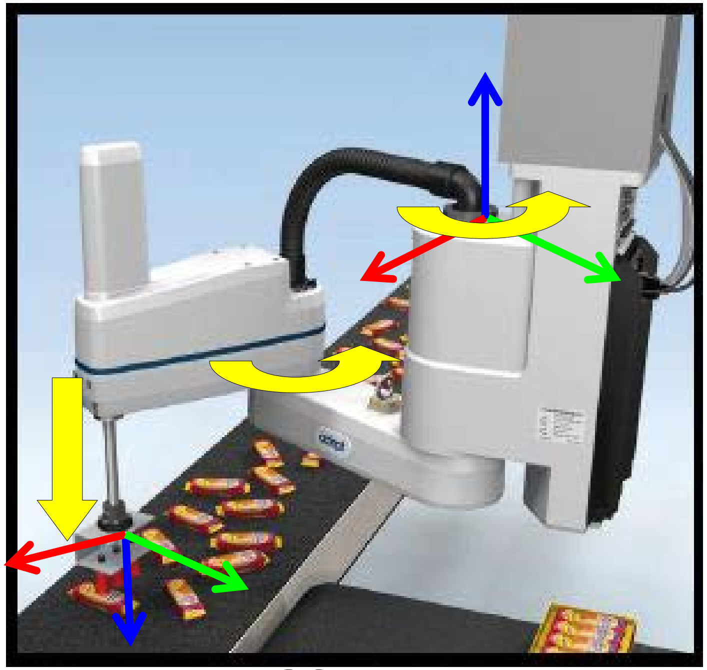
Robot Kinematics

Simon Leonard
Department of Computer Science
Johns Hopkins University

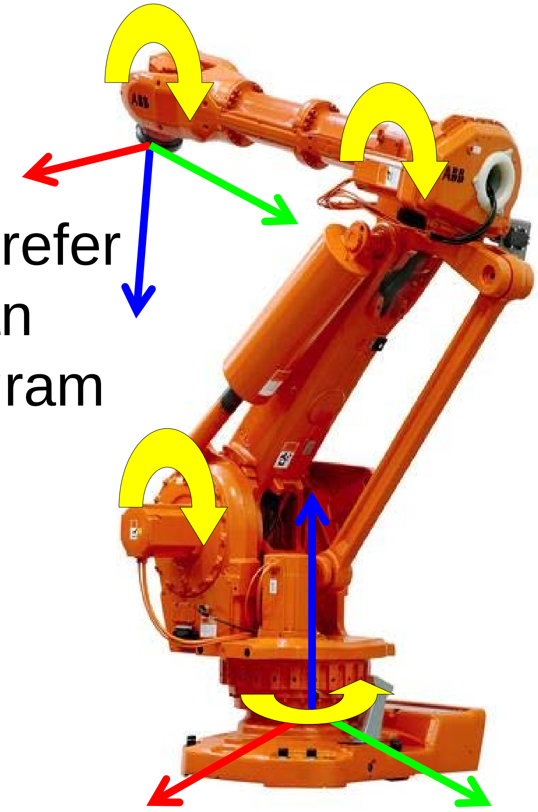
Robot Manipulators

- A robot manipulator is typically moved through its joints
  - Revolute: rotate about an axis
  - Prismatic: translate along an axis



SCARA

But we often prefer using Cartesian frames to program motions



6 axes robot arm

Kinematics

FORWARD KINEMATICS

$[{}^{B}R_T, {}^{B}\mathbf{t}_T] = f(\mathbf{q})$

Cartesian Space

Tool Frame (*T*)
Base Frame (*B*)

$[\,{}^{B}R_T,\ {}^{B}\mathbf{t}_T\,]$

${}^{B}R_T$:Orientation of T wrt B
${}^{B}\mathbf{t}_T$: Position of T wrt B

$\mathbf{q} = f^{-1}(\,[{}^{B}R_T, {}^{B}\mathbf{t}_T]\,)$

INVERSE KINEMATICS

Joint Space

Joint 1 = q_1
Joint 2 = q_2
...
Joint N = q_N

Rigid body motion
Transformation between coordinate frames

Linear algebra

Transformation Within Joint Space

Joint spaces are defined in $\mathbb{R}^N$

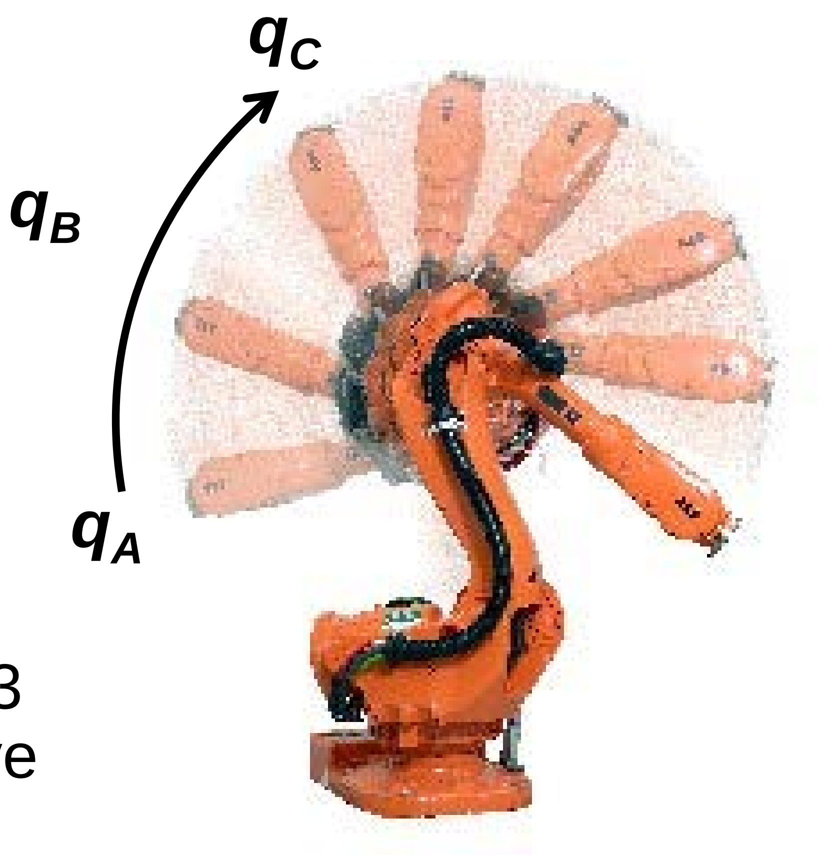
Thus for a vector of joint values

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_N \end{bmatrix}$$

we can add/subtract joint values

$$\boldsymbol{q}_c = \boldsymbol{q}_A + \boldsymbol{q}_B$$

How many joints do you need? It depends on the task. But ISO 8373 requires all industrial robots to have at least three or more axes.



Kinematics

Cartesian Space
Tool Frame (***T***)
Base Frame (***B***)

$[\,{}^{B}R_{T},\ {}^{B}\boldsymbol{t}_{T}\,]$

${}^{B}R_{T}$:Orientation of T wrt B
${}^{B}\boldsymbol{t}_{T}$: Position of T wrt B

Rigid body motion
Transformation between coordinate frames

FORWARD KINEMATICS

$[{}^{B}R_{T},\ {}^{B}\boldsymbol{t}_{T}] = f(\boldsymbol{q})$

$\boldsymbol{q} = f^{-1}(\ [{}^{B}R_{T},\ {}^{B}\boldsymbol{t}_{T}]\ )$

INVERSE KINEMATICS

Joint Space

Joint 1 = q_1
Joint 2 = q_2
...
Joint N = q_N

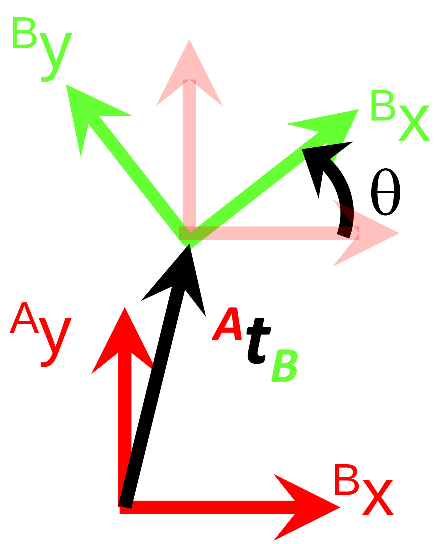
Linear algebra ✓

2D Rigid Motion

- Combine position and orientation:
  - Special Euclidean Group: SE(2)

Special Orthogonal (SO)

$$\mathrm{SE}(2) = \{(\boldsymbol{t}, \mathrm{R}) : \boldsymbol{t} \in \mathbb{R}^2, \mathrm{R} \in \mathrm{SO}(2)\} = \mathbb{R}^2 \times \mathrm{SO}(2)$$



${}^{A}\mathbf{t}_{B} \in \mathbb{R}^2$ is the translation between A and B

${}^{A}\mathrm{R}_{B} \in \mathrm{SO}(2)$ is the rotation between A and B

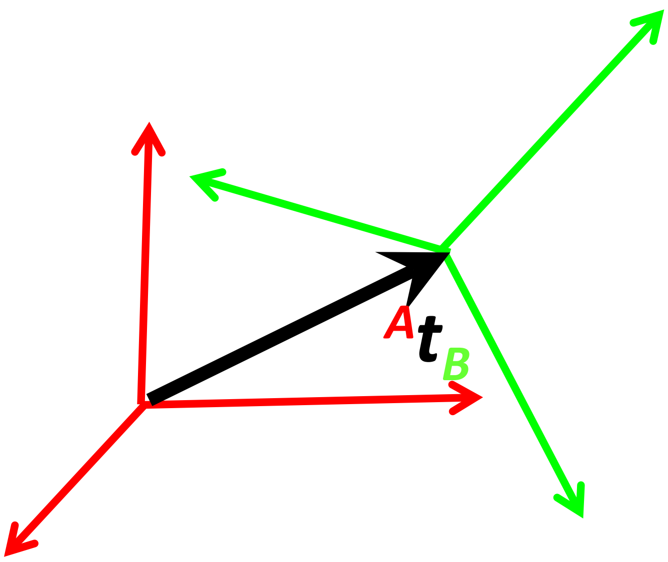
If $\mathrm{R} \in \mathrm{SO}(2)$, then $\mathrm{R} \in \mathbb{R}^{2\times 2}$, $\mathrm{R}\,\mathrm{R}^{T} = \mathrm{I}$ and $\det(\mathrm{R}) = 1$

$${}^{A}\mathrm{R}_{B} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

3D Rigid Motion

- Combine position and orientation:
  - Special Euclidean Group: SE(3)

$$\mathrm{SE}(3) = \{(\boldsymbol{t}, \mathrm{R}) : \boldsymbol{t} \in \mathbb{R}^3, \mathrm{R} \in \mathrm{SO}(3)\} = \mathbb{R}^3 \times \mathrm{SO}(3)$$



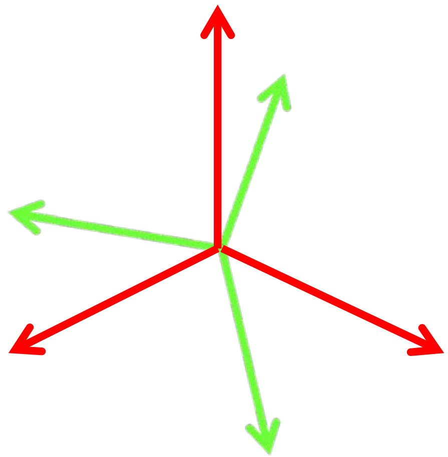
$^{A}\mathbf{t}_{B} \in \mathbb{R}^3$ is the translation between A and B

$^{A}\mathrm{R}_{B} \in \mathrm{SO}(3)$ is the rotation between A and B

If R ∈ $\mathrm{SO}(3)$, then $\mathrm{R} \in \mathbb{R}^{3\times3}$, R R^{T} = I and det(R) = 1

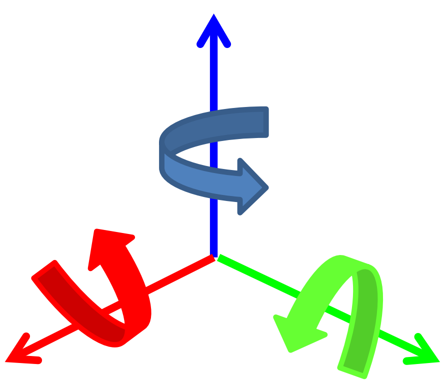
$$^{A}\mathrm{R}_{B} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

3D Rotations



$$ {}^{A}R_{B} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} $$

Can be factorized into a product of elementary rotations



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

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

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

Be careful of Commutations

R = Rx Ry Rz $\neq$ Rz Ry Rx

3D Rotations

- Lots of different ways to represent 3D rotations:
  - Quaternion, Euler angles, axis/angle, Rodrigues
  - They all have strengths (i.e. less than 9 numbers) and weaknesses (i.e. singularities)
    - "It is a fundamental topological fact that singularities can never be eliminated in any 3-dimensional representation of SO(3)." A Math. Introduction to Robotic Manipulation
  - They represent a different way to represent the SAME concept:

A 3x3 matrix R such that

$$(R^T)\,R = R\,(R^T) = I$$

$$\det(R^T) = +1$$

Homogeneous Representation

- A 2D point is represented by appending a "1" to yield a vector in R^3 $\mathbf{P}=[\ x\ y\ 1\ ]^T$
- A 3D point is represented by appending a "1" to yield a vector in R^4 $\mathbf{P}=[\ x\ y\ z\ 1\ ]^T$
- They are called *homogenous coordinates*
- The *affine transformation* of a point

$$ {}^{A}\mathbf{P} = {}^{A}R_{B}\, {}^{B}\mathbf{P} + {}^{A}\mathbf{t}_{B} $$

is represented by a *linear transformation* using a homogeneous coordinates

$$ \begin{bmatrix} {}^{A}\mathbf{P} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{A}R_{B} & {}^{A}\mathbf{t}_{B} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{B}\mathbf{P} \\ 1 \end{bmatrix} $$

Homogeneous Representation

$$\left.\begin{aligned} {}^{A}\mathbf{P} &= {}^{A}R_{B}\,{}^{B}\mathbf{P} + {}^{A}\mathbf{t}_{B} \\ {}^{B}\mathbf{P} &= {}^{B}R_{C}\,{}^{C}\mathbf{P} + {}^{B}\mathbf{t}_{C} \end{aligned}\right\}\text{ Affine transformations}$$

This is annoying $\Rightarrow$ $${}^{A}\mathbf{P} = {}^{A}R_{B}\,({}^{B}R_{C}\,{}^{C}\mathbf{P} + {}^{B}\mathbf{t}_{C}) + {}^{A}\mathbf{t}_{B}$$

$$\left.\begin{aligned} \begin{bmatrix} {}^{A}\mathbf{P} \\ 1 \end{bmatrix} &= \begin{bmatrix} {}^{A}R_{B} & {}^{A}\mathbf{t}_{B} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{B}\mathbf{P} \\ 1 \end{bmatrix} = {}^{A}E_{B} \begin{bmatrix} {}^{B}P \\ 1 \end{bmatrix} \\ \begin{bmatrix} {}^{B}\mathbf{P} \\ 1 \end{bmatrix} &= \begin{bmatrix} {}^{B}R_{C} & {}^{B}\mathbf{t}_{C} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{C}\mathbf{P} \\ 1 \end{bmatrix} = {}^{B}E_{C} \begin{bmatrix} {}^{C}P \\ 1 \end{bmatrix} \end{aligned}\right\}\text{ Linear transformations}$$

This is convenient $\Rightarrow$ $${}^{A}\mathbf{P} = {}^{A}E_{B}\,{}^{B}E_{C} \begin{bmatrix} {}^{C}P \\ 1 \end{bmatrix}$$

$${}^{A}\mathbf{P} = {}^{A}E_{C} \begin{bmatrix} {}^{C}P \\ 1 \end{bmatrix}$$

Kinematics

Cartesian Space
Tool Frame (***T***)
Base Frame (***B***)

$[\ {}^{B}R_{T},\ {}^{B}\mathbf{t}_{T}\ ]$

${}^{B}R_{T}$:Orientation of T wrt B
${}^{B}\mathbf{t}_{T}$: Position of T wrt B

FORWARD KINEMATICS

$[{}^{B}R_{T},\ {}^{B}\mathbf{t}_{T}] = f(\mathbf{q})$

$\mathbf{q} = f^{-1}(\ [{}^{B}R_{T},\ {}^{B}\mathbf{t}_{T}]\ )$

INVERSE KINEMATICS

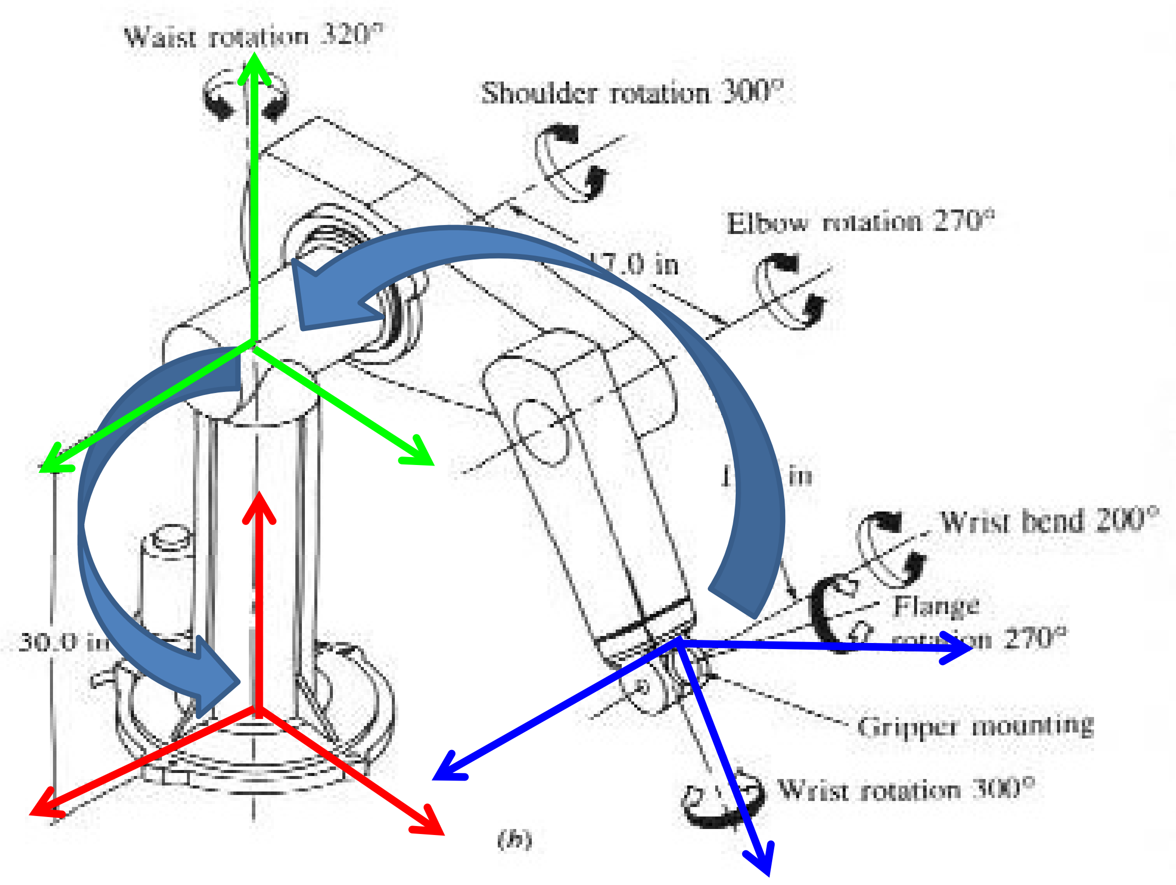
Joint Space

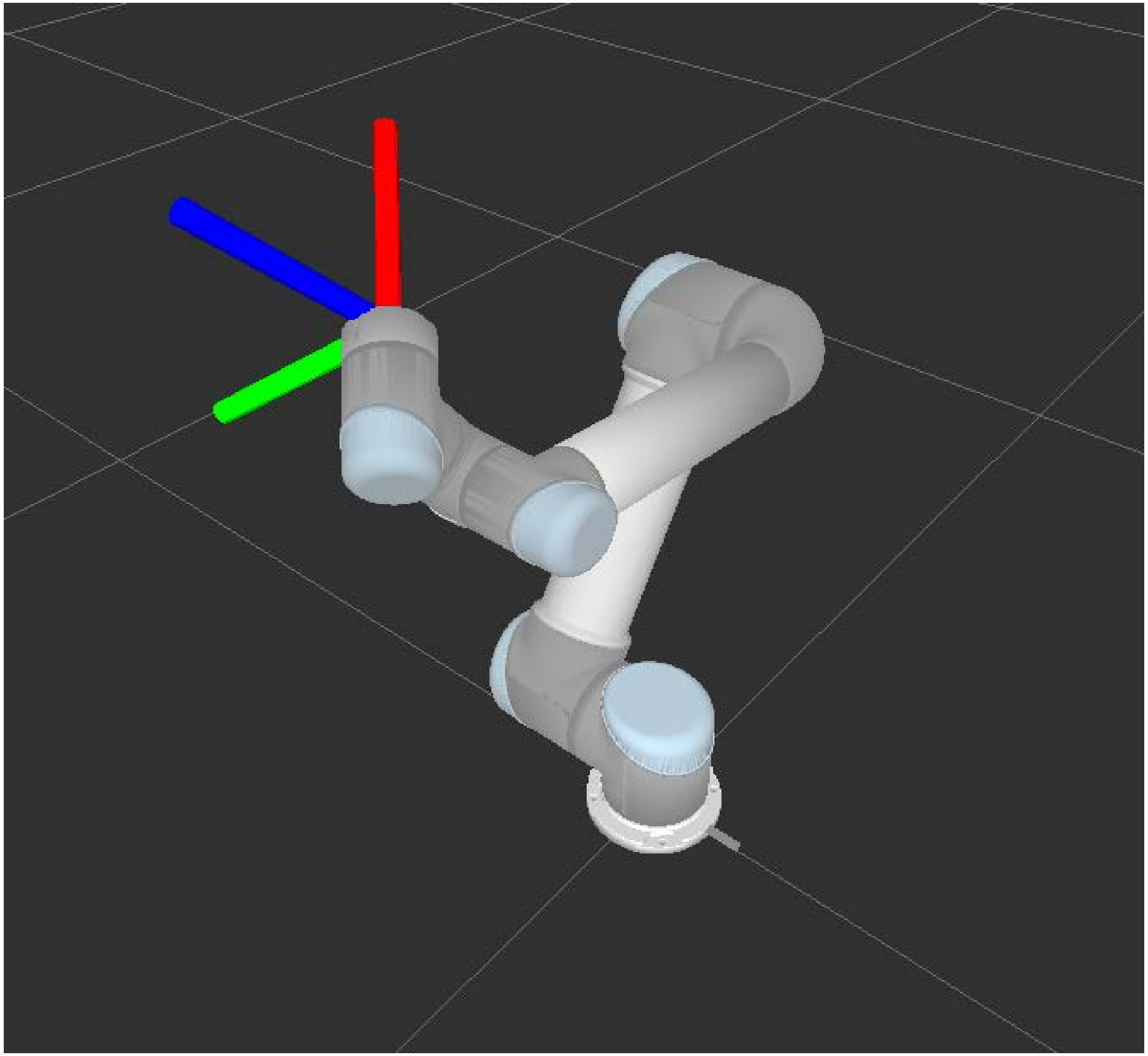
Joint 1 = q_1
Joint 2 = q_2
...
Joint N = q_N

Rigid body motion
Transformation between coordinate frames ✓

Linear algebra ✓

Cartesian Transformation
Kinematic Chain





Forward Kinematics

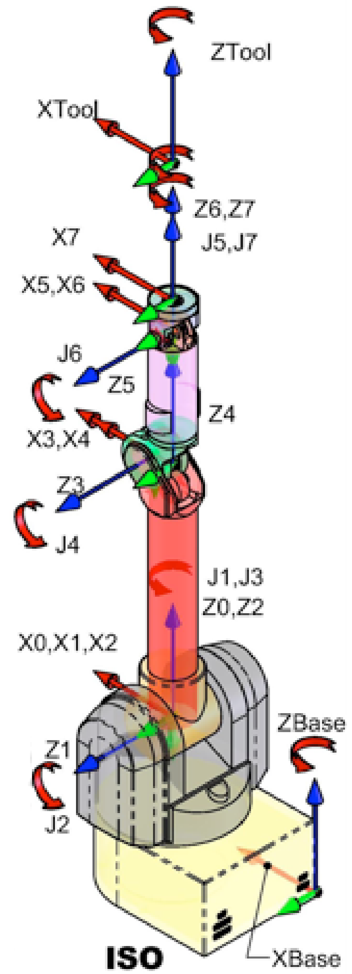
Guidelines for assigning frames to robot links:

- There are several conventions
  - Denavit Hartenberg (DH), modified DH, Hayati, etc.
  - They are “conventions” not “laws”
  - Mainly used for legacy reason (when using 4 numbers instead of 6 per link made a difference).

1) Choose the base and tool coordinate frame
   - Make your life easy!

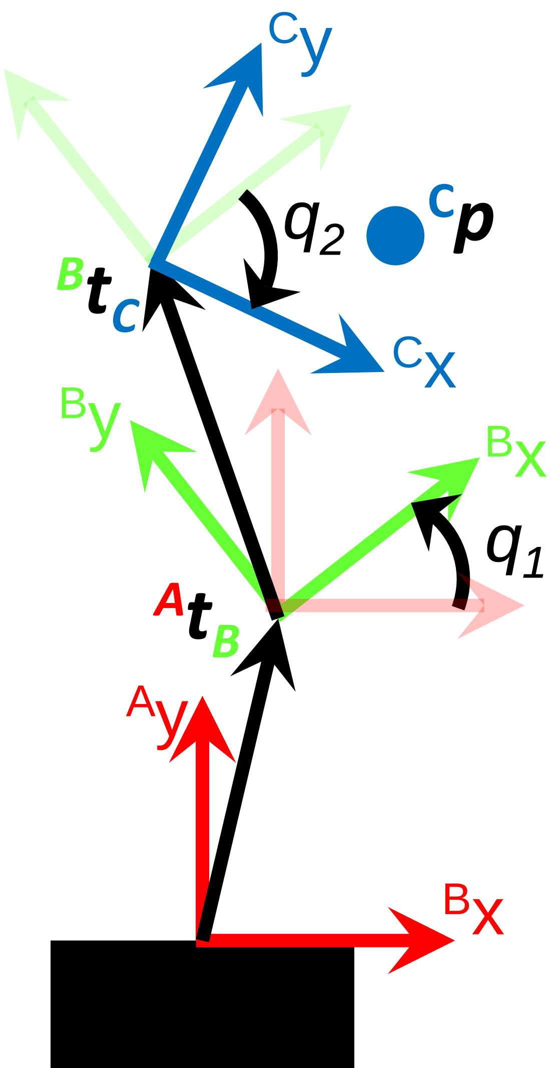
2) Start from the base and move towards the tool
   - Make your life easy!
   - In general each actuator has a coordinate frame.

3) Align each coordinate frame with a joint actuator
   - Traditionally it’s the “Z” axis but this is **not** necessary and any axis can be use to represent the motion of a joint



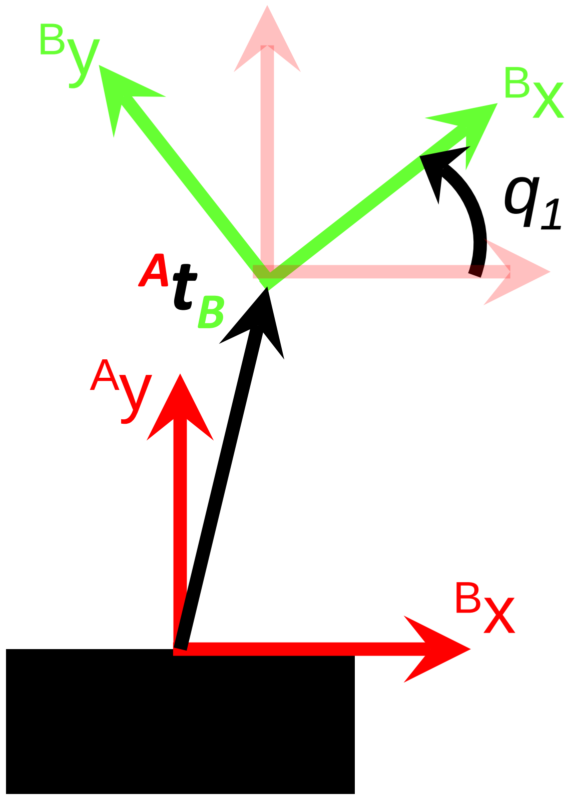
Barrett WAM

Rigid Body Motion 2D



- We have the coordinates of a point P in the coordinate frame "C"
- Given the following robot, what are the coordinates of P in the coordinate frame "A"?

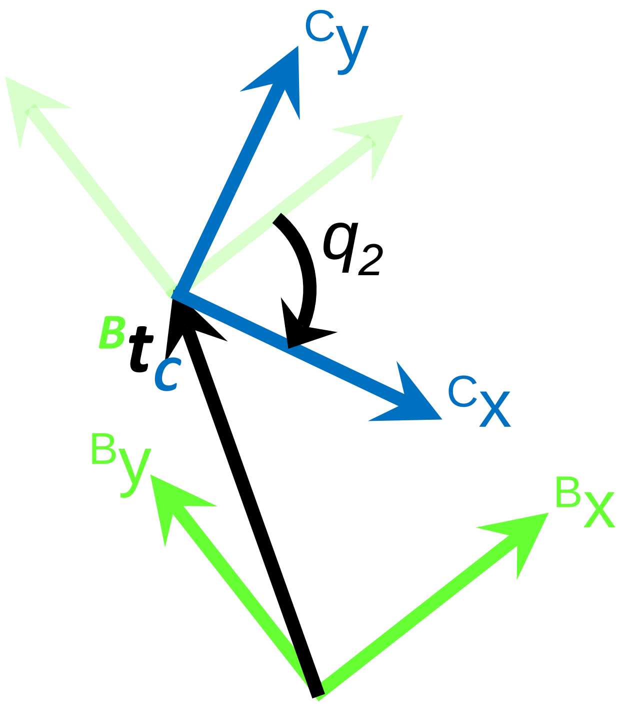
Forward Kinematics 2D



- First, what is the position and orientation of coordinate frame "B" with respect to coordinate frame "A"?
  - The position of B with respect to A is constant ^At_B
  - The orientation of B with respect to A depends on the angle q_1

$$^AR_B = \begin{bmatrix} \cos(q_1) & -\sin(q_1) \\ \sin(q_1) & \cos(q_1) \end{bmatrix}$$

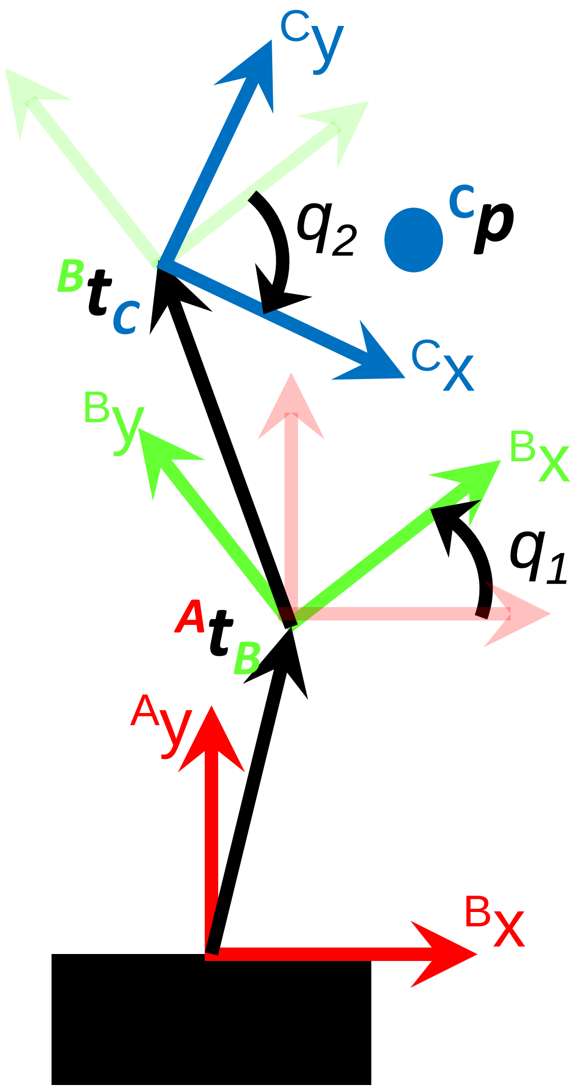
Forward Kinematics 2D



- Second, what is the position and orientation of coordinate frame "C" with respect to coordinate frame "B"?
  - The position of C with respect to B is constant $^{B}t_{C}$
  - The orientation of C with respect to B depends on the angle q_2

$$^{B}R_{C}=\begin{bmatrix}\cos(q_2) & -\sin(q_2)\\ \sin(q_2) & \cos(q_2)\end{bmatrix}$$

Forward Kinematics 2D



$$ {}^{B}E_{C} = \begin{bmatrix} {}^{B}R_{C} & {}^{B}\boldsymbol{t}_{C} \\ 0 & 1 \end{bmatrix} $$

$$ {}^{B}\mathbf{P} = {}^{B}E_{C}\,{}^{C}\mathbf{P} $$

$$ {}^{A}E_{B} = \begin{bmatrix} {}^{A}R_{B} & {}^{A}\boldsymbol{t}_{B} \\ 0 & 1 \end{bmatrix} $$

$$ {}^{A}\mathbf{P} = {}^{A}E_{B}\,{}^{B}\mathbf{P} $$

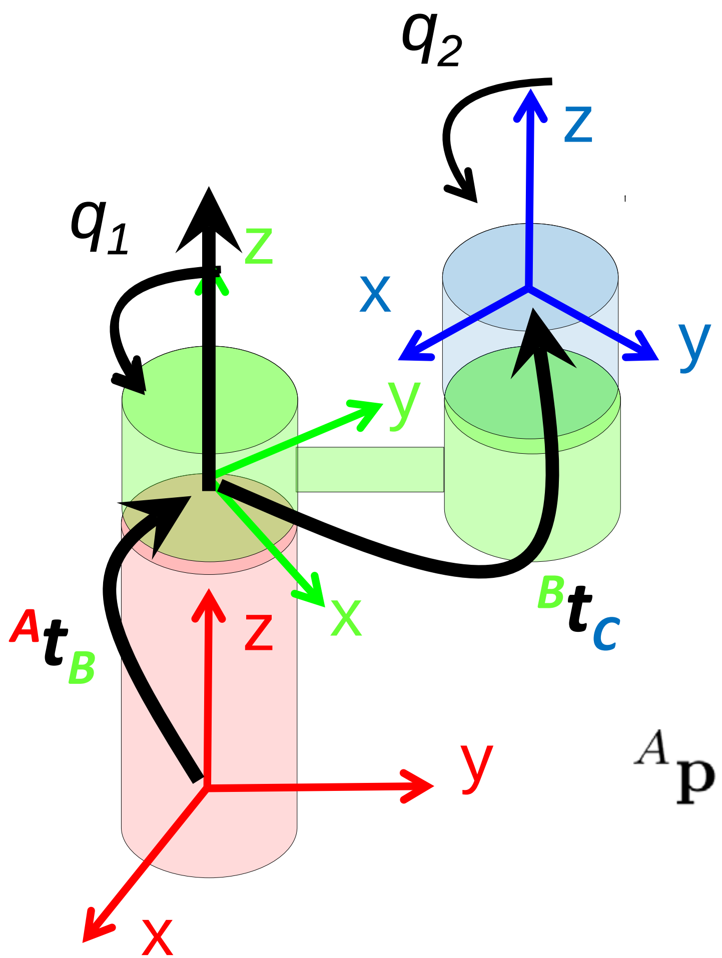
$$ {}^{A}\mathbf{P} = {}^{A}E_{B}\,{}^{B}E_{C}\,{}^{C}\mathbf{P} = {}^{A}E_{C}\,{}^{C}\mathbf{P} $$

$$ {}^{A}E_{C} = \begin{bmatrix} {}^{A}R_{C} & {}^{A}\boldsymbol{t}_{B} \\ 0 & 1 \end{bmatrix} $$

$$ {}^{A}\mathbf{P} = {}^{A}E_{C}\,{}^{C}\mathbf{P} $$

Forward kinematics
${}^{A}R_{C}$ and ${}^{A}\mathbf{t}_{B}$ are functions of q_1 and q_2

Forward Kinematics 3D



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

$${}^{A}\mathbf{p} = {}^{A}E_B\ {}^{B}\mathbf{p}$$

$$= \begin{bmatrix} R_z(q_1) & {}^{A}\mathbf{t}_B \\ \mathbf{0} & 1 \end{bmatrix}$$

$${}^{B}\mathbf{p} = {}^{B}E_C\ {}^{C}\mathbf{p}$$

$$= \begin{bmatrix} R_z(q_2) & {}^{B}\mathbf{t}_C \\ \mathbf{0} & 1 \end{bmatrix}$$

$${}^{A}\mathbf{p} = {}^{A}E_B\ {}^{B}E_C^{C}\mathbf{p}$$

$$= \begin{bmatrix} R_z(q_1)R_z(q_2) & {}^{A}\mathbf{t}_B + R_z(q_1)^{B}\mathbf{t}_C \\ \mathbf{0} & 1 \end{bmatrix}$$

Kinematics

FORWARD KINEMATICS

Cartesian Space
Tool Frame (***T***)
Base Frame (***B***)

$[\,{}^{B}R_{T},\ {}^{B}\mathbf{t}_{T}\,]$

${}^{B}R_{T}$:Orientation of T wrt B
${}^{B}\mathbf{t}_{T}$: Position of T wrt B

$[{}^{B}R_{T}, {}^{B}\mathbf{t}_{T}] = f(\mathbf{q})$

$\mathbf{q} = f^{-1}(\,[{}^{B}R_{T}, {}^{B}\mathbf{t}_{T}]\,)$

INVERSE KINEMATICS

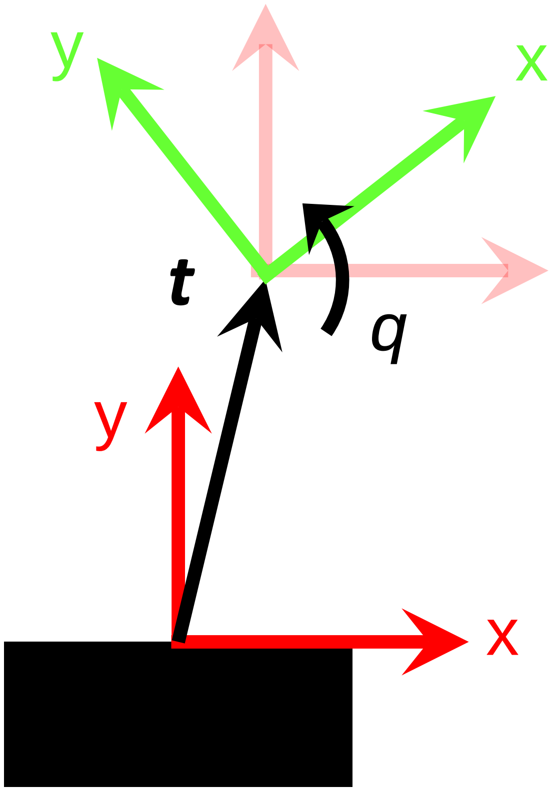
Joint Space

Joint 1 = q_1
Joint 2 = q_2
...
Joint N = q_N

Rigid body motion
Transformation between coordinate frames

Linear algebra

Inverse Kinematics 2D



$$ {}^{A}E_{B}(q) = \begin{bmatrix} \cos q & -\sin q & t_x \\ \sin q & \cos q & t_y \\ 0 & 0 & 1 \end{bmatrix} $$

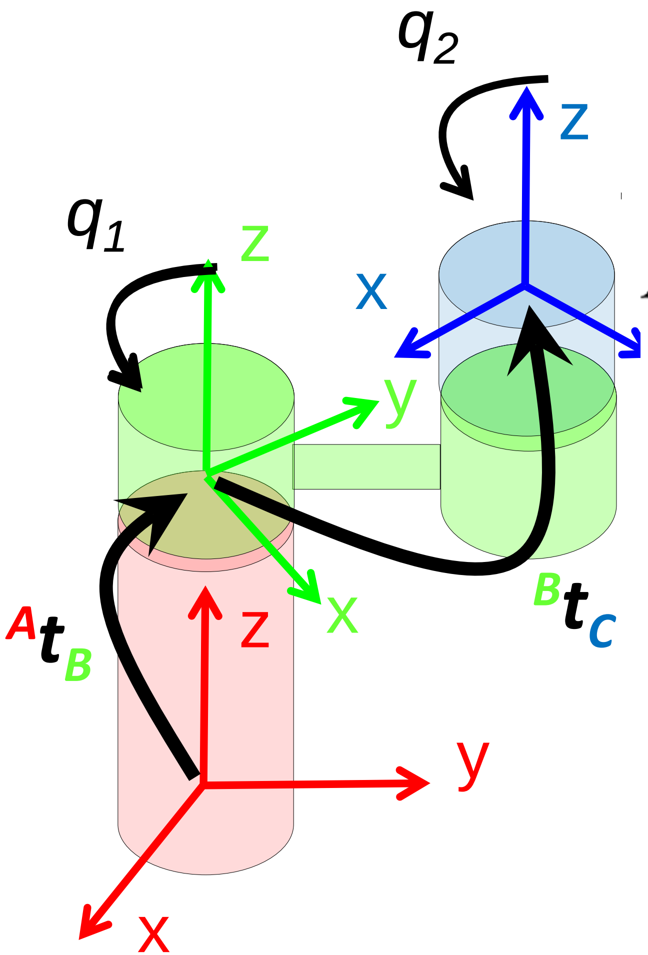
Given ${}^{A}R_{B}$ and ${}^{A}\boldsymbol{t}_{B}$ find $\boldsymbol{q}$

$\boldsymbol{q}$ only appears in ${}^{A}R_{B}$ so solving R for $\boldsymbol{q}$ is pretty easy. With several joints, the inverse kinematics gets very messy.

$$ {}^{A}R_{B}(\phi) = \begin{bmatrix} 0.7071 & -0.7071 & 0 \\ 0.7071 & 0.7071 & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

q = 45 degrees

Inverse Kinematics 3D



Likewise, in 3D we want to solve for the position and orientation of the last coordinate frame: Find q_1 and q_2 such that

$$^{A}E_{C} = \begin{bmatrix} R_z(q_1)R_z(q_2) & {}^{A}\mathbf{t}_B + R_z(q_1)\ {}^{B}\mathbf{t}_C \\ \mathbf{0} & 1 \end{bmatrix}$$

Solving the inverse kinematics gets messy fast!

A) For a robot with several joints, a symbolic solution can be difficult to get

B) A numerical solution (Newton's method) is more generic

Note that the inverse kinematics is NOT the inverse of the forward kinematics $\left({}^{A}E_{B}^{-1}\right)$

Kinematics

Cartesian Space
Tool Frame (***T***)
Base Frame (***B***)

$[\ {}^{B}v_T,\ \ {}^{B}w_T\ ]$

${}^{B}v_T$:linear vel. of T wrt B
${}^{B}w_T$: angular vel. of T wrt B

JACOBIAN

$[\ v, w\ ]^T = J(\boldsymbol{q})\ \dot{\boldsymbol{q}}$

$\dot{\boldsymbol{q}} = J^{-1}(\boldsymbol{q})\ [\ v, w\ ]^T$

INVERSE
JACOBIAN

Joint Space

Joint 1 = $\dot{q}_1$
Joint 2 = $\dot{q}_2$
...
Joint N = $\dot{q}_N$

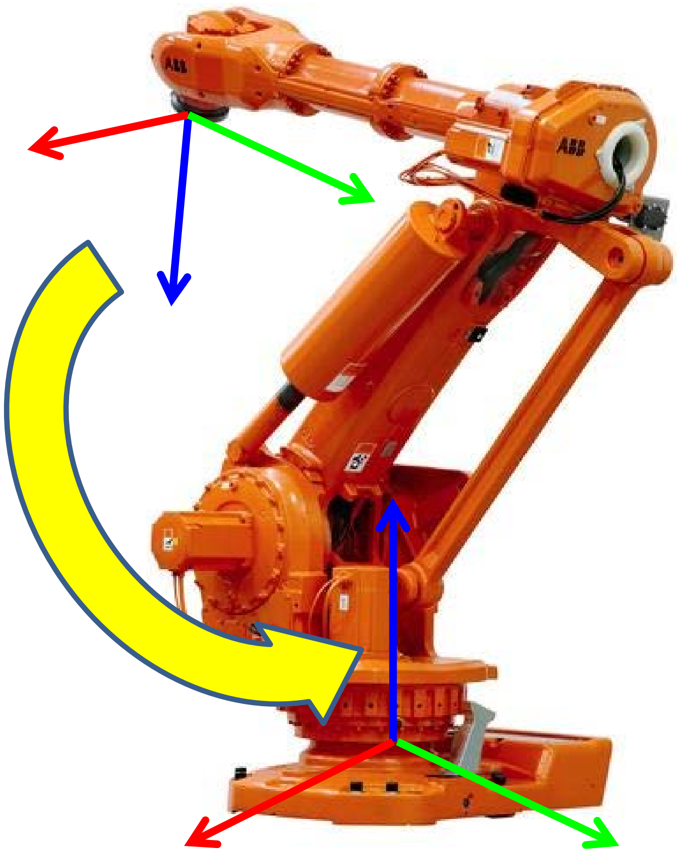
Rigid body motion
Transformation between coordinate frames

Linear algebra ✓

G.D. Hager
S. Leonard

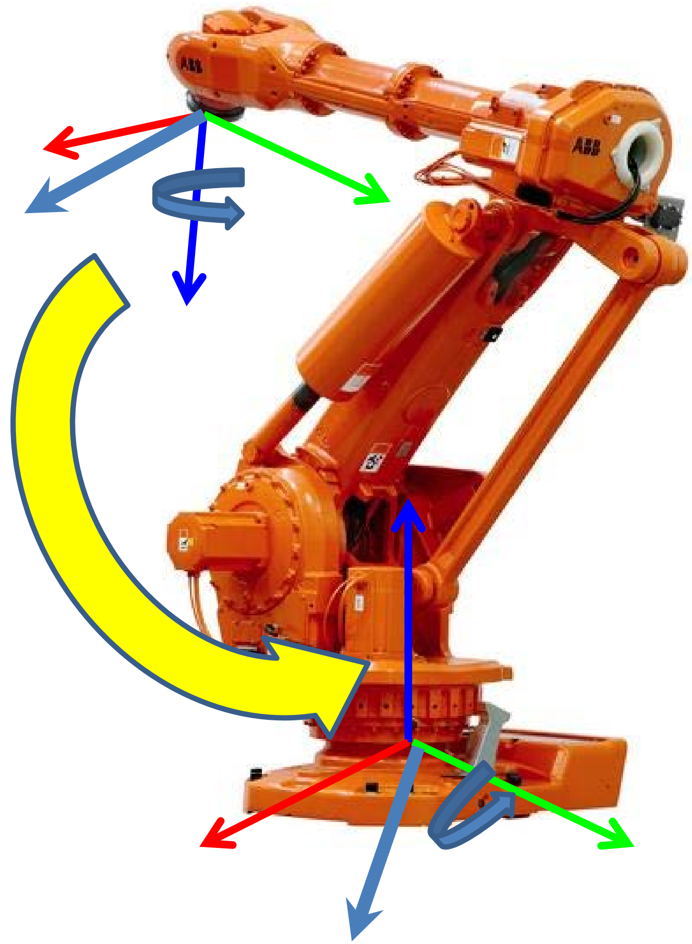
Rigid Body Transformation

Relates two coordinate frames



Rigid Body Velocity

Relate a 3D velocity in one coordinate frame to an equivalent velocity in another coordinate frame

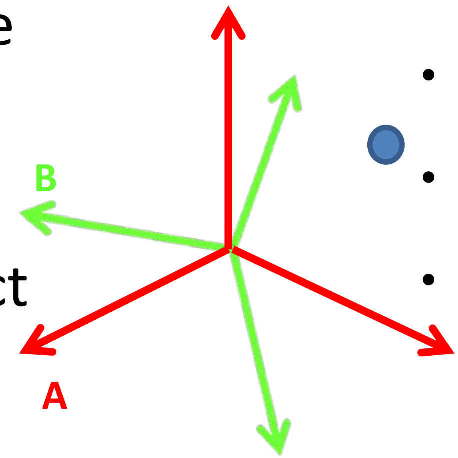


Rotational Velocity

We note that a rotation relates the coordinates of 3D points with

$${}^{A}p(t) = {}^{A}R_B(t)\,{}^{B}p$$

Deriving on both sides with respect to time we get



- Point P is attached to frame B
- Frame B moves wrt to frame A
- Frame A is inertial

$$v_{{}^{A}p}(t) = \frac{d\,{}^{A}p(t)}{dt} = {}^{A}\dot{R}_B\,{}^{B}p$$

$$v_{{}^{A}p}(t) = {}^{A}\dot{R}_B\left({}^{A}R_B^{-1}\,{}^{A}R_B\right){}^{B}p$$

Identity

$$v_{{}^{A}p}(t) = \left({}^{A}\dot{R}_B\,{}^{A}R_B^{-1}\right){}^{A}p$$

This skew symmetric matrix defines the ***spatial angular velocity***

Rotational Velocity

$${}^{A}\dot{R}_{B}\,{}^{A}R_{b}^{-1}$$ is skew symmetric $$\hat{a} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

The instantaneous spatial angular velocity is defined by

$${}^{A}\hat{\omega}_{B} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} = {}^{A}R_{B}\,{}^{A}R_{B}^{-1} \Longrightarrow {}^{A}\omega_{B} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Rotational Velocity

$${}^{A}\dot{R}_{B}\,{}^{A}R_{b}^{-1}$$ is skew symmetric $$\hat{a} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

The instantaneous spatial angular velocity is defined by

$${}^{A}\hat{\omega}_{B} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} = {}^{A}\dot{R}_{B}\,{}^{A}R_{B}^{-1} \Rightarrow {}^{A}\omega_{B} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Rigid Body Velocity

We note that a rotation relates the coordinates of 3D points with

$$ {}^{A}p(t) = \begin{bmatrix} {}^{A}R_{B}(t) & {}^{A}t_{B}(t) \\ 0 & 1 \end{bmatrix} {}^{B}p = {}^{A}E_{B}(t)\ {}^{B}p $$

Just like we did for rotations, deriving on both sides with respect to time we get

$$ v_{{}^{A}p}(t) = ({}^{A}\dot{E}_{B}\ {}^{A}E_{B}^{-1})\,{}^{A}p $$

and we expand the matrices to be

angular velocity

linear velocity

$$ {}^{A}\dot{E}_{B}\ {}^{A}E_{B}^{-1} = \begin{bmatrix} {}^{A}\dot{R}_{B} & {}^{A}\dot{t}_{B} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} {}^{A}R_{B}^{T} & -{}^{A}R_{B}^{T}\ {}^{A}t_{B} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^{A}\dot{R}_{B}\ {}^{A}R_{B}^{T} & -{}^{A}\dot{R}_{B}\ {}^{A}R_{B}^{T}\ {}^{A}t_{B} + {}^{A}\dot{t}_{B} \\ 0 & 0 \end{bmatrix} $$

Rigid Body Velocity

The "***s"patial velocity*** is defined by

$$^{A}\hat{V}_{B}^{s} = {}^{A}\dot{E}_{B}\,{}^{A}E_{B}^{-1}$$

Where the linear velocity is defined by

$$^{A}v_{B}^{s} = -{}^{A}\dot{R}_{B}\,{}^{A}R_{B}^{T}\,{}^{A}t_{B} + {}^{A}\dot{t}_{B}$$

And the angular velocity is define as before by

$$^{A}\hat{\omega}_{B}^{s} = {}^{A}\dot{R}_{B}\,{}^{A}R_{B}^{T}$$

Combining these two we obtain the 6D vector

$$^{A}V_{B}^{s} = \begin{bmatrix} {}^{A}v_{B}^{S} \\ {}^{A}\omega_{B}^{S} \end{bmatrix}$$

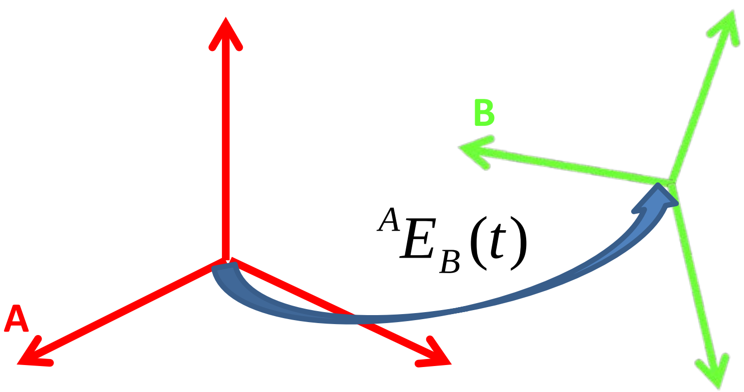
Body Velocity

- If we have $^{A}E_{B}(t)$ but we want to know the velocity of frame B with respect to frame B?

$$^{A}p(t) = {}^{A}E_{B}(t)\,{}^{B}p$$

$$^{A}E_{B}^{-1}\,v_{^{A}p}(t) = {}^{A}E_{B}^{-1}\,{}^{A}\dot{E}_{B}\,{}^{B}p$$

$$v_{^{B}p}(t) = \left({}^{A}E_{B}^{-1}\,{}^{A}\dot{E}_{B}\right){}^{B}p$$



Most intuitive: This is your velocity with respect to yourself

Body Velocity

The "***b*"*ody velocity*** is defined by

$$^{A}\hat{V}_{B}^{b} = {}^{A}E_{B}^{-1}\,{}^{A}\dot{E}_{B} = \begin{bmatrix} {}^{A}R_{B}^{T}\,{}^{A}\dot{R}_{B} & {}^{A}R_{B}^{T}\,{}^{A}\dot{t}_{B} \\ 0 & 0 \end{bmatrix}$$

Where the linear velocity is defined by

$$^{A}v_{B}^{b} = {}^{A}R_{B}^{T}\,{}^{A}\dot{t}_{B}$$

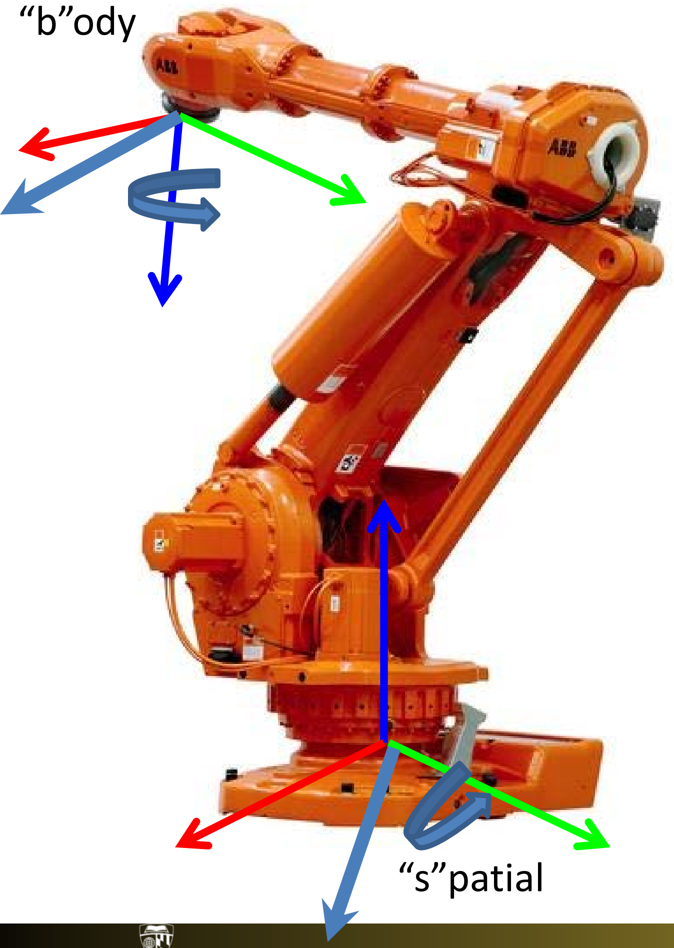
And the angular velocity is define as before by

$$^{A}\hat{\omega}_{B}^{b} = {}^{A}R_{B}^{T}\,{}^{A}\dot{R}_{B}$$

Combining these two we obtain the 6D vector

$$^{A}V_{B}^{b} = \begin{bmatrix} {}^{A}v_{B}^{b} \\ {}^{A}\omega_{B}^{b} \end{bmatrix}$$

Transform Body Velocity to Spatial Velocity



If you are given a body velocity, for example say you want to:

1) Rotate the tool about a given axis (in the tool frame)

2) Drive the tool along a given axis (in the tool frame)

Then you can compute the equivalent velocity in the base frame according to

$$\boxed{\begin{bmatrix} {}^{A}v_{B}^{s} \\ {}^{A}\omega_{B}^{s} \end{bmatrix} = \begin{bmatrix} {}^{A}R_{B} & {}^{A}\hat{t}_{B}\,{}^{A}R_{B} \\ 0 & {}^{A}R_{B} \end{bmatrix} \begin{bmatrix} {}^{A}v_{B}^{b} \\ {}^{A}\omega_{B}^{b} \end{bmatrix}}$$

Kinematics

Cartesian Space Velocity

$[\ v,\ \ w\ ]$

v: linear velocity
w: angular velocity

JACOBIAN

$[\ v, w\ ]^T = J(\boldsymbol{q})\ \dot{\boldsymbol{q}}$

$\dot{\boldsymbol{q}} = J^{-1}(\boldsymbol{q})\ [\ v, w\ ]^T$

INVERSE JACOBIAN

Joint Space Velocity

Joint 1 = $\dot{q}_1$
Joint 2 = $\dot{q}_2$
...
Joint N = $\dot{q}_N$

Rigid body motion
Transformation between coordinate frames ✓

Linear algebra ✓

Manipulator Jacobian

Spatial velocity of the "T"ool frame in the "B"ase frame is

$${}^{B}\hat{V}_{T}^{s} = {}^{B}\dot{E}_{T}(t)\,{}^{B}E_{T}^{-1}(t)$$

Let's change the time varying trajectory ${}^{B}E_{T}(t)$ to be a time varying joint trajectory q(t)

Derivative of the forward kinematics wrt q_i

Inverse of the forward kinematics

$${}^{B}\hat{V}_{T}^{s} = {}^{B}\dot{E}_{T}(\mathbf{q}(t))\,{}^{B}E_{T}^{-1}(\mathbf{q}(t))$$

Applying the chain rule

$$\frac{\partial E(q(t))}{\partial t} = \frac{\partial E(q(t))}{\partial q}\frac{\partial q(t)}{\partial t}$$

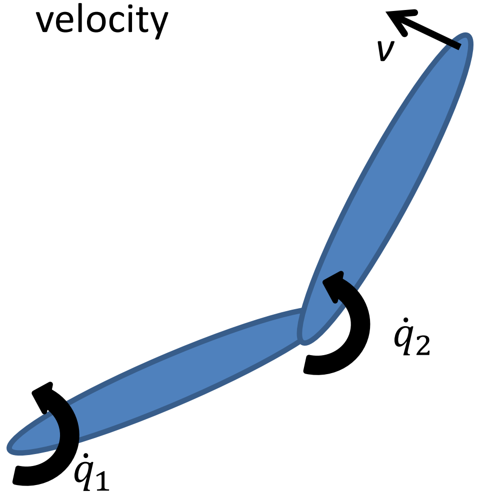
$${}^{B}\hat{V}_{T}^{s} = \sum_{i=1}^{N}\left(\frac{\partial\, {}^{B}E_{T}}{\partial q_i}\dot{q}_i\right){}^{B}E_{T}^{-1}(\mathbf{q}(t))$$

$${}^{B}\hat{V}_{T}^{s} = \sum_{i=1}^{N}\left(\frac{\partial\, {}^{B}E_{T}}{\partial q_i}\;{}^{B}E_{T}^{-1}(\mathbf{q}(t))\right)\dot{q}_i$$

Manipulator Jacobian

$${}^B\hat{V}_T^s = \sum_{i=1}^{N} \left(\frac{\partial\, {}^B E_T}{\partial q_i} \; {}^B E_T^{-1}(\mathbf{q}(t)) \right) \dot{q}_i$$

Sum the contribution of each joint to the tool's velocity



Lets rewrite this result as

$$\boxed{\begin{bmatrix} {}^B v_T^s \\ {}^B \omega_T^s \end{bmatrix} = J(\mathbf{q})\dot{\mathbf{q}}}$$

Where $J(\boldsymbol{q})$ is a 6xN matrix called the <u>*manipulator Jacobian*</u> that relates joint velocities to the Cartesian velocity of the tool. Note that $J(\boldsymbol{q})$ depends on "$\boldsymbol{q}$" and, therefore, on the robot's configuration

Kinematics

JACOBIAN

Cartesian Space
Tool Frame (***T***)
Base Frame (***B***)

$[\ ^{B}v_{T},\ \ ^{B}w_{T}\ ]$

$^{B}v_{T}$:linear vel. of T wrt B
$^{B}w_{T}$: angular vel. of T wrt B

$[\ v,\ w\ ]^{T} = J(\boldsymbol{q})\ \dot{\boldsymbol{q}}$

$\dot{\boldsymbol{q}} = J^{-1}(\boldsymbol{q})\ [\ v,\ w\ ]^{T}$

INVERSE
JACOBIAN

Joint Space

Joint 1 = $\dot{q}_1$
Joint 2 = $\dot{q}_2$
...
Joint N = $\dot{q}_N$

Rigid body motion
Transformation between coordinate frames

Linear algebra

Manipulator Jacobian

We just derived that given a vector of joint velocities, the velocity of the tool as seen in the base of the robot is given by

$$\begin{bmatrix} {}^{B}v_T^s \\ {}^{B}\omega_T^s \end{bmatrix} = J(\mathbf{q})\dot{\mathbf{q}}$$

If, instead we want the tool to move with a velocity expressed in the **base** frame, the corresponding joint velocities can be computed by

$$\dot{\mathbf{q}} = J^{-1}(\mathbf{q})\begin{bmatrix} {}^{B}v_T^s \\ {}^{B}\omega_T^s \end{bmatrix}$$

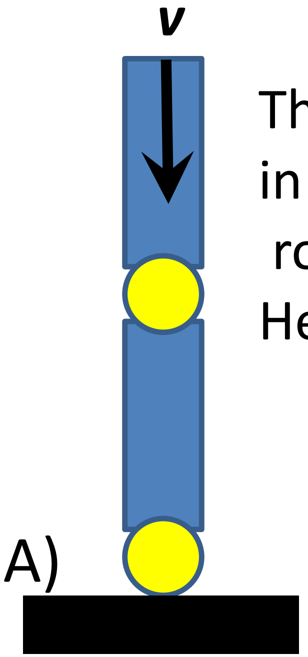
Inverting a matrix is much easier than computing the inverse kinematics!

Manipulator Jacobian

What if the Jacobian has no inverse?

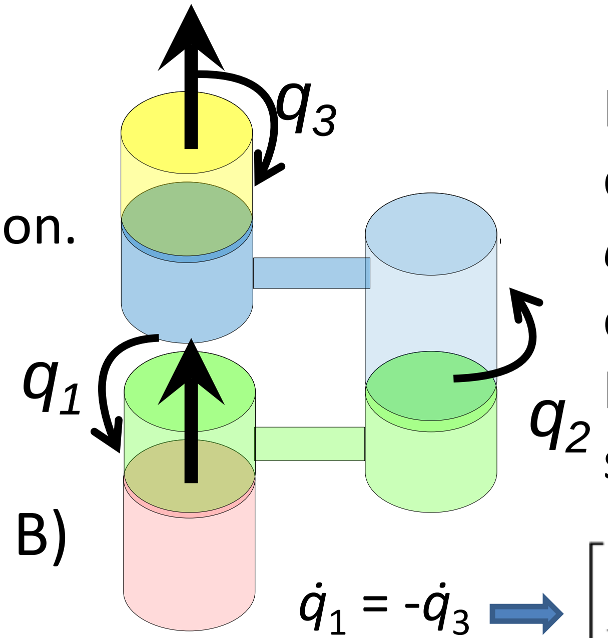
A) No solution: The velocity is impossible

B) Infinity of solutions: Some joints can be moved without affecting the velocity (i.e. when two axes are colinnear)



A)

The robot cannot move in this direction when the robot is in this configuration. Hence $J(\boldsymbol{q})$ is singular.



B)

In this configuration, q_1 and q_3 can counter rotate. Hence $J(\boldsymbol{q})$ is singular.

$\dot{q}_1 = -\dot{q}_3 \Rightarrow \begin{bmatrix} {}^B\mathbf{v} \\ {}^B\boldsymbol{\omega} \end{bmatrix} = \mathbf{0}$