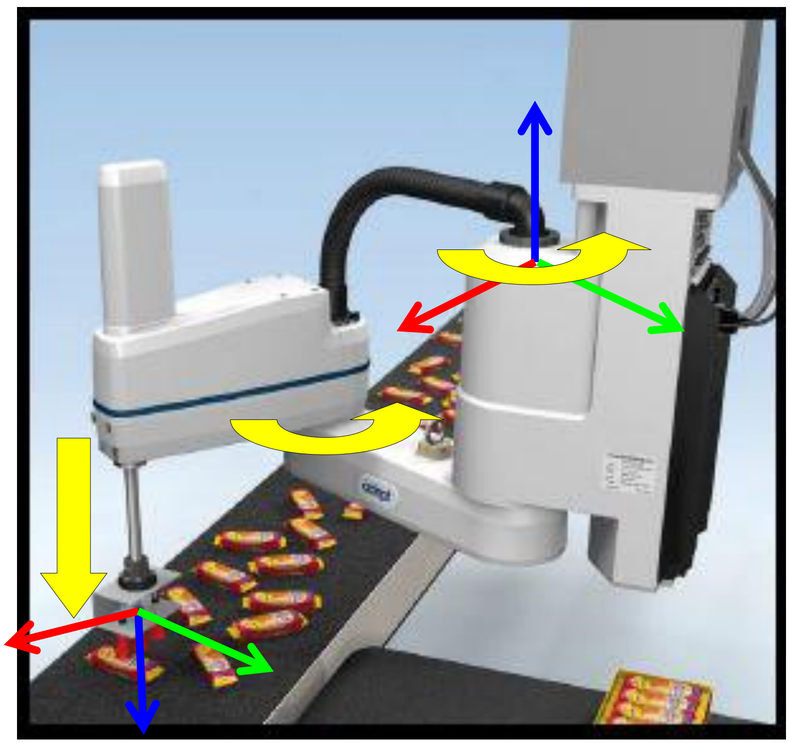
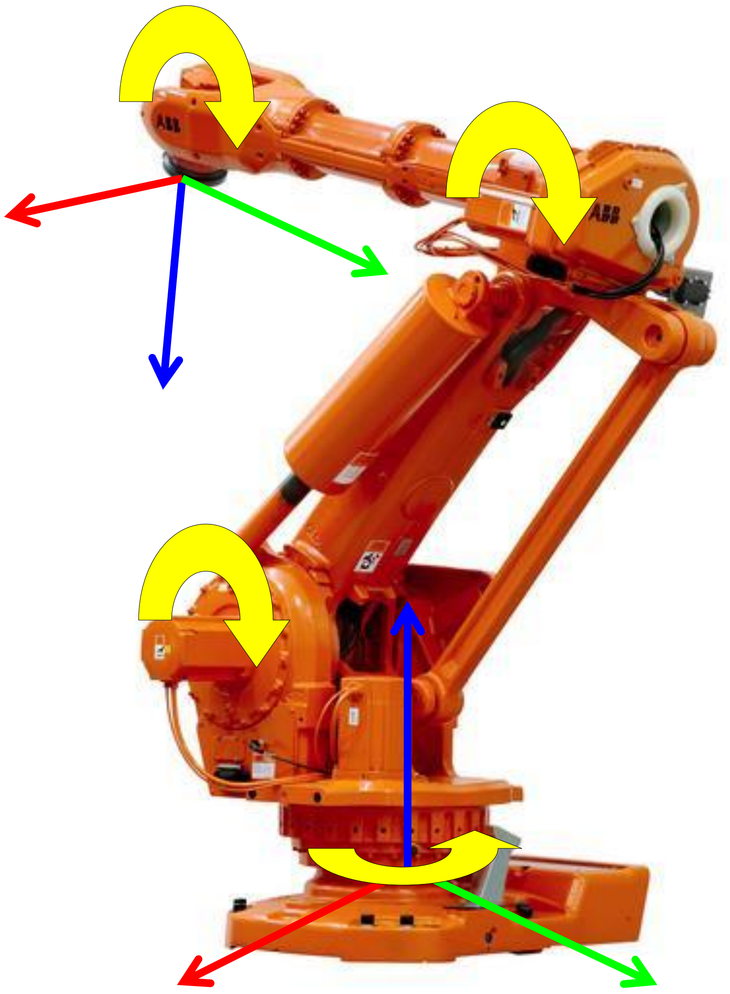
Robot Kinematics

Robot Manipulators

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

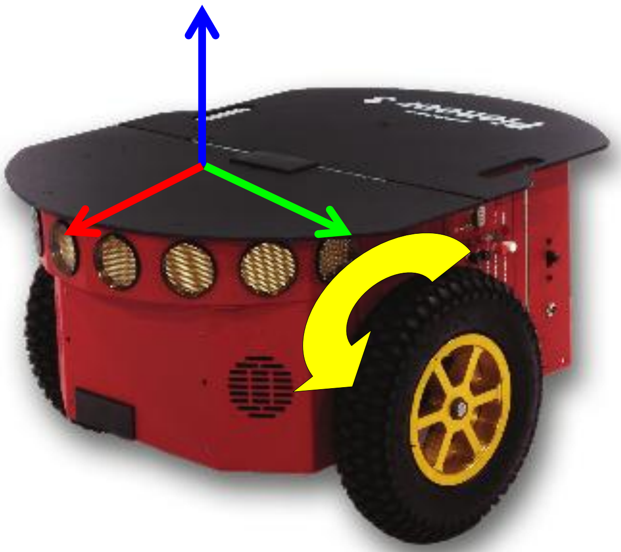


SCARA



6 axes robot arm

Other Robots

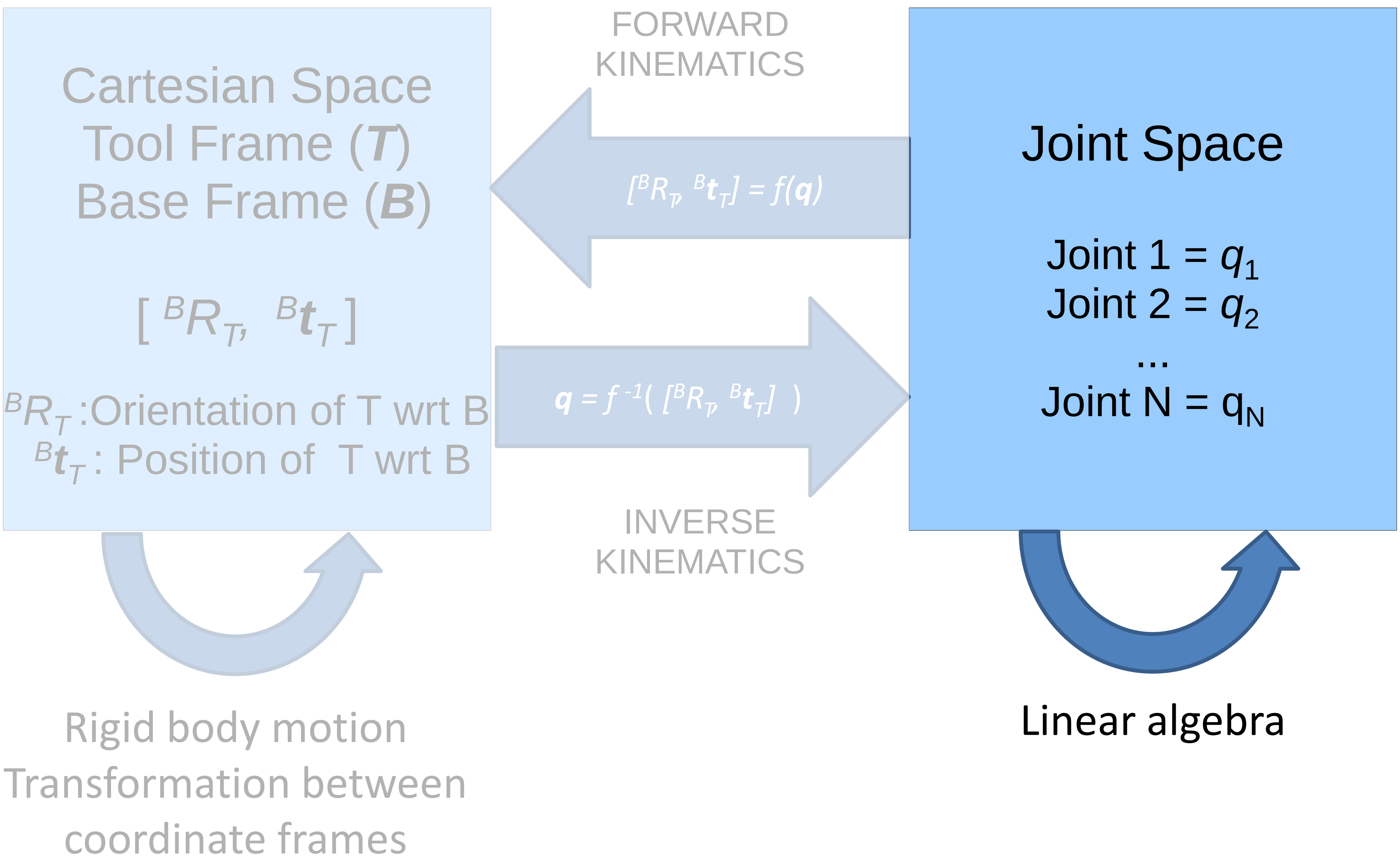


Mobile robots

Delta Robot

Stewart Platform

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

Transformation Within Joint Space

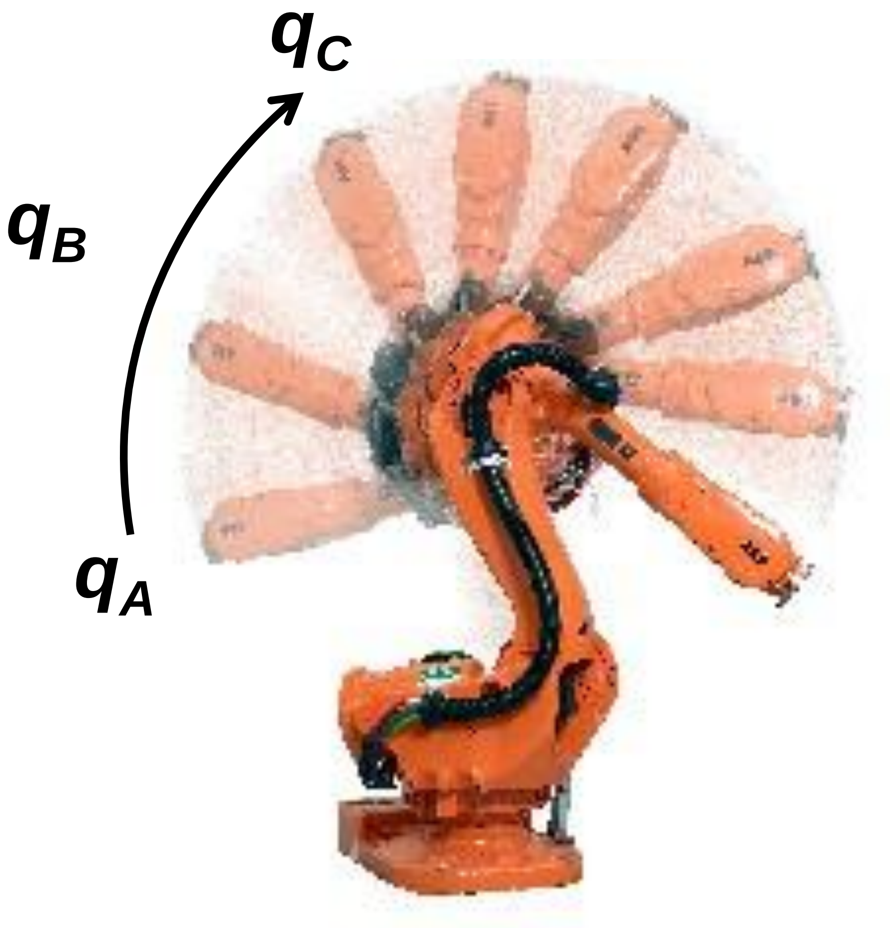
- Joint spaces are typically defined in $\boldsymbol{R}^n$

  Thus for a vector

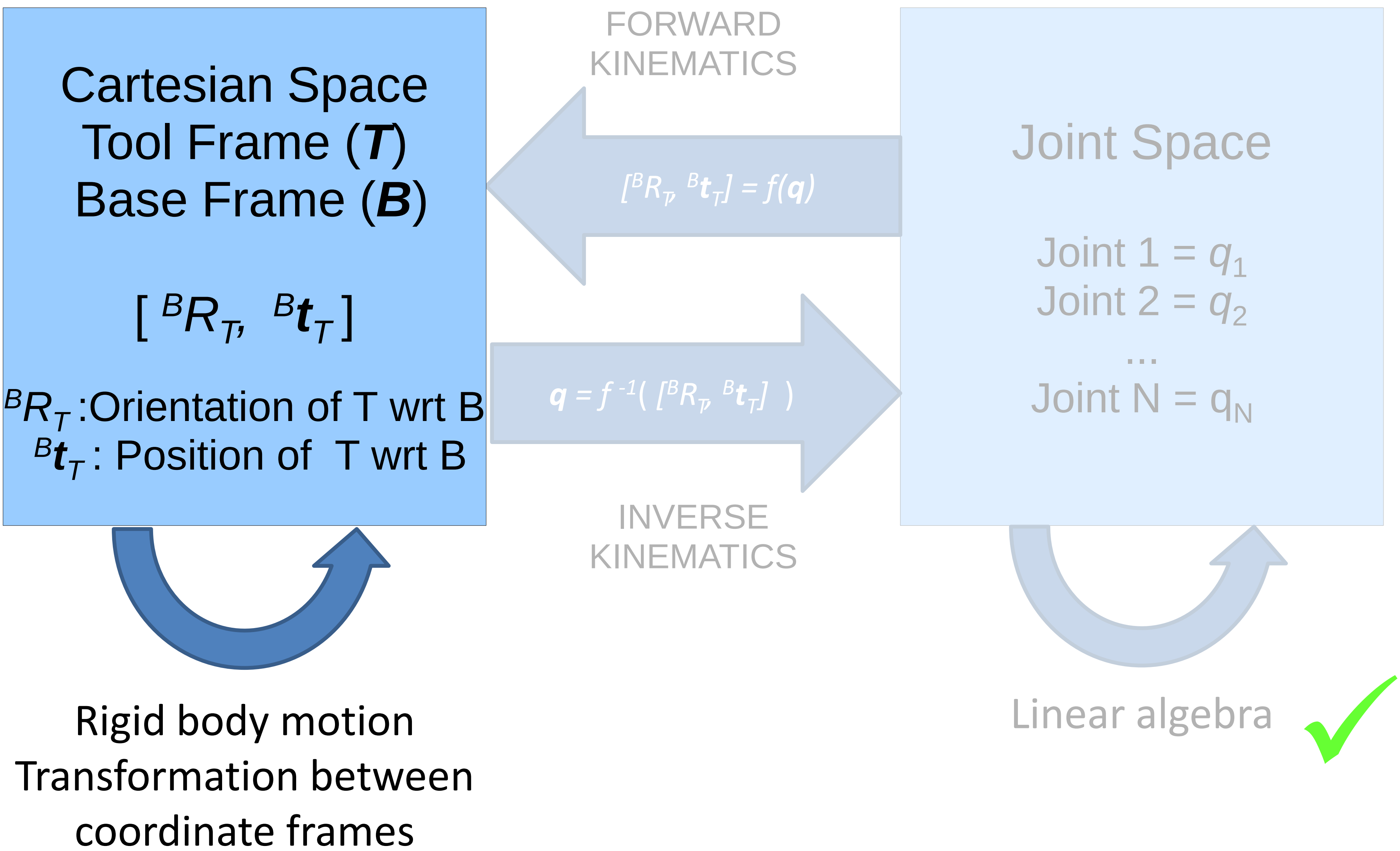
  $$\mathbf{q} = \begin{bmatrix} q_1 & \dots & q_n \end{bmatrix}$$

  we can use additions subtractions

  $$\mathbf{q}_c = \mathbf{q}_a + \mathbf{q}_b$$



Kinematics

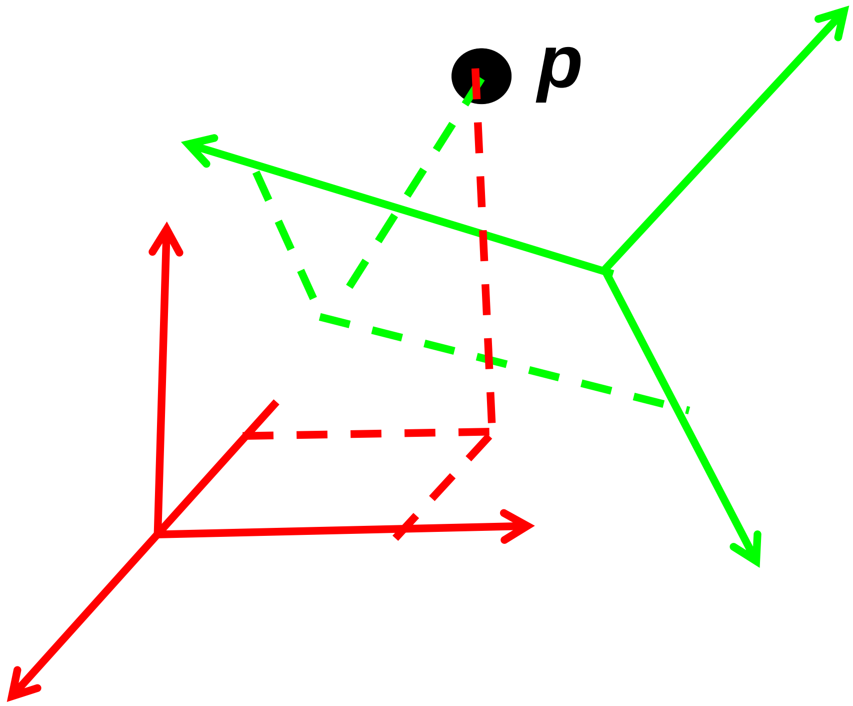


Cartesian Space
Tool Frame (*T*)
Base Frame (*B*)
$[\,{}^{B}R_{T},\ {}^{B}\mathbf{t}_{T}\,]$
${}^{B}R_{T}$:Orientation of T wrt B
${}^{B}\mathbf{t}_{T}$: Position of T wrt B
FORWARD KINEMATICS
$[{}^{B}R_{T},\ {}^{B}\mathbf{t}_{T}] = f(\mathbf{q})$
$\mathbf{q} = f^{-1}(\ [{}^{B}R_{T},\ {}^{B}\mathbf{t}_{T}]\ )$
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
Position and Orientation

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

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

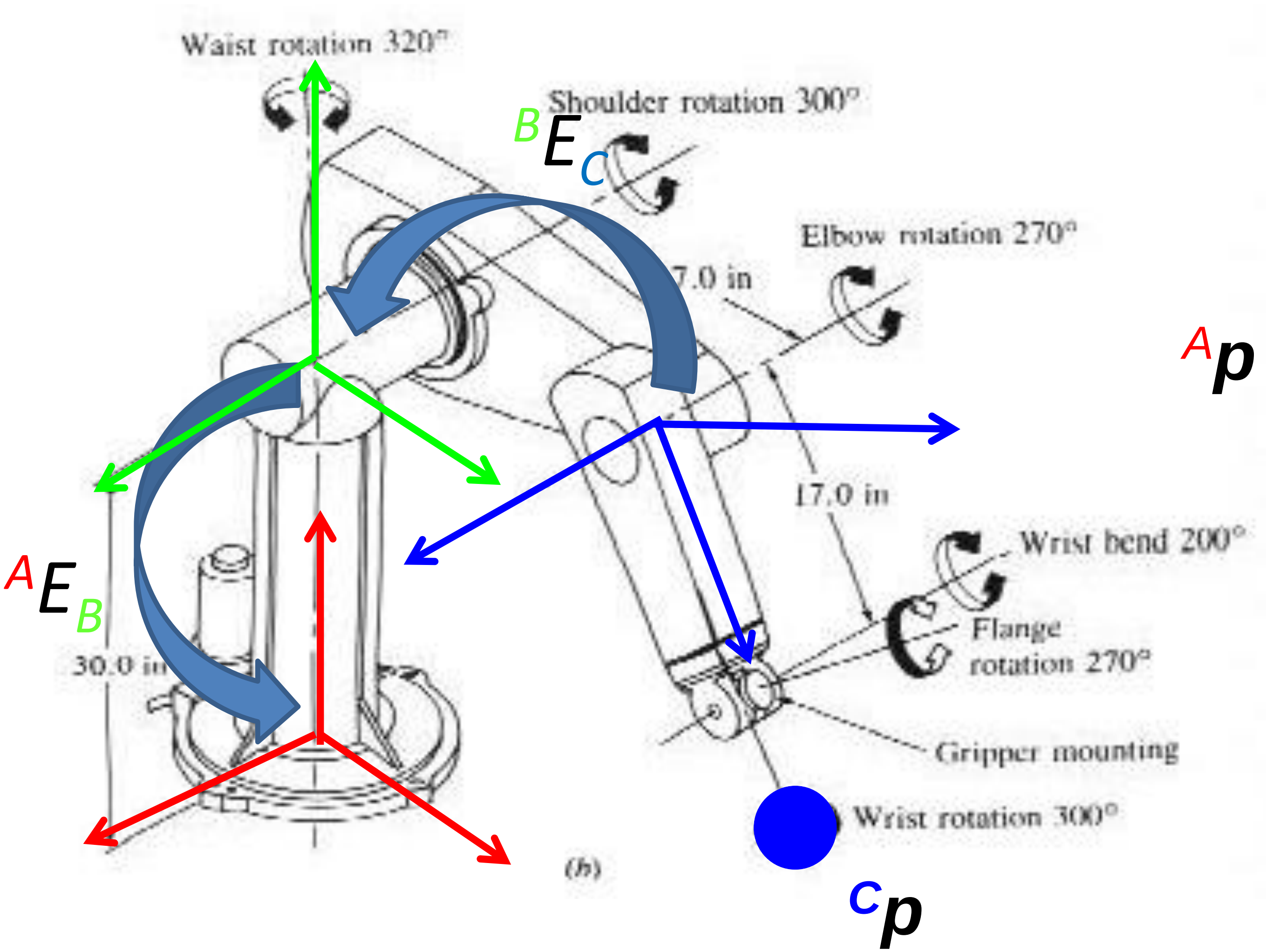


$${}^{A}\boldsymbol{p} = {}^{A}R_{B}\,{}^{B}\boldsymbol{p} + {}^{A}\mathbf{t}_{B}$$

Homogeneous representation

$${}^{A}\boldsymbol{p} = {}^{A}E_{B}\,{}^{B}\boldsymbol{p}$$

Cartesian Transformation Kinematic Chain



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

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

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

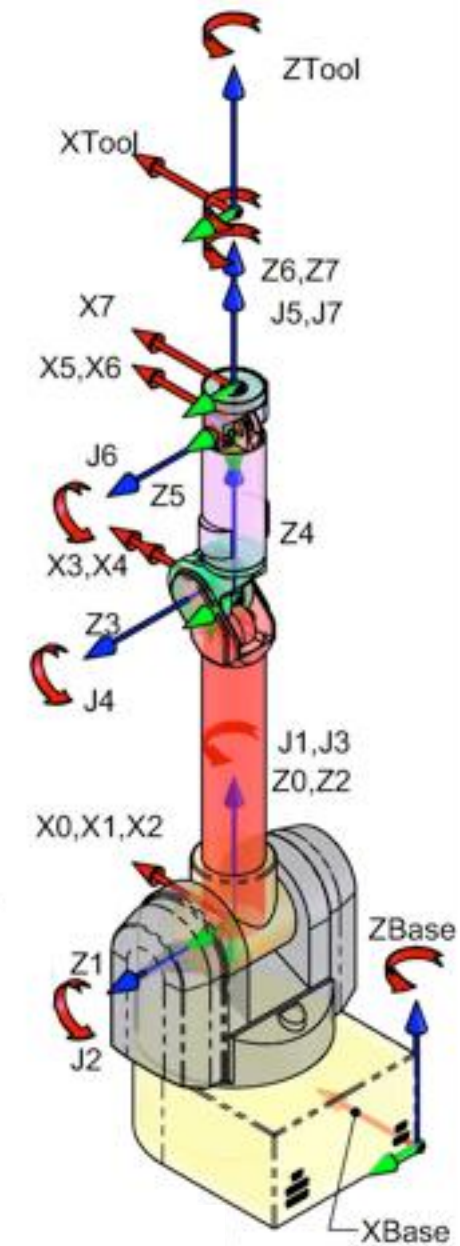
Rigid body motion
Transformation between coordinate frames ✓

Linear algebra ✓

Forward Kinematics

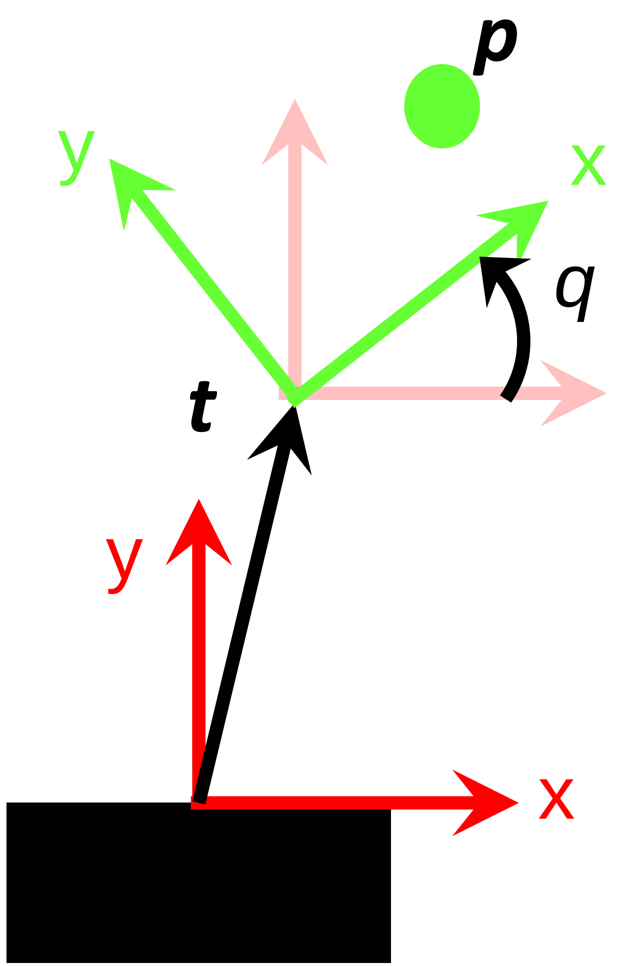
Guidelines for assigning frames:

- There are several conventions
  - Denavit Hartenberg (DH), modified DH, Hayati, etc.
- Choose the base and tool coordinate frame
  - Make your life easy!
- Start from the base and move towards the tool
  - Make your life easy!
  - In general each actuator has a coordinate frame.
- Align each coordinate frame with a joint actuator



Barrett WAM

Forward Kinematics 2D

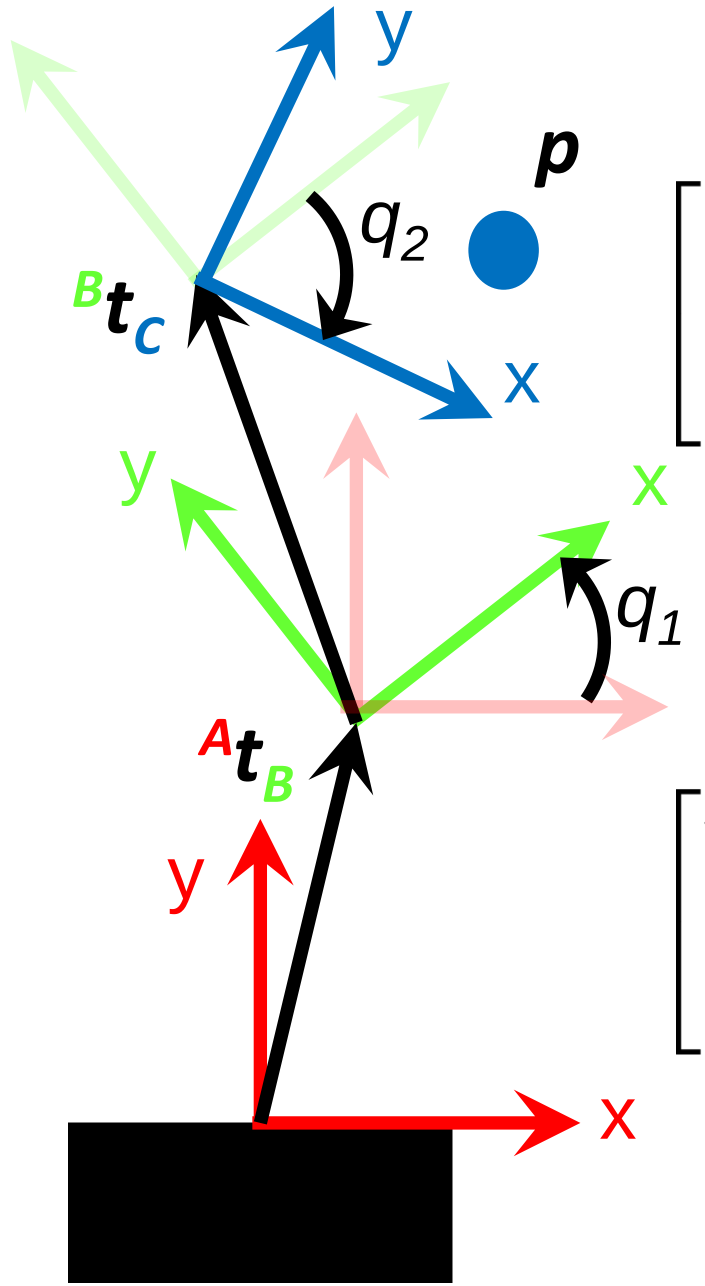


$$^{A}\boldsymbol{p} = {}^{A}E_{B}\,{}^{B}\boldsymbol{p}$$

$$\begin{bmatrix} ^{A}x \\ ^{A}y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos q & -\sin q & t_x \\ \sin q & \cos q & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Forward Kinematics}} \begin{bmatrix} ^{B}x \\ ^{B}y \\ 1 \end{bmatrix}$$

Forward Kinematics

Forward Kinematics 2D



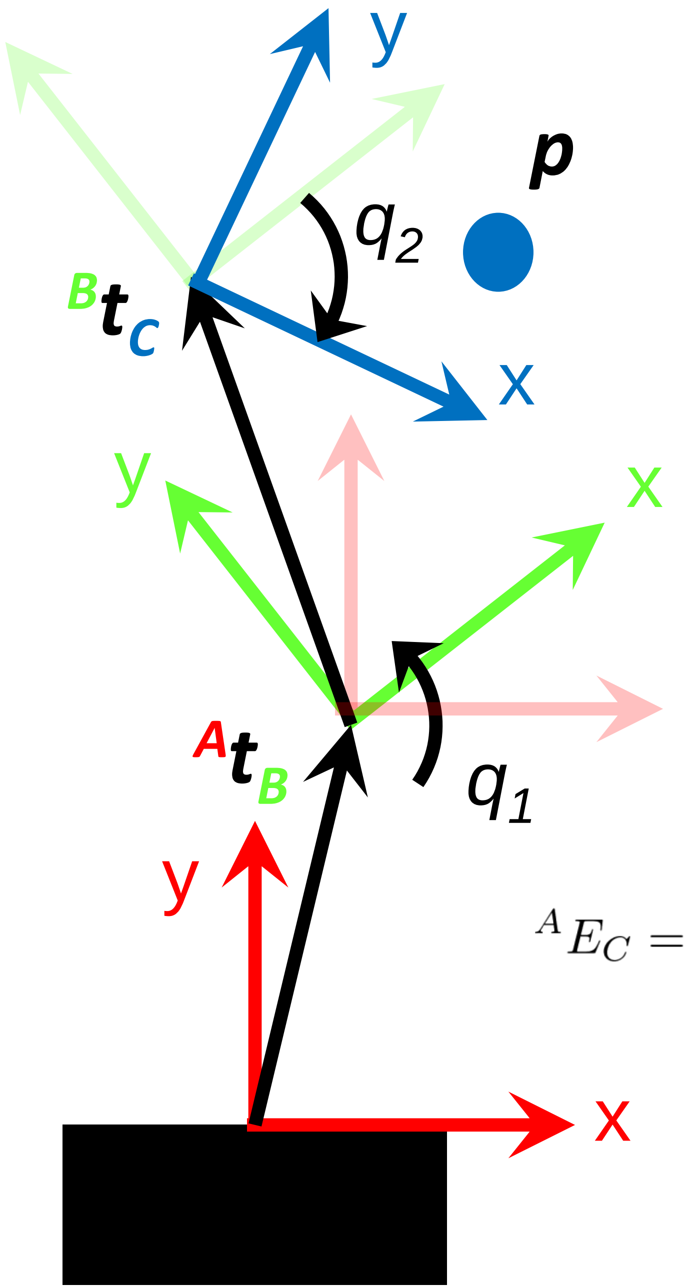
$$ {}^{B}\mathbf{p} = {}^{B}E_{C}\,{}^{C}\mathbf{p} $$

$$ \begin{bmatrix} {}^{B}x \\ {}^{B}y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos q_2 & -\sin q_2 & {}^{B}t_x \\ \sin q_2 & \cos q_2 & {}^{B}t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{C}x \\ {}^{C}y \\ 1 \end{bmatrix} $$

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

$$ \begin{bmatrix} {}^{A}x \\ {}^{A}y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos q_1 & -\sin q_1 & {}^{A}t_x \\ \sin q_1 & \cos q_1 & {}^{A}t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{B}x \\ {}^{B}y \\ 1 \end{bmatrix} $$

Forward Kinematics 2D



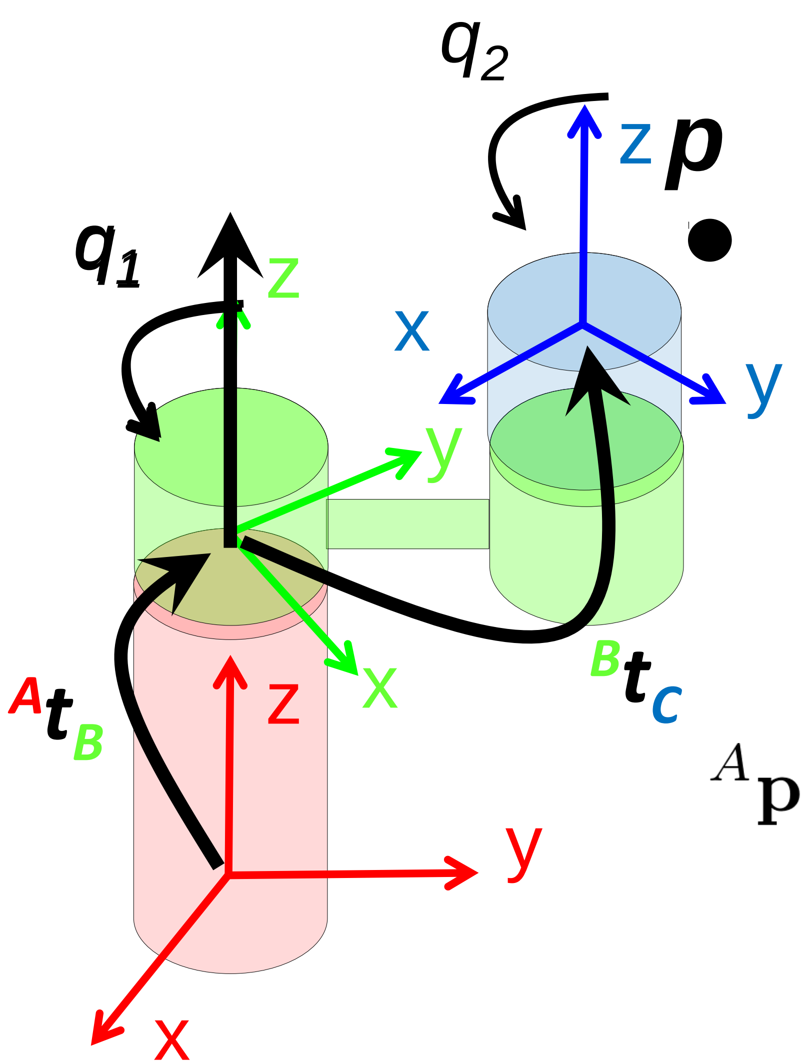
Substituting ${}^B\boldsymbol{p} = {}^BE_C\,{}^C\boldsymbol{p}$ in ${}^A\boldsymbol{p} = {}^AE_B\,{}^B\boldsymbol{p}$

gives ${}^A\boldsymbol{p} = {}^AE_B\,{}^BE_C\,{}^C\boldsymbol{p}$

${}^A\boldsymbol{p} = {}^AE_C\,{}^C\boldsymbol{p}$

$$
{}^AE_C = \underbrace{\begin{bmatrix} \cos(q_1+q_2) & -\sin(q_1+q_2) & {}^At_x +{}^B t_x\cos(q_1) -{}^B t_y\sin(q_1) \\ \sin(q_1+q_2) & \cos(q_1+q_2) & {}^At_y +{}^B t_y * \cos(q_1) +{}^B t_x\sin(q_1) \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Forward Kinematics}}
$$

Forward Kinematics 3D



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

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

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

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

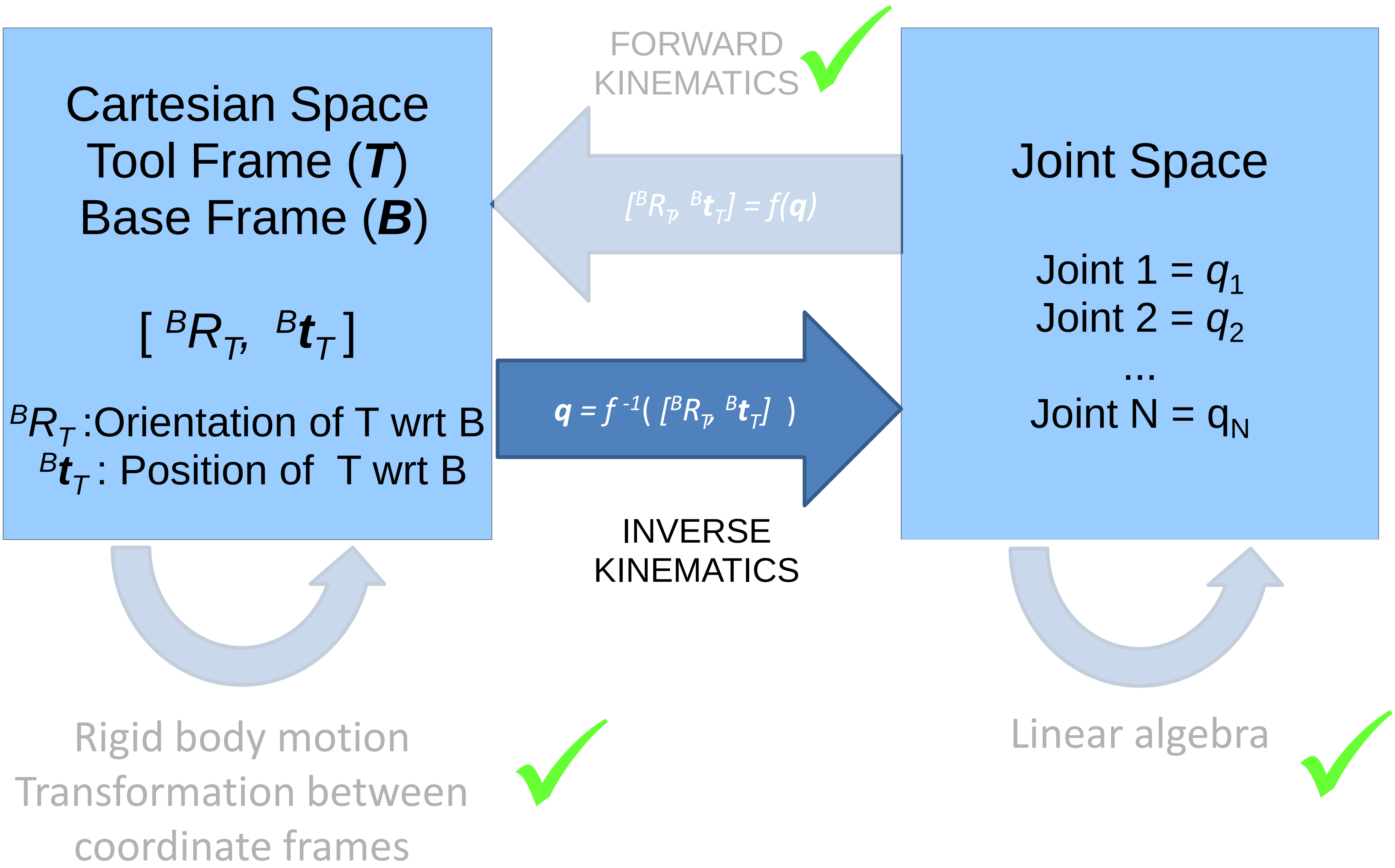
$$= \begin{bmatrix} R_z(q_2) & {}^{B}\mathbf{t}_C \\ \mathbf{0} & 1 \end{bmatrix}$$

$$^{A}\mathbf{p} = {}^{A}E_{B}\ {}^{B}E_{C}^{C}\mathbf{p}$$

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

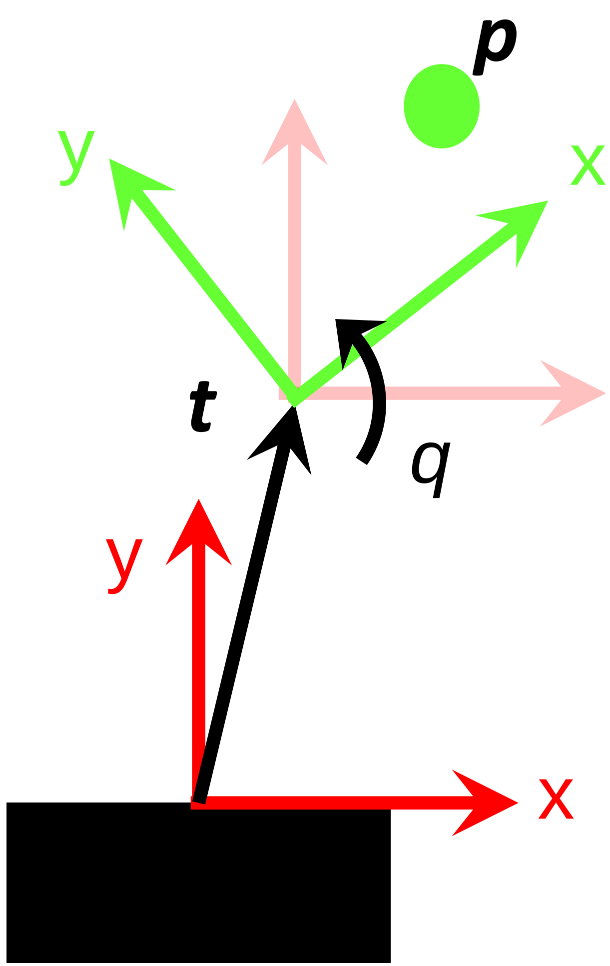
Forward Kinematics

Kinematics



Cartesian Space
Tool Frame (T)
Base Frame (B)
[BRT, BtT]
BRT :Orientation of T wrt B
BtT : Position of T wrt B
FORWARD KINEMATICS
[BRT, BtT] = f(q)
q = f -1([BRT, BtT])
INVERSE KINEMATICS
Joint Space
Joint 1 = q1
Joint 2 = q2
...
Joint N = qN
Rigid body motion
Transformation between coordinate frames
Linear algebra

Inverse Kinematics 2D



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

Given ${}^{A}\boldsymbol{p}$, ${}^{B}\boldsymbol{p}$, ${}^{A}\boldsymbol{t}_{B}$ find q:

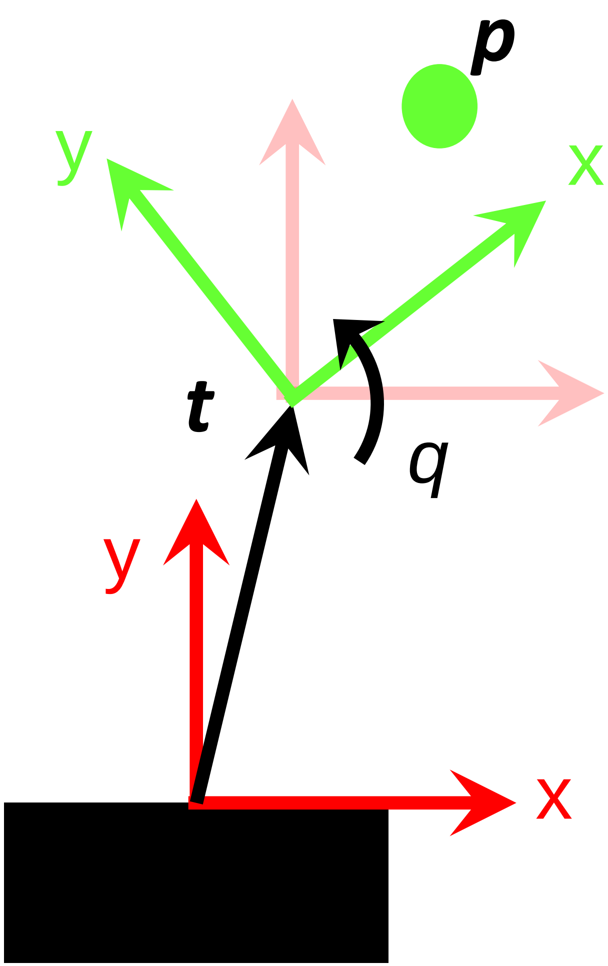
$$ {}^{A}x - t_x = {}^{B}x \cos q - {}^{B}y \sin q $$

$$ a \sin q + b \cos q = c $$

$$ a \sin q + b \cos q = \sqrt{a^2 + b^2} \sin(x + \alpha) $$

$$ \alpha = \begin{cases} \tan^{-1}(b/a) & \text{if } a > 0 \\ \pi + \tan^{-1}(b/a) & \text{if } a < 0 \end{cases} $$

Inverse Kinematics 2D



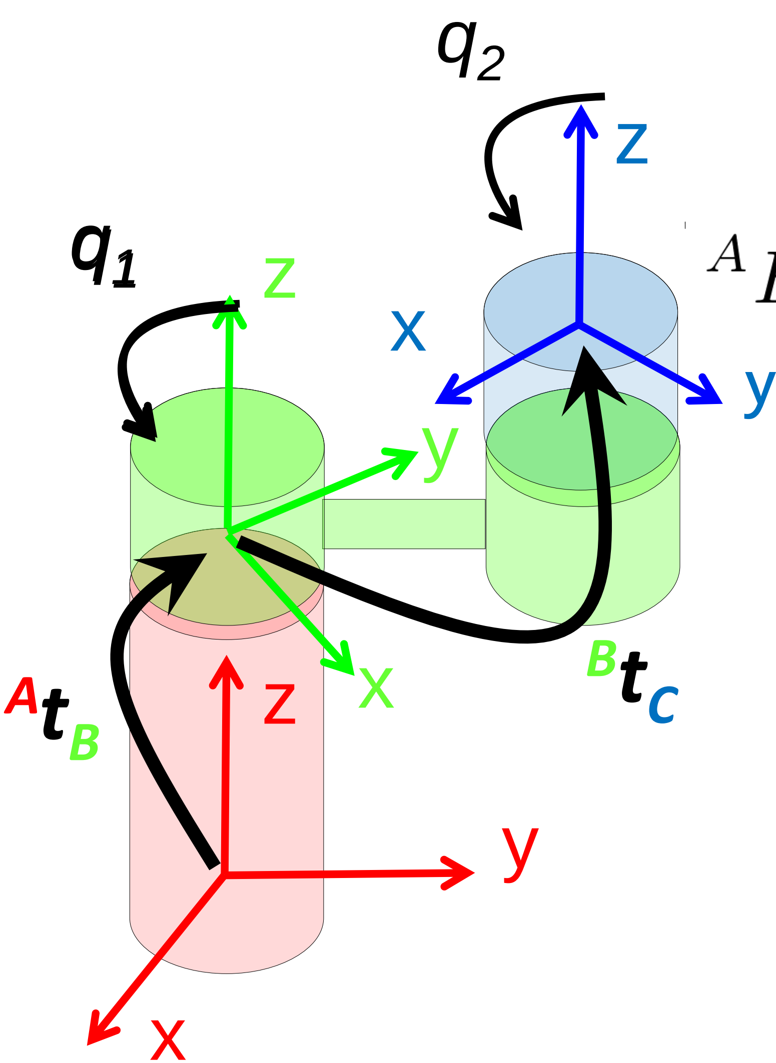
In practice, however, we are interested in solving the inverse kinematics for the basis vectors

$$^{B}\boldsymbol{p} = [\ 0\ 0\ ]^{T}$$
$$^{B}\boldsymbol{p} = [\ 1\ 0\ ]^{T}$$
$$^{B}\boldsymbol{p} = [\ 0\ 1\ ]^{T}$$

which gives the friendlier solution (using $^{B}\boldsymbol{p} = [\ 1\ 0\ ]^{T}$)

$$\mathrm{acos}(\ {}^{A}x - t_x\ ) = q$$

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!

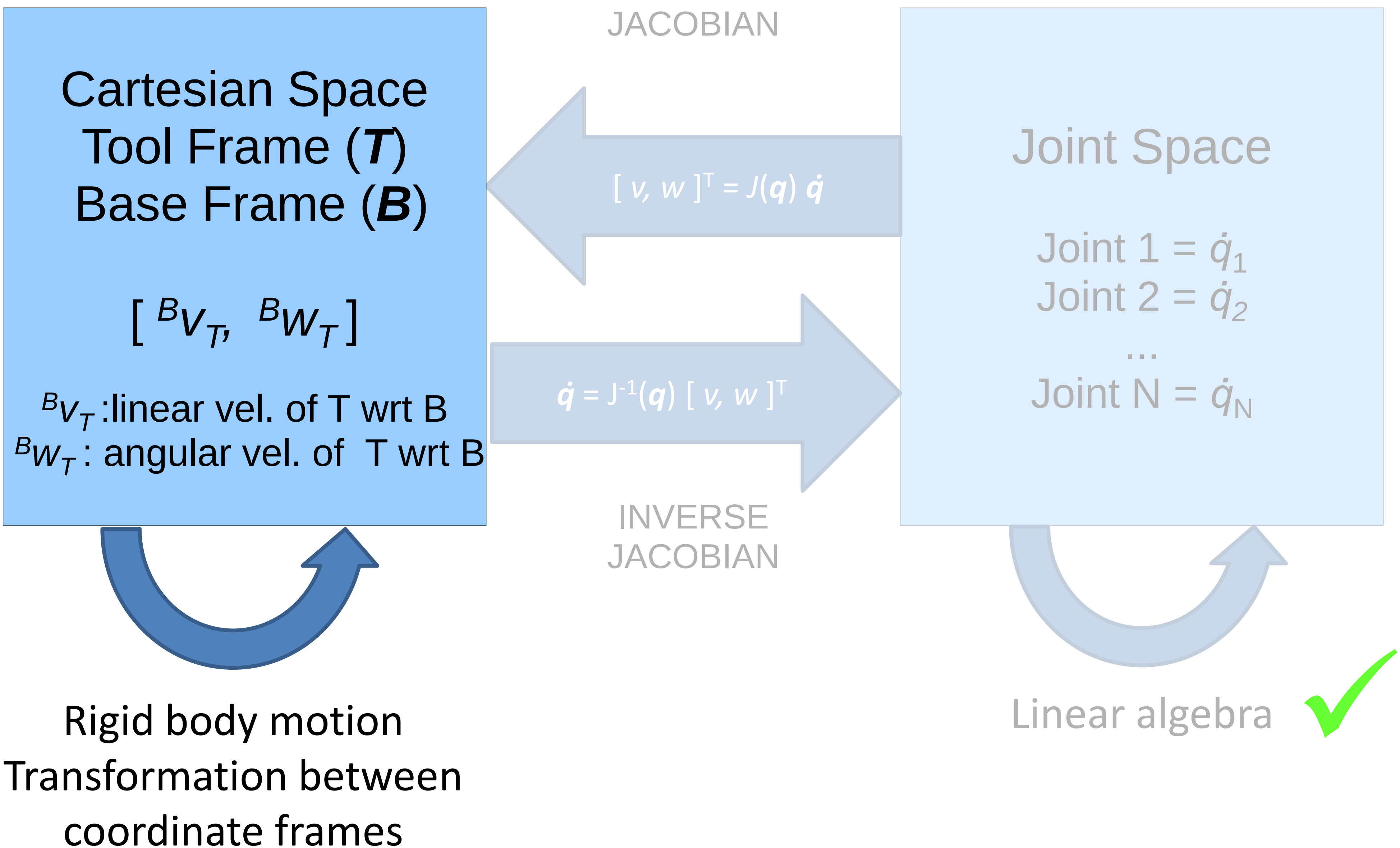
A) For a robot with several joints, a symbolic solution can be difficult to get

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

Note that the inverse kinematics is NOT

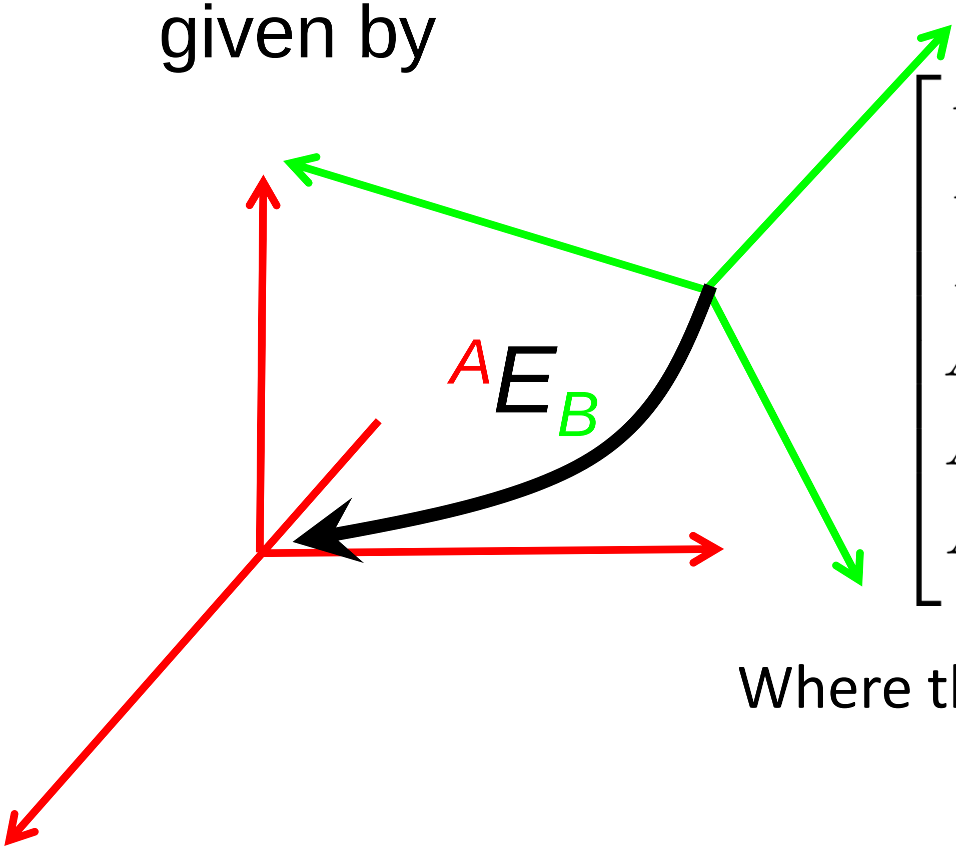
$$ {}^{A}E_{C}^{-1} = {}^{C}E_{A} $$

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

Cartesian Transformation
Linear and Angular Velocities

Given two coordinate systems A and B related by the transformation ${}^{A}E_{B}$, the velocity between A and B is given by



$$\begin{bmatrix} {}^{A}v_x \\ {}^{A}v_y \\ {}^{A}v_z \\ {}^{A}w_x \\ {}^{A}w_y \\ {}^{A}w_z \end{bmatrix} = \begin{bmatrix} {}^{A}R_B & {}^{A}\hat{\mathbf{t}}_B \; {}^{A}R_B \\ 0 & {}^{A}R_B \end{bmatrix} \begin{bmatrix} {}^{B}v_x \\ {}^{B}v_y \\ {}^{B}v_z \\ {}^{B}w_x \\ {}^{B}w_y \\ {}^{B}w_z \end{bmatrix}$$

Where the “^” indicates a skew symmetric matrix

Kinematics

JACOBIAN

Cartesian Space
Tool Frame ($\boldsymbol{T}$)
Base Frame ($\boldsymbol{B}$)

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

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

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

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

Joint Space

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

INVERSE
JACOBIAN

Rigid body motion
Transformation between
coordinate frames ✓

Linear algebra ✓

Manipulator Jacobian

<u>Recall</u>: The linear/angular velocity of the tool frame *T* in the base frame *B*

$$
{}^{B}\hat{V} = \begin{bmatrix} 0 & -{}^{B}\omega_z & {}^{B}\omega_y & {}^{B}v_x \\ {}^{B}\omega_z & 0 & -{}^{B}\omega_x & {}^{B}v_y \\ -{}^{B}\omega_y & {}^{B}\omega_x & 0 & {}^{B}v_z \\ 0 & 0 & 0 & 0 \end{bmatrix} = {}^{B}\dot{E}_T(t)\ {}^{T}E_B(t)
$$

The "∨" operator is to extract the meaningful information from $\hat{V}$

$$
{}^{B}\hat{V}^{\vee} = \begin{bmatrix} 0 & -{}^{B}\omega_z & {}^{B}\omega_y & {}^{B}v_x \\ {}^{B}\omega_z & 0 & -{}^{B}\omega_x & {}^{B}v_y \\ -{}^{B}\omega_y & {}^{B}\omega_x & 0 & {}^{B}v_z \\ 0 & 0 & 0 & 0 \end{bmatrix}^{\vee} = \begin{bmatrix} {}^{B}v_x \\ {}^{B}v_y \\ {}^{B}v_z \\ {}^{B}\omega_x \\ {}^{B}\omega_y \\ {}^{B}\omega_z \end{bmatrix} \begin{matrix} \text{linear} \\ \\ \text{angular} \end{matrix}
$$

Manipulator Jacobian

We change the time varying trajectory to be a time varying joint trajectory

$$^{B}\hat{V} = \ ^{B}\dot{E}_T(t)\ ^{T}E_B(t)$$

$$^{B}V = \ ^{B}\dot{E}_T(\mathbf{q}(t))\ ^{T}E_B(\mathbf{q}(t))$$

Applying the chain rule

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

$$^{B}\hat{V} = \sum_{i=1}^{N}\left(\frac{\partial\ ^{B}E_T}{\partial q_i}\dot{q}_i\right)\ ^{T}E_B(\mathbf{q}(t))$$

Derivative of the forward kinematics wrt q_i

Inverse of the forward kinematics

Manipulator Jacobian

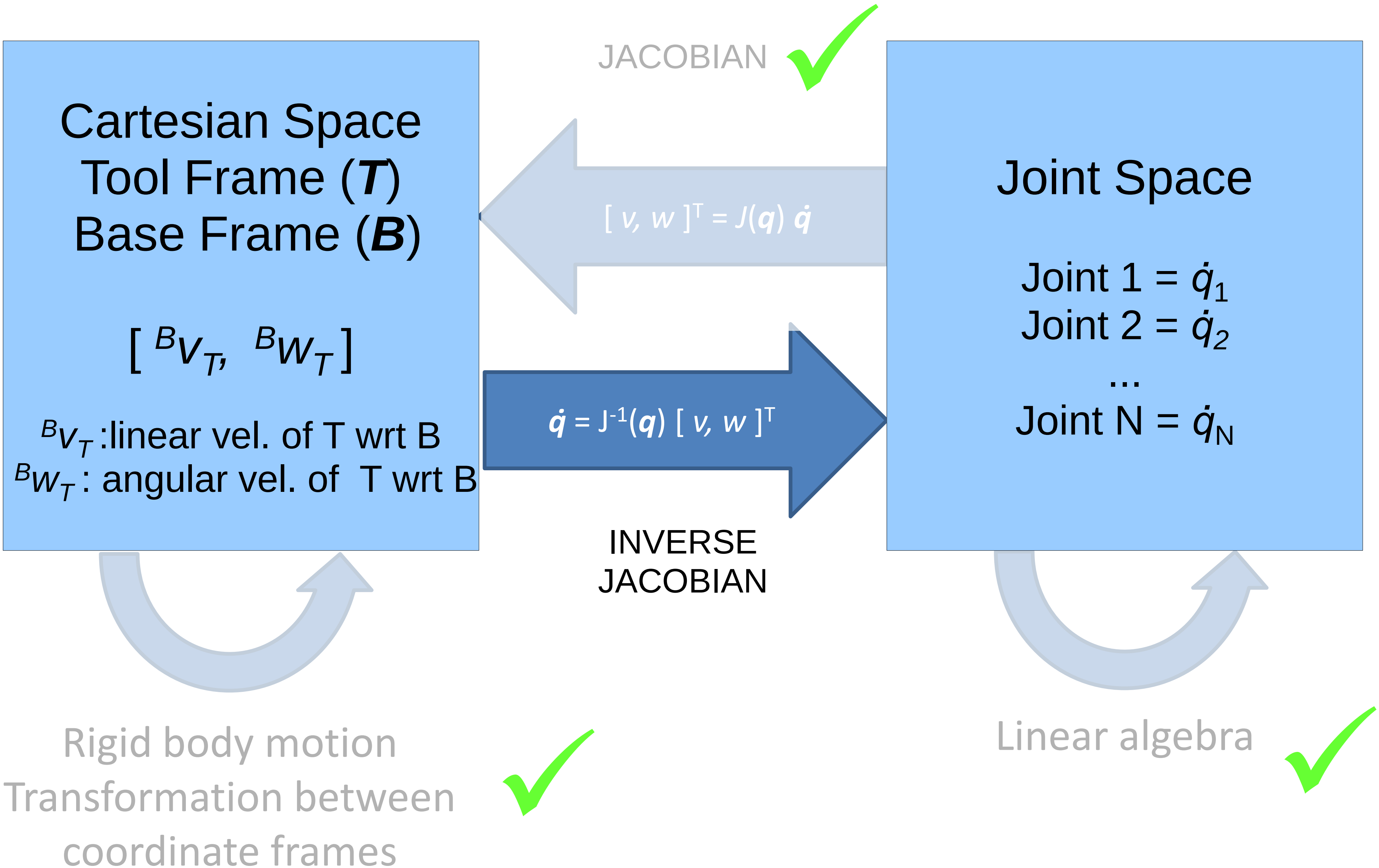
Lets rewrite the previous result as

$$\begin{bmatrix} {}^B v_x \\ {}^B v_y \\ {}^B v_z \\ {}^B \omega_x \\ {}^B \omega_y \\ {}^B \omega_z \end{bmatrix} = \mathrm{J(\ q\ )} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_N \end{bmatrix}$$

Where $J(\boldsymbol{q})$ is a 6xN matrix called the manipulator Jacobian that relates joint velocities to Cartesian velocities

$$J(\mathbf{q}) = \left[\left(\frac{\partial\ {}^B E_T}{\partial q_1} {}^T E_B \right)^{\vee} \quad \dots \quad \left(\frac{\partial\ {}^B E_T}{\partial q_N} {}^T E_B \right)^{\vee} \right]$$

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

$$\begin{bmatrix} {}^B\mathbf{v} \\ {}^B\boldsymbol{\omega} \end{bmatrix} = J(\mathbf{q})\dot{\mathbf{q}}$$

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

$$\dot{\mathbf{q}} = J^{-1}(\mathbf{q}) \begin{bmatrix} {}^B\mathbf{v} \\ {}^B\boldsymbol{\omega} \end{bmatrix}$$

Manipulator Jacobian

If, instead we want to tool to move with a velocity expressed in the **tool** frame, we can first transform the velocity in the **base** frame and then use the inverse Jacobian to compute joint velocities

$$\begin{bmatrix} {}^B\mathbf{v} \\ {}^B\boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} {}^BR_T & {}^B\hat{\mathbf{t}}_T \, {}^BR_T \\ 0 & {}^BR_T \end{bmatrix} \begin{bmatrix} {}^T\mathbf{v} \\ {}^T\boldsymbol{\omega} \end{bmatrix}$$

$$\dot{\mathbf{q}} = J^{-1}(\mathbf{q}) \begin{bmatrix} {}^B\mathbf{v} \\ {}^B\boldsymbol{\omega} \end{bmatrix}$$

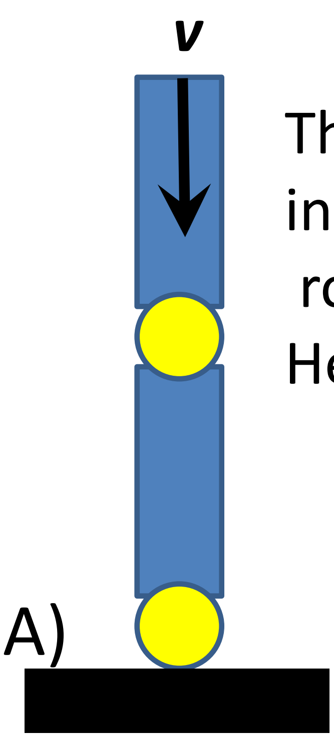
Manipulator Jacobian

What if the Jacobian has no inverse?

A) No solution: The velocity is impossible

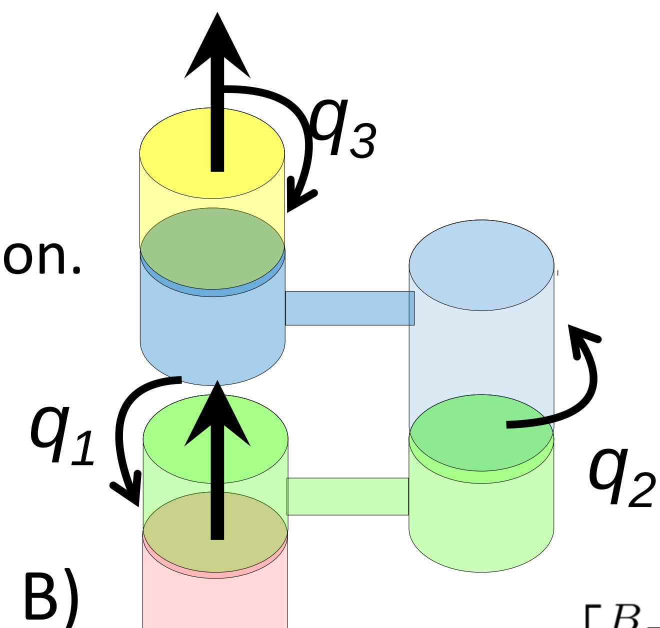
B) Infinity of solutions: Some joints can be moved without affecting the velocity (i.e. when two axes are colinnear)

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.

$\dot{q}_1 = -\dot{q}_3$

$$\begin{bmatrix} {}^{B}\mathbf{v} \\ {}^{B}\boldsymbol{\omega} \end{bmatrix} = \mathbf{0}$$