegie Mellon University Africa

www.vernon.eu

Robot Programming by Task Specification

By defining a series of manipulator end-effector positions *Mn,* a task can be described as a sequence of manipulator movements to these defined positions

For example, a task to pick and place an object might be formulated as follows

*M*0: Move out of the field of view of the camera

Determine the pose of a object and a suitable grasp point (possibly using a camera)

*M*1: Move to an approach position close to the grasp point

*M*2: Move to the grasp position

Grasp the object

*M*3: Move to the depart position above the grasp point

*M*4: Move to the approach position in above the destination position

*M*5: Move to the destination position

Release the object

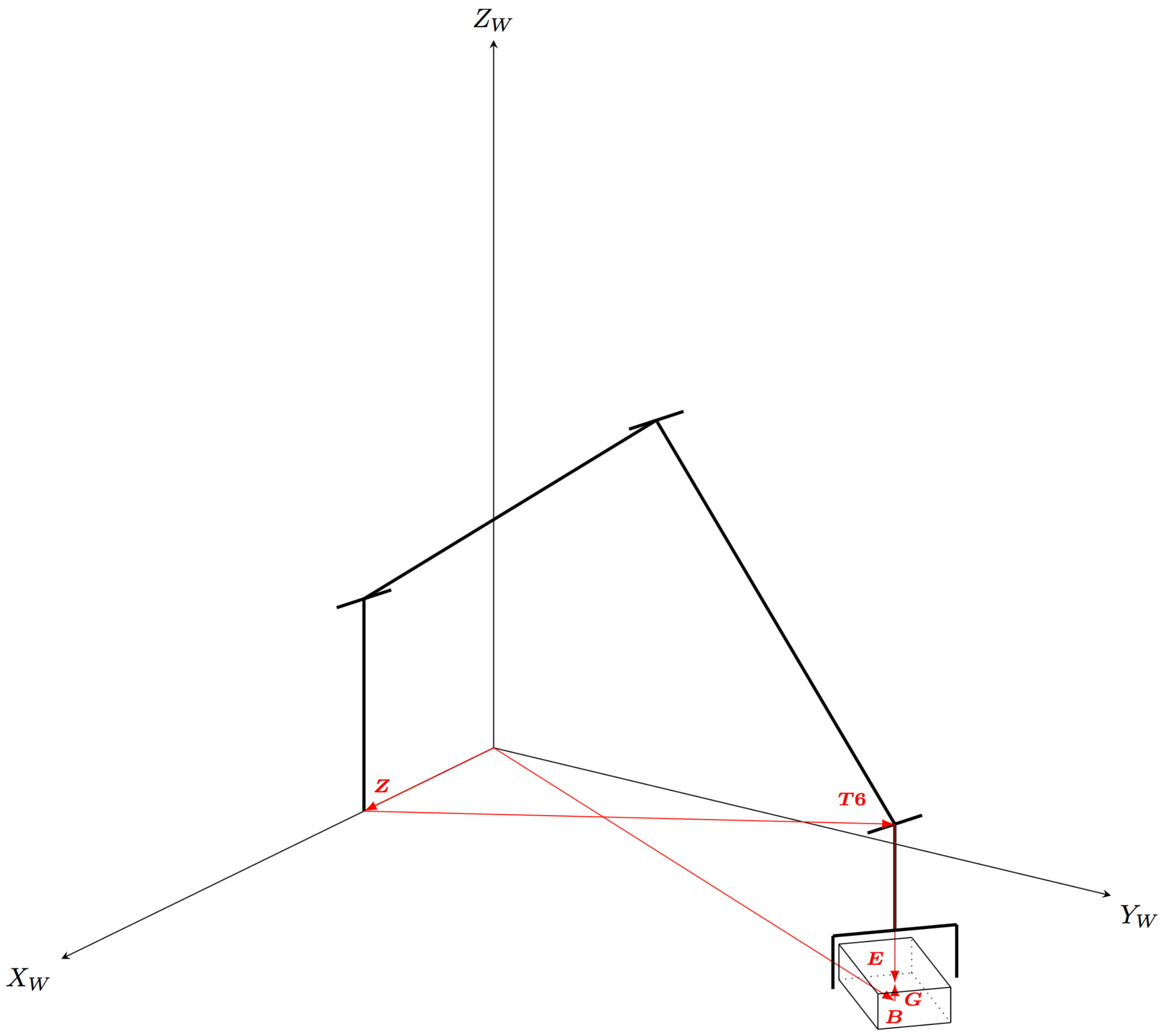
*M*6: Move to the depart position away from the destination position

- We are specifying the task in terms of movements of the robot but the object are what we are really interested in
- The object movements are implicit in the fact that the manipulator has grasped it
- We make up for this when we describe the structure of the task by considering the structure of the task's component objects:
  - the manipulator
  - the end-effector
  - the object being manipulated
  - the object grasp pose
- In particular, we will use the explicit positional relationships between these objects to describe the task structure

Since coordinate frames can be used to describe object position and orientation …

And since we may need to describe a coordinate frame in two or more ways (there is more than one way to reach any given position and orientation) …

We use transform equations to relate the two descriptions



A manipulator grasping a block

$\boldsymbol{Z}$ is the transformation (frame) which describes the position of manipulator with respect to the base co-ordinate reference frame

$^{Z}\boldsymbol{T6}$ describes the end of the manipulator (*i.e.* the wrist) with respect to the base of manipulator, *i.e.* with respect to $\boldsymbol{Z}$

$^{T6}\boldsymbol{E}$ describes the end-effector with respect to the end of the manipulator, *i.e.* with respect to $\boldsymbol{T6}$

$\boldsymbol{B}$ describes a block's position with respect to the base coordinate reference frame

$^{B}\boldsymbol{G}$ describes the manipulator end-effector with respect to the block, *i.e.* with respect to $\boldsymbol{B}$.

In this example, the end-effector is described in two ways, by the transformations leading from the base to the wrist to the end-effector :

$$Z * {}^{Z}T6 * {}^{T6}E$$

and by the transformations leading from the block to the end-effector grip position:

$$B * {}^{B}G$$

Equating these descriptions, we get the following transformation equation:

$$\boldsymbol{Z}\ {}^{Z}\boldsymbol{T6}\ {}^{T6}\boldsymbol{E} = \boldsymbol{B}\ {}^{B}\boldsymbol{G}$$

Alternatively, including the explicit composition operator in Corke (2016)

$$\boldsymbol{Z} \oplus {}^{Z}\boldsymbol{T6} \oplus {}^{T6}\boldsymbol{E} = \boldsymbol{B} \oplus {}^{B}\boldsymbol{G}$$

- Solving for $\boldsymbol{T6}$ by multiplying across by the inverse of $\boldsymbol{Z}$ and $^{T6}\boldsymbol{E}$

$$^{Z}\boldsymbol{T6} = \boldsymbol{Z}^{-1}\ \boldsymbol{B}\ ^{B}\boldsymbol{G}\ ^{T6}\boldsymbol{E}^{-1}$$

- $\boldsymbol{T6}$ is a function of the joint variables of the manipulator and, if known, then the appropriate joint variables can be computed using the inverse kinematic solution

- ***T6*** then is the coordinate frame which we wish to program in order to effect the manipulation task

- An arm position and orientation specified by ***T6*** is, thus, equivalent to our previous informal movement ***Mn***

$$\text{Move } \boldsymbol{Mn} = \text{Move } {}^{Z}\boldsymbol{T6}$$

  - since we can compute ***T6*** in terms of our known frame we now have an arm movement which is specified in terms of the frames which describe the task structure

- Assigning the appropriate value to $\boldsymbol{T6}$ and moving to that position, implicitly using the inverse kinematic solution

$$^{Z}\boldsymbol{T6} = \boldsymbol{Z}^{-1}\ \boldsymbol{B}\ {}^{B}\boldsymbol{G}\ {}^{T6}\boldsymbol{E}^{-1}$$

Move $^{Z}\boldsymbol{T6}$

- What we have not yet done is to fully specify each of these frames by embedding them in the appropriate objects and specifying the transformations which define them

- Note that the position of the end-effector with respect to the base reference system is represented by

$$\boldsymbol{Z} \;\; {}^{Z}\boldsymbol{T6} \;\; {}^{T6}\boldsymbol{E}$$

- This allows you to generate general-purpose and reusable robot programs

- In particular, the calibration of the manipulator to the workstation is represented by $\boldsymbol{Z}$, while if the task is to be performed with a change of tool, only E need be altered

- As we have seen, we specify the orientation of $T6$ by solving for it in terms of other frames/transformations in the task specification …

- We do this by

1. Embedding a frame in an object (or a desired point in space)

2. Specifying the position of the origin of the frame by applying a translation

3. Specifying the orientation of the frame by applying one or more rotations

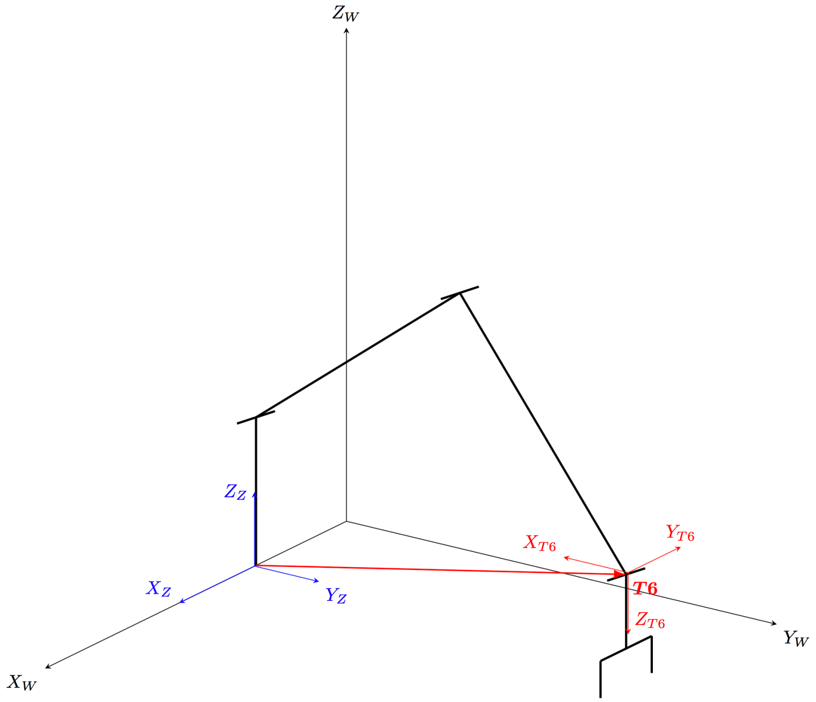
There is a convention that the ***T6*** frame should be embedded in the manipulator

- with the origin at the wrist

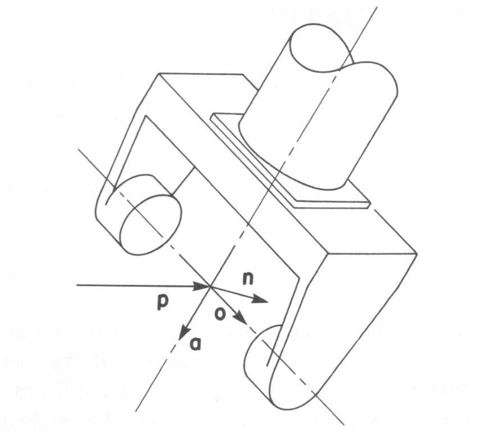
- with the Z axis directed outward from the wrist to the gripper

- with the Y axis directed in the plane of movement of the gripper when it is opening and closing

- with the X axis making up a right-hand system



The same convention applies to the $\boldsymbol{E}$ frame that is embedded in a two-finger gripper (end-effector … hence $\boldsymbol{E}$)

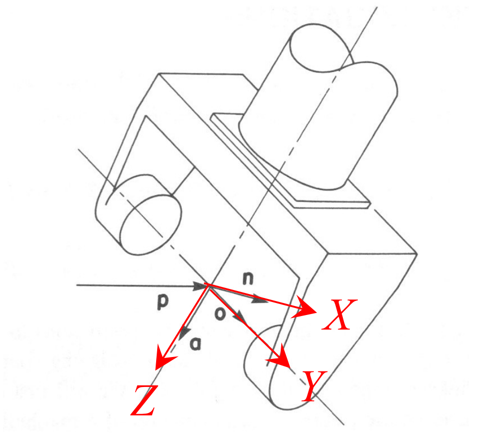


(Paul, 1981)

$$\boldsymbol{E} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

n Normal
o Orientation
a Approach

The same convention applies to the $\boldsymbol{E}$ frame that is embedded in a two-finger gripper (end-effector … hence $\boldsymbol{E}$)



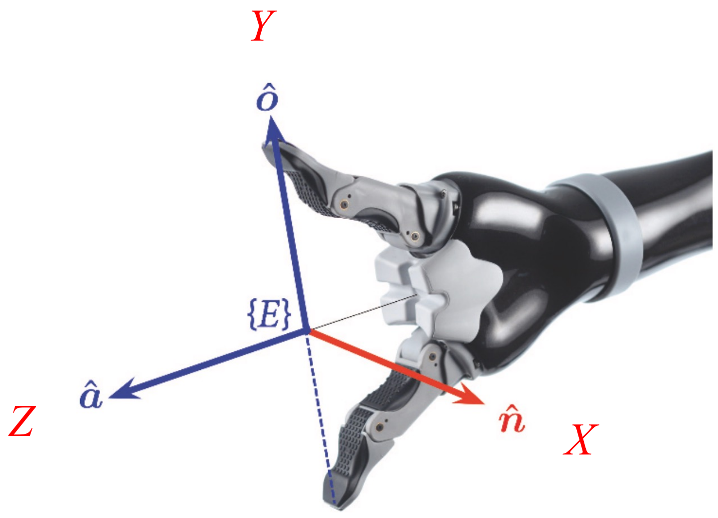
(Paul, 1981)

Direction of X axis
Direction of Y axis
Direction of Z axis

$$E = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

n Normal
o Orientation
a Approach

The same convention applies to the $\boldsymbol{E}$ frame that is embedded in a two-finger gripper (end-effector … hence $\boldsymbol{E}$)



(Corke, 2017), p. 41

Direction of X axis

Direction of Y axis

Direction of Z axis

$$E = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

n Normal

o Orientation

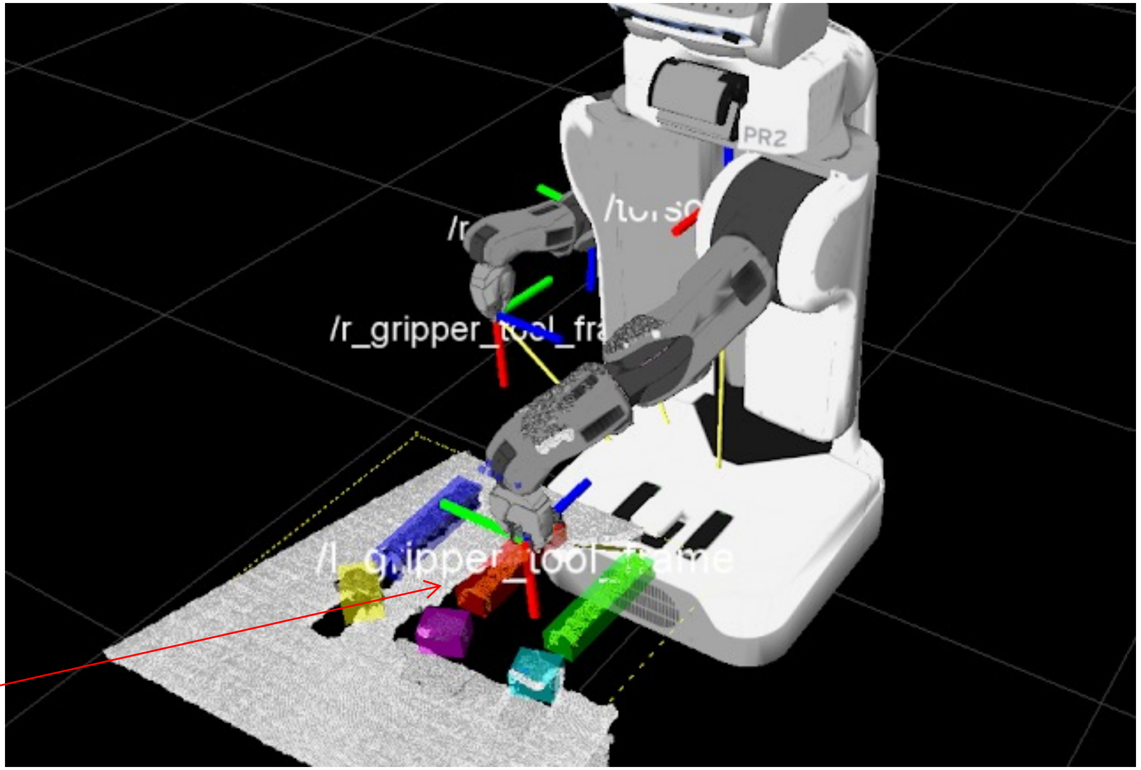
a Approach

ROS uses a different convention

"If the end effector is a grasping device, the frame should be located at the recommended object grasping location. The frame orientation is defined as X the axis going 'toward' the object. Y the main dimension in which the grasping device moves and Z orthogonal to X and Y axes."

https://www.ros.org/reps/rep-0120.html#l-gripper-and-r-gripper

This approach is consistent with the convention of embedding a frame in a vehicle, with the X axis aligned with the direction of travel; see conventions on specifying orientation using roll, pitch, and yaw in the following slides.

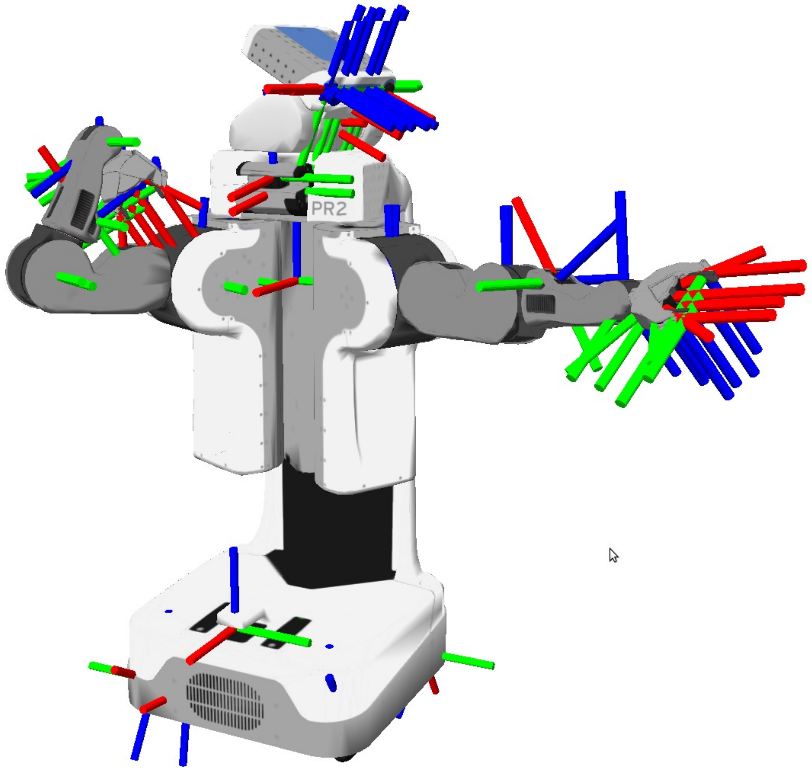


https://alliance.seas.upenn.edu/~meam620/wiki/index.php?n=IanMcMahon2011.Final

ROS uses a different convention

"If the end effector is a grasping device, the frame should be located at the recommended object grasping location. The frame orientation is defined as X the axis going 'toward' the object. Y the main dimension in which the grasping device moves and Z orthogonal to X and Y axes."

https://www.ros.org/reps/rep-0120.html#l-gripper-and-r-gripper

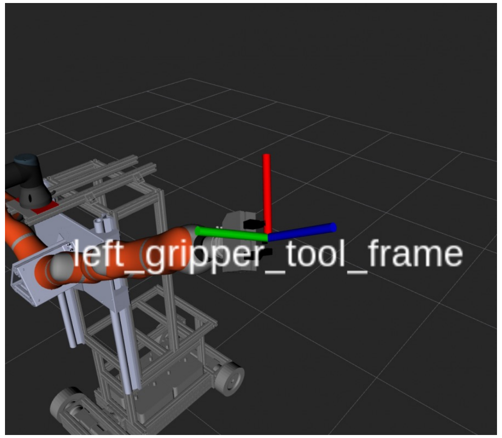


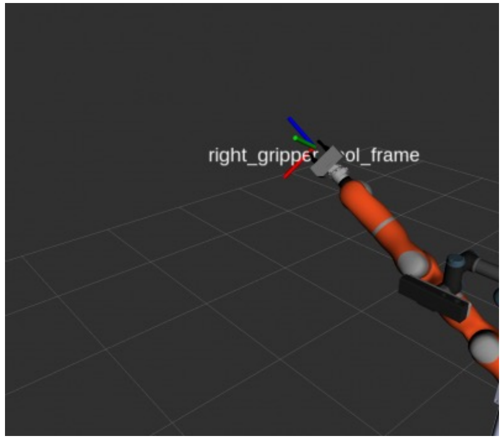
http://library.isr.ist.utl.pt/docs/roswiki/tf2.html

CRAM uses third convention

The frame orientation is defined as X the main dimension in which the grasping device moves, Y orthogonal to X and Z axes, and Z the axis going "toward" the object

This is similar to the standard approach, but with a rotation of 90° about the Z axis



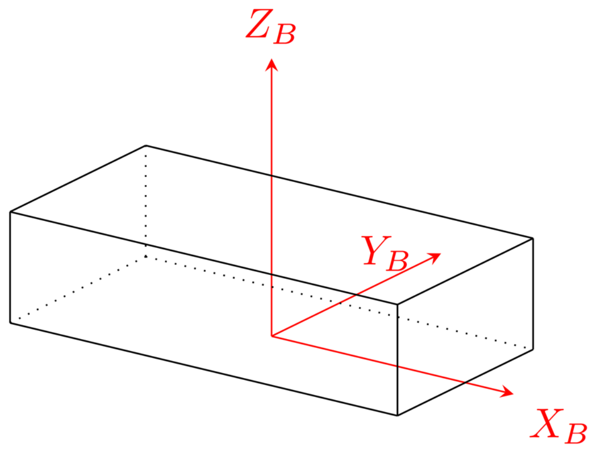


http://cram-system.org/tutorials/demo/fetch_and_place

Specifying Pose

We have seen that the pose of an object can be specified by embedding a frame in the object in some appropriate manner … for example:

- Placing the origin at the centre of the object
- Aligning the axes with the major and minor axes of the object

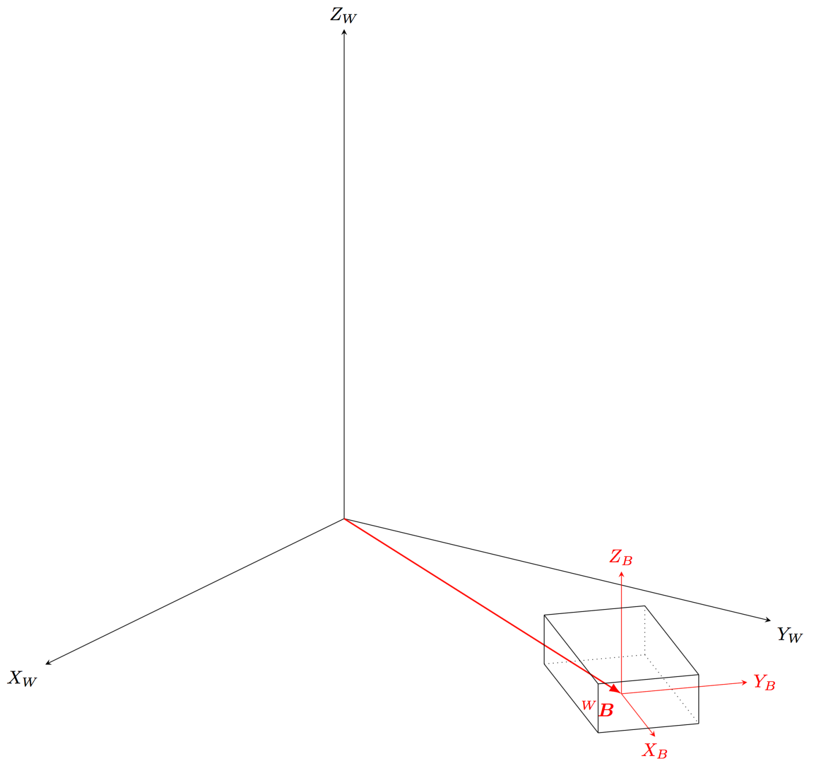


Specifying Pose

Then applying a homogenous transformation, e.g. $\boldsymbol{B} = \boldsymbol{Trans}\,(10, 20, 0)\ \boldsymbol{Rot}\,(Z, 50)$

- Translation part
  - Possibly several translations, applied in turn
- Rotation part
  - Possibly several rotations, applied in turn

You can specify them in whatever order you like, yielding a valid transform equation such as
$\boldsymbol{B} = \boldsymbol{Trans}\,(10, 20, 0)\ \boldsymbol{Rot}\,(Z, 50)\ \boldsymbol{Rot}\,(X, 10)\ \boldsymbol{Rot}\,(Z, 30)$



Specifying Orientation

That said, there are several conventions for the way these rotations are specified

One is Roll-Pitch-Yaw (RPY) … sometimes referred to as Cardan angles

RPY can be confusing. There are two reasons.

1. There are two conventions, each specifying a different sequence of axes about which to rotate:

   - $Z\,Y\,X$ normally used with vehicles

   - $X\,Y\,Z$ normally used with end-effectors

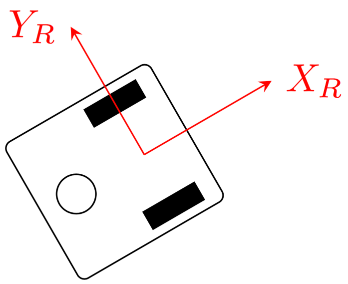
2. The angles are specified in the order yaw, pitch, roll (despite the name roll-pitch-yaw)

Specifying Orientation

That said, there are several conventions for the way these rotations are specified

Roll-Pitch-Yaw (RPY) with vehicles $Z\ Y\ X$

- The frame embedded in a vehicle normally has
  - X axis in the direction of travel
  - Z axis directly up
  - Y axis specified a right-hand system

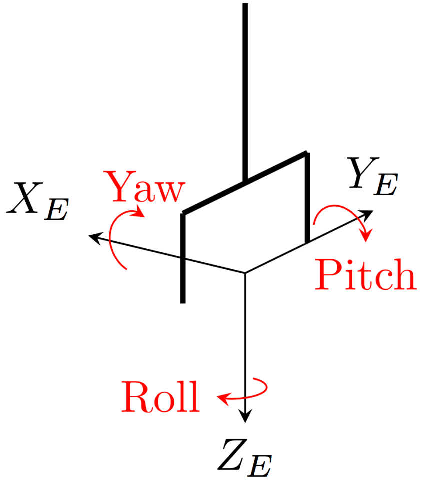


- The orientation is specified by $\boldsymbol{RPY}(\theta_y, \theta_p, \theta_r) = \boldsymbol{Rot}(Z, \theta_y)\ \boldsymbol{Rot}(Y, \theta_p)\ \boldsymbol{Rot}(X, \theta_r)$
  - First, rotate the yaw angle θ_y about the Z axis (i.e. about the vertical, thus specifying the direction of travel)
  - Second, rotate the pitch angle θ_p about the Y axis (thus specifying the angle of ascent or descent)
  - Third, rotate the roll angle θ_r about the X axis (thus specifying the banking angle)

Specifying Orientation

That said, there are several conventions for the way these rotations are specified

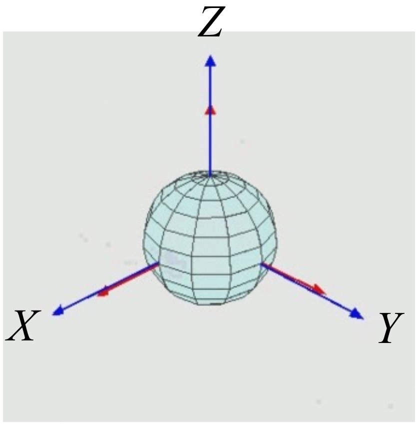
Roll-Pitch-Yaw (RPY) with end-effectors $X\ Y\ Z$



- The frame embedded in an end-effector or two-finger gripper normally has
  - X axis in the normal direction (i.e. normal to the movement of the fingers)
  - Z axis directed in the approach direction
  - Y axis direction in the orientation direction (i.e. parallel to the movement of the fingers)
- The orientation is specified by $\boldsymbol{RPY}\,(\theta_y, \theta_p, \theta_r) = \boldsymbol{Rot}\,(X, \theta_y)\,\boldsymbol{Rot}\,(Y, \theta_p)\,\boldsymbol{Rot}\,(Z, \theta_r)$
  - First, rotate the yaw angle θ_y about the X axis (i.e. about the normal)
  - Second, rotate the pitch angle θ_p about the Y axis (about the orientation)
  - Third, rotate the roll angle θ_r about the Z axis (about the approach)

Specifying Orientation
Euler Angles

- There are other commonly-used conventions for specifying the orientation of objects/frames
  - For example: Euler angles (e.g. rotation about Z, X, Z axes, in that order)



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

  - Note that there are twelve Euler angle conventions; this is just one of them

- We also use quaternions, especially in ROS

Recommended Reading

D. Vernon, Machine Vision – Automated Visual Inspection and Robot Vision, Prentice Hall International, 1991. Chapter 8.

http://vernon.eu/publications/91_Vernon_Machine_Vision.pdf

Similar material to that presented in this lecture.

R. P. Paul, Robot Manipulators – Mathematics, Programming, and Control, MIT Press, 1981. Chapter 1.

https://books.google.rw/books?id=UzZ3LAYqvRkC&printsec=frontcover&source=gbs_ViewAPI&redir_esc=y#v=onepage&q&f=false

Similar material to that presented in this lecture but complete comprehensive treatment.

P. Corke, Robotics, Vision and Control, 2nd Edition, Springer, 2017.
Comprehensive contemporary treatment; highly recommended.