ECE 470: Introduction to Robotics
Week 04

Liangjing Yang
Assistant Professor, ZJU-UIUC Institute
liangjingyang@intl.zju.edu.cn

Wechat ID: Liangjing_Yang

Schedule Check

- **Lecture**

O. Overview
- Science & Engineering in Robotics

I. Spatial Representation & Transformation
- Coordinate Systems; Pose Representations; Homogeneous Transformations

II. Kinematics
- Multi-body frame assignment; D-H Convention; Joint-space; Work-space; Forward/Inverse Kinematics

WK 04 → III. Velocity Kinematics and Static Forces
- Translational/Rotational Velocity; Joint torque; Generalized Force Coordinates; Jacobian; Singularity

Fundamentals
Week 1-4
Revision/ Quiz on Week 5

IV. Dynamics
- Lagrangian Formulation; Newton-Euler Equations of Motion

V. Planning
- Joint-based Motion Planning; Cartesian-based Path Planning

VI. Control
- Independent Joint/Feedforward/Inverse Dynamics Controls; Controller Architectures

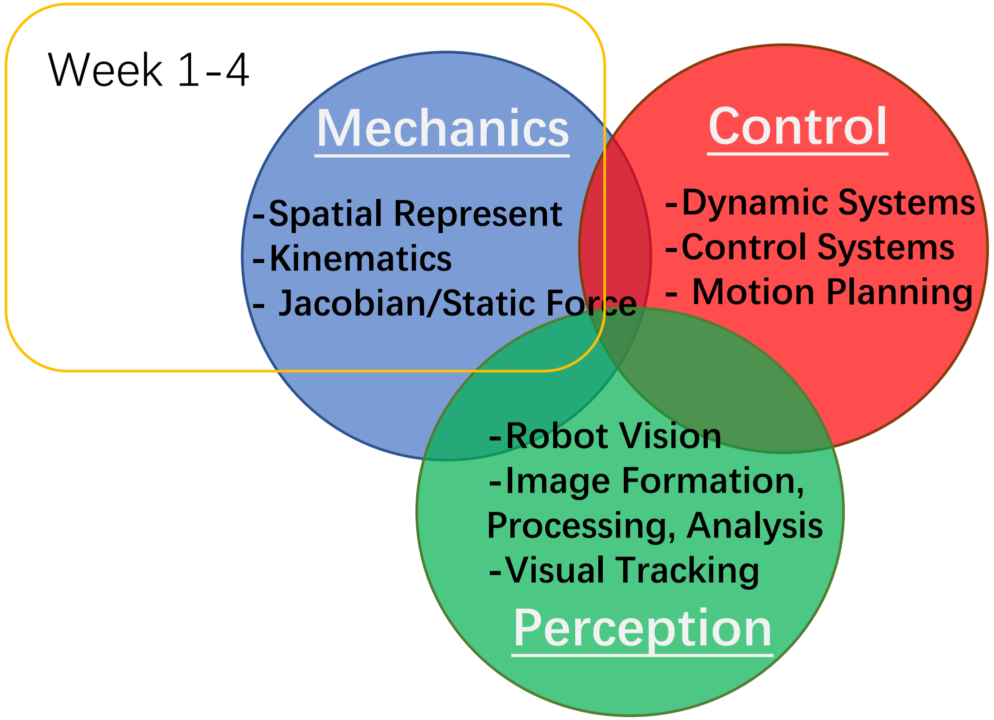
Essentials
Week 6-9
Revision/ Quiz on Week 10

VII. Robot Vision (and Perception)
- Image Formation; Image Processing; Visual Tracking & Pose Estimation; Vision-based Control & Image-guided robotics

Applied
Week 11-14
Reading Wk/ Exam on Week 15-16

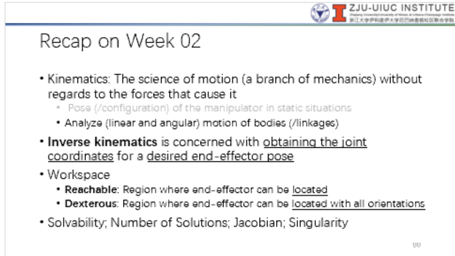
Relooking at the big Picture

- *Robot Mechanics, Control, Planning & Perception (Vision)*



Recap on Week 02

- Forward kinematics
  - Frame Assignment: D-H Convention
- Inverse kinematics
  - Solve joint variables by defining constraints
- Workspace
  - Reachable: Region where end-effector can be located
  - Dexterous: Region where end-effector can be located with all orientations
- Solvability; Number of Solutions; Jacobian; Singularity
- Velocity Kinematics

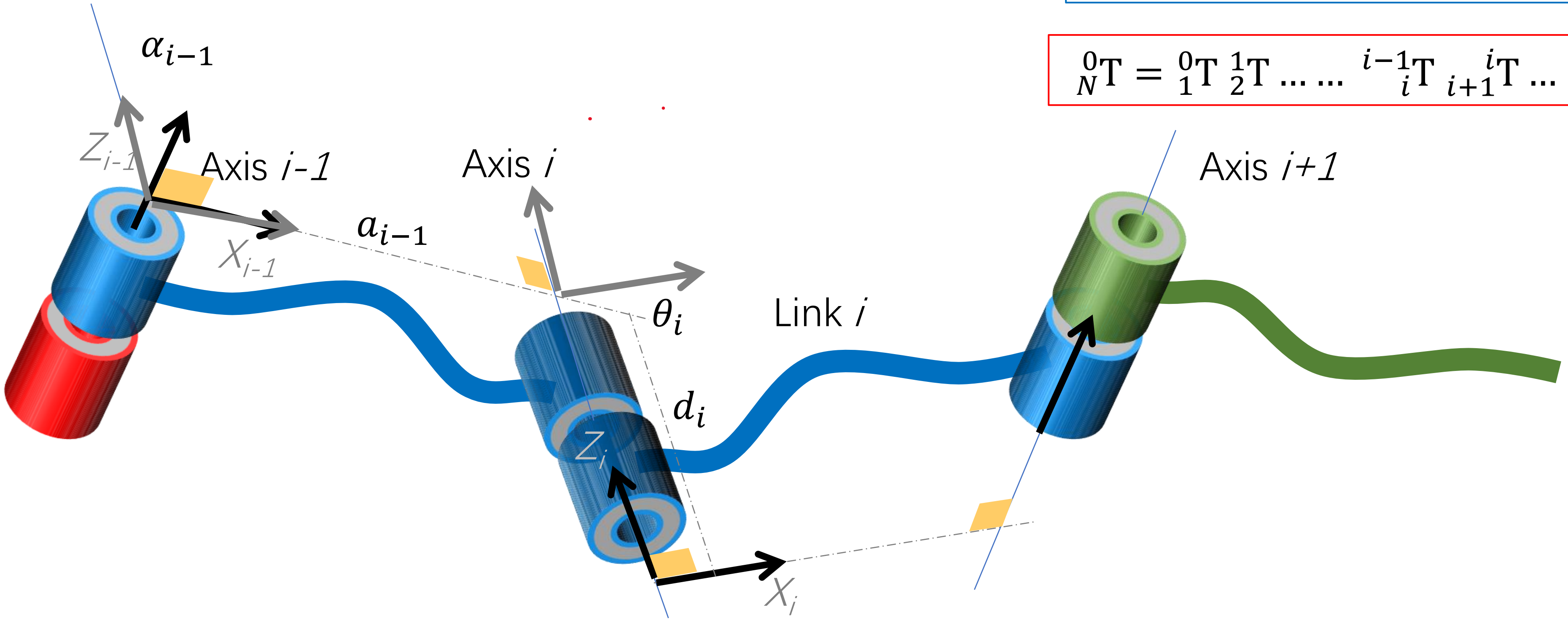


Kinematics Representation

1. Schematic of Serial Arm
2. Establish the DH parameters
3. Tabulate on the DH table
4. Obtain the relevant transformation matrix

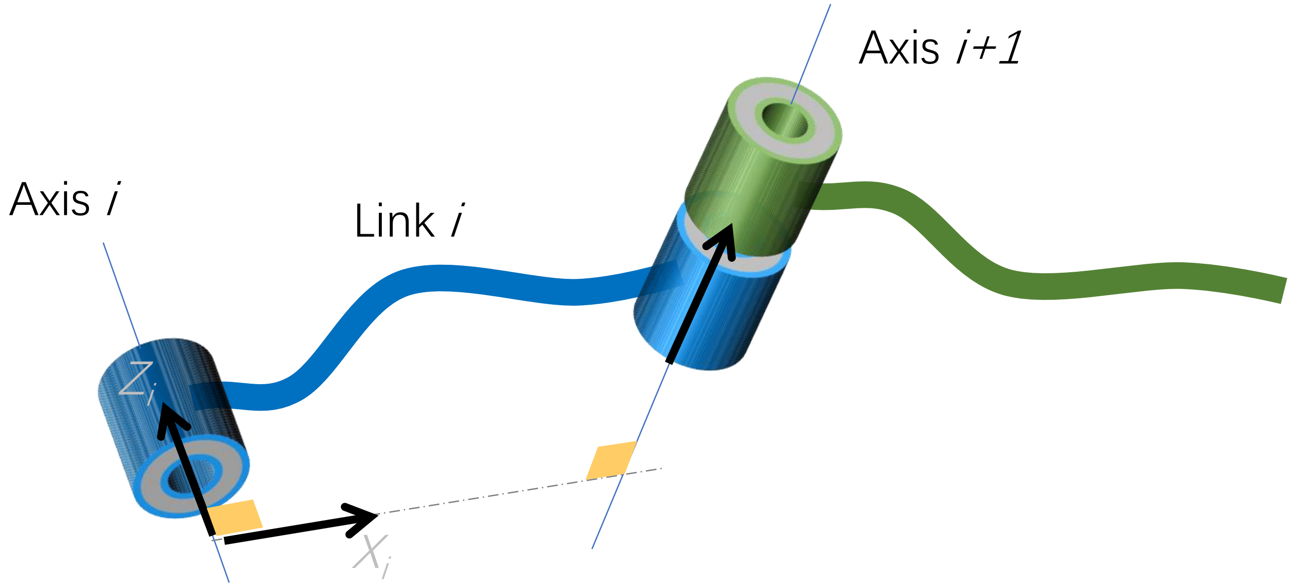
$$ {}^{i-1}_{i}T = R_x(\alpha_{i-1})D_x(a_{i-1})\ R_z(\theta_i)D_z(d_i) $$

$$ {}^{0}_{N}T = {}^{0}_{1}T\ {}^{1}_{2}T \dots\dots\ {}^{i-1}_{i}T\ {}^{i}_{i+1}T \dots\dots\ {}^{N-2}_{N-1}T\ {}^{N-1}_{N}T $$



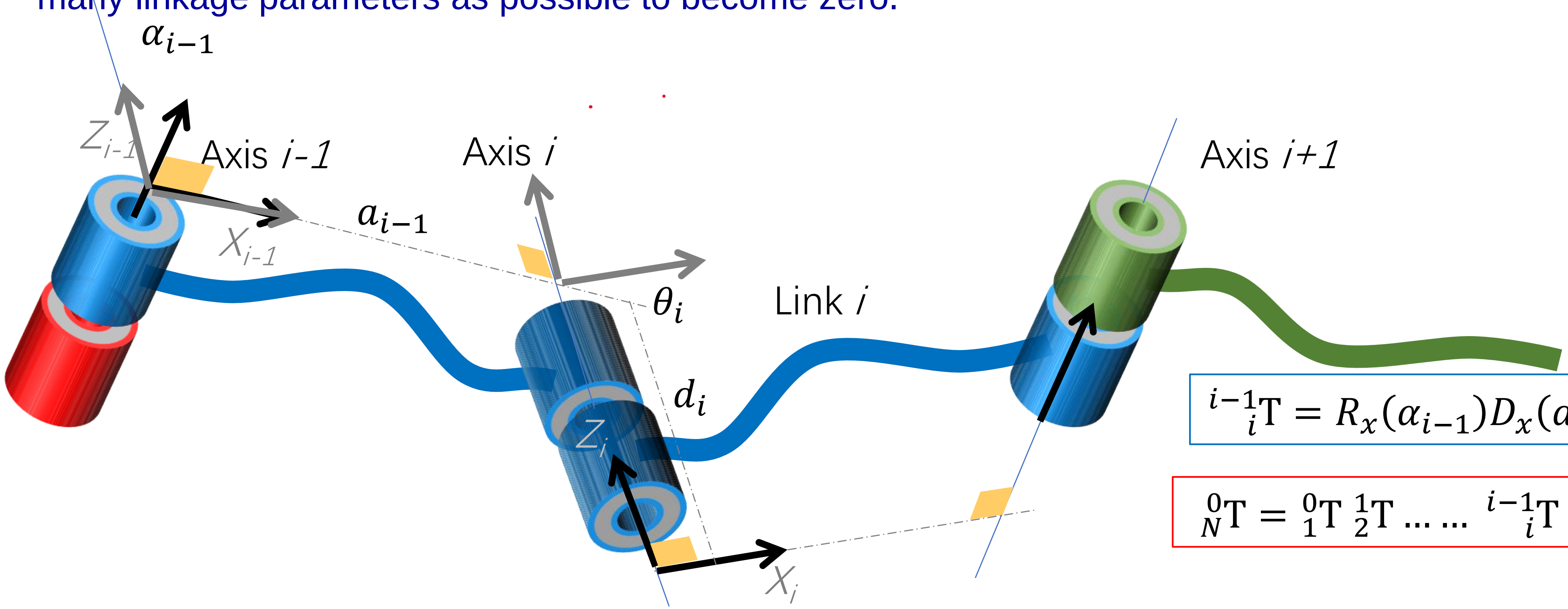
Recap: Summary of DH Frame Assignment

1. Identify the joint axes and attach infinite lines along them. For neighboring pair (*i* and *i*+1)
2. Identify the common perpendicular or point of intersection. At the point of intersection, or at the point where the common perpendicular meets i^{th} axis, assign the link-frame origin.
3. Assign the Z_i axis pointing along the i^{th} joint axis.
4. Assign the X_i axis pointing along the direction normal to the two neighboring Z-axes.
5. Assign the Y_i axis to complete a right-hand coordinate system.
6. Assign {0} to match {1}. For {N}, choose an origin location and X direction freely, but generally so as to cause as many linkage parameters as possible to become zero.



Recap: Summary of DH Frame Assignment

1. Identify the joint axes and attach infinite lines along them. For neighboring pair (*i* and *i*+1)
2. Identify the common perpendicular or point of intersection. At the point of intersection, or at the point where the common perpendicular meets i^{th} axis, assign the link-frame origin.
3. Assign the Z_i axis pointing along the i^{th} joint axis.
4. Assign the X_i axis pointing along the direction normal to the two neighboring Z-axes.
5. Assign the Y_i axis to complete a right-hand coordinate system.
6. Assign {0} to match {1}. For {N}, choose an origin location and X direction freely, but generally so as to cause as many linkage parameters as possible to become zero.

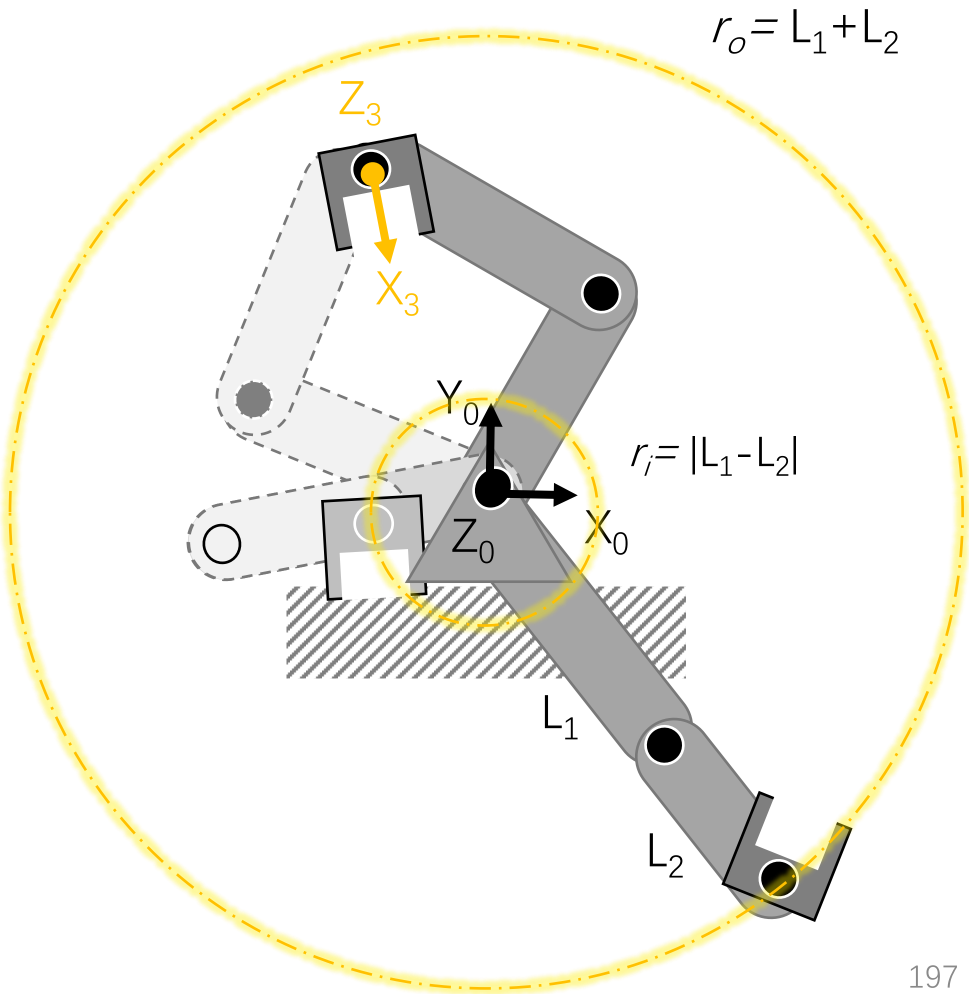


$${}^{i-1}_{i}T = R_x(\alpha_{i-1})D_x(a_{i-1})\, R_z(\theta_i)D_z(d_i)$$

$${}^{0}_{N}T = {}^{0}_{1}T\, {}^{1}_{2}T \dots\dots {}^{i-1}_{i}T\, {}^{i}_{i+1}T \dots\dots {}^{N-2}_{N-1}T\, {}^{N-1}_{N}T$$

Recap: Solvability

- Workspace
  - **Reachable**: Region where the end-effector can be located
  - **Dexterous**: Region where the end-effector can be located with all orientations
- Multiple solutions
  - For the same end-effector pose, there could be 2 possible solutions
- Approach to solutions:
  - Numerical
  - Closed-form



Jacobians (where we left off……)

- In general, seen as the mapping of velocities in X to Y

$$\dot{Y} = J(X)\dot{X}$$

- In robotics, used to relate joint velocities to cartesian velocities

$$^{0}v =^{0} J(\Theta)\,\dot{\Theta}$$

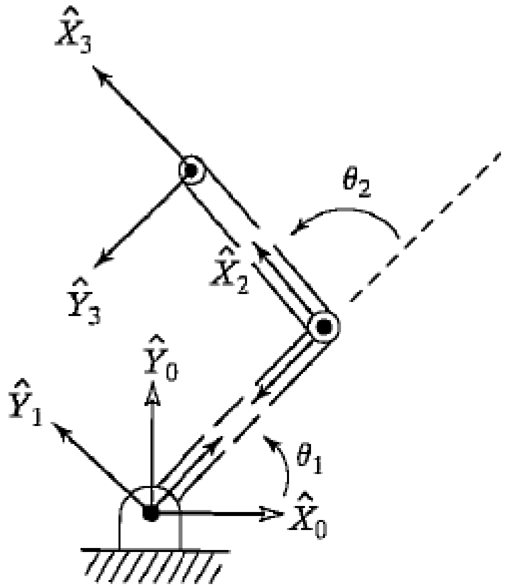
- In 3D space, a six-joint robot,
  - Jacobian $J(\Theta)$ is 6 x 6,
  - Joint velocity is $\dot{\Theta}$ is 6 x 1,
  - Cartesian velocity is $^{0}v = [\ ^{0}\dot{\mathrm{P}} \quad ^{0}\dot{\Theta}]^{\mathrm{T}}$ is 6 x 1

Jacobians (where we left off……)

- For an *N*-joint robot in 3D space,
  - Joint velocity is $\dot{\Theta}$ is *N* x 1,
  - Jacobian $J(\Theta)$ is 6 x *N*,
  - Cartesian velocity is $^0v = [\,^0\dot{\mathrm{P}} \quad ^0\dot{\Theta}]^{\mathrm{T}}$ is 6 x 1
    - Linear velocity stacked with rotational velocity
  - Cartesian velocity is $v_N = [J_1 \quad \dots J_i \dots \quad J_N] \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_N \end{bmatrix}$
    - Column J_i represents motion contribution of Joint i

Q3.5 Example on Jacobian

- Using the previous question Q3.4, obtain the 2 x 2 Jacobian that relates joint rates to end-effector velocity in both frame {3} and frame {0}.



Q3.5 Example on Jacobian

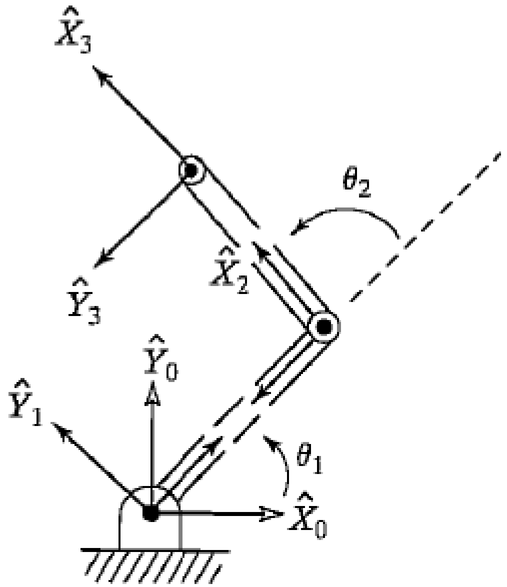
- Using the previous case problem, obtain the 2 x 2 Jacobian that relates joint rates to end-effector velocity in both frame {3} and frame {0}.

$$^{3}v_3 = \begin{bmatrix} l_1 s_2 \dot{\theta}_1 \\ l_1 c_2 \dot{\theta}_1 + l_2(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix} = \begin{bmatrix} l_1 s_2 & 0 \\ l_1 c_2 + l_2 & l_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$$^{0}v_3 = \begin{bmatrix} -l_1 s_1 \dot{\theta}_1 - l_2 s_{12}(\dot{\theta}_1 + \dot{\theta}_2) \\ l_1 c_1 \dot{\theta}_1 + l_2 c_{12}(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$$^{3}J(\Theta) = \begin{bmatrix} l_1 s_2 & 0 \\ l_1 c_2 + l_2 & l_2 \end{bmatrix}$$

$$^{0}J(\Theta) = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}$$



Changing Jacobian's frame of reference

- Given a Jacobian in frame {B},

$$\begin{bmatrix} {}^{B}v \\ {}^{B}\omega \end{bmatrix} = {}^{B}J(\Theta)\,\dot{\Theta}$$

- The velocity in {B} is described relative to {A} by the transformation

$$\begin{bmatrix} {}^{A}v \\ {}^{A}\omega \end{bmatrix} = \begin{bmatrix} {}^{A}_{B}R & 0 \\ 0 & {}^{A}_{B}R \end{bmatrix} \begin{bmatrix} {}^{B}v \\ {}^{B}\omega \end{bmatrix} = \begin{bmatrix} {}^{A}_{B}R & 0 \\ 0 & {}^{A}_{B}R \end{bmatrix} {}^{B}J(\Theta)\,\dot{\Theta}$$

- Hence,

$${}^{A}J(\Theta) = \begin{bmatrix} {}^{A}_{B}R & 0 \\ 0 & {}^{A}_{B}R \end{bmatrix} {}^{B}J(\Theta)$$

Singularities

$$v = J(\Theta)\,\dot{\Theta}$$
$$J^{-1}(\Theta)v = \dot{\Theta}$$

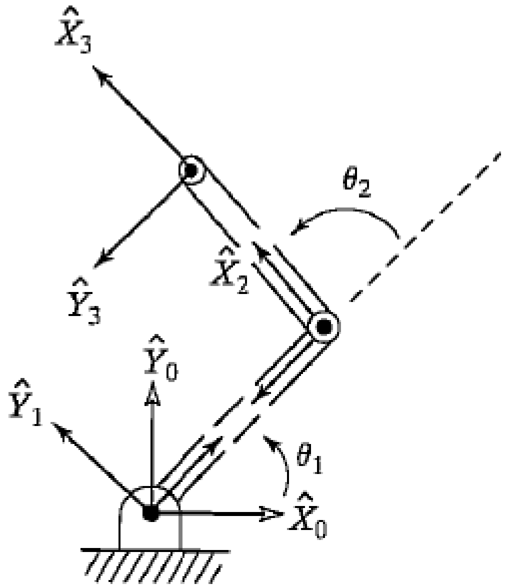
- This is important when a certain velocity vector of the end-effector is desired

- But what happens when Jacobian becomes singular (ie no inverse)?
  - Workspace-boundary singularities
  - Workspace-interior singularities
  - Inverse Jacobian blows up when at singular point

Q3.6 Example on Singularity

Continuing from the case problem, inverse of the Jacobian can be written as:

$$^{0}J^{-1}(\Theta) = \frac{1}{l_1 l_2 s_2}\begin{bmatrix} l_2 c_{12} & l_2 s_{12} \\ -l_1 c_1 - l_2 c_{12} & -l_1 s_1 - l_2 s_{12} \end{bmatrix}$$



For a desired velocity of $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ m/s,

$$\dot{\theta}_1 = \frac{c_{12}}{l_1 s_2}$$

$$\dot{\theta}_2 = -\frac{\dot{c}_1}{l_2 s_2} - \frac{c_{12}}{l_1 s_2}$$

As arm stretches out towards $\theta_2 = 0$, both joint rates go to infinity

Jacobian: Static Forces

Introduction to Robotics: Fundamentals

Robot Mechanics: Statics

- Concern with static forces in Manipulator

Jacobian: Static Forces

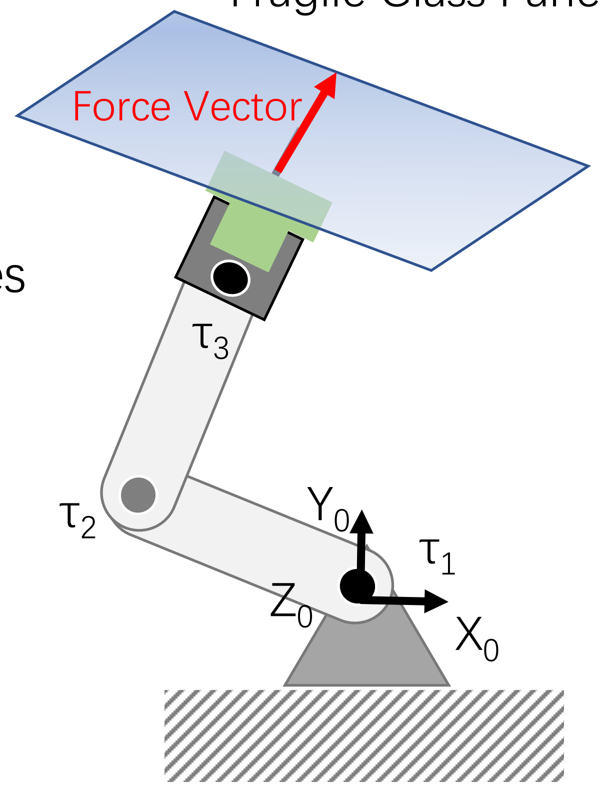
Introduction to Robotics: Fundamentals

Robot Mechanics: Statics

- Interested in knowing the static forces in manipulator
  - Compute forces at joints, given forces exerted by end-effector on environment
- Transform forces
  - Expressed in different frames
  - Applied at different points to create the same effect

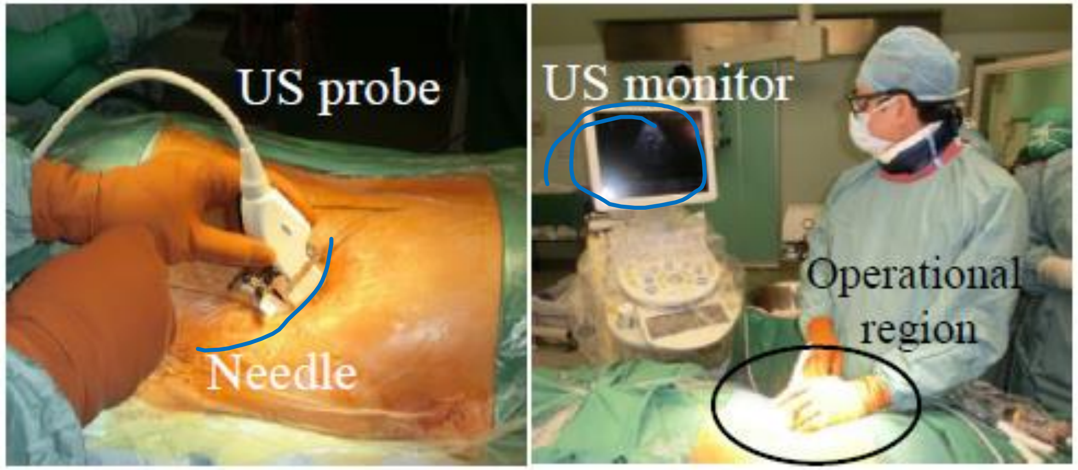
Why is understanding forces important?

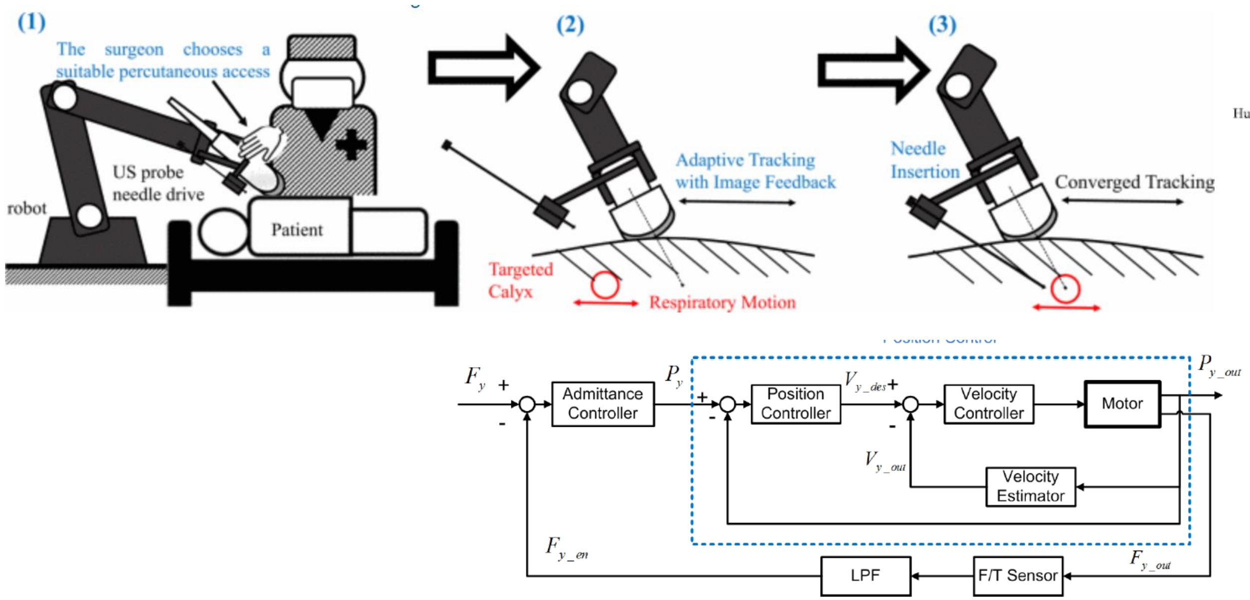
- Case examples
  - Cleaning the glass with the right amount of force so that it can clean but not break the glass
  - How do we know the relationship between the force vector and the joint torques?
    - a mapping between the cartesian and joint coordinates

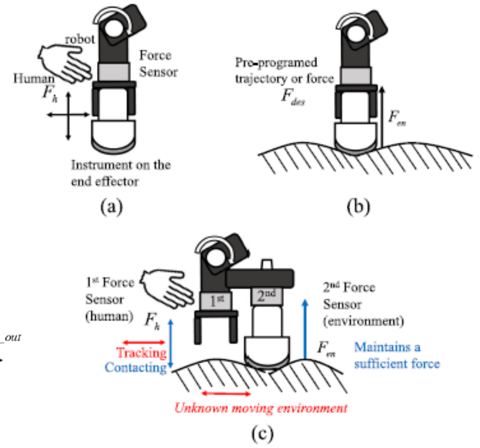


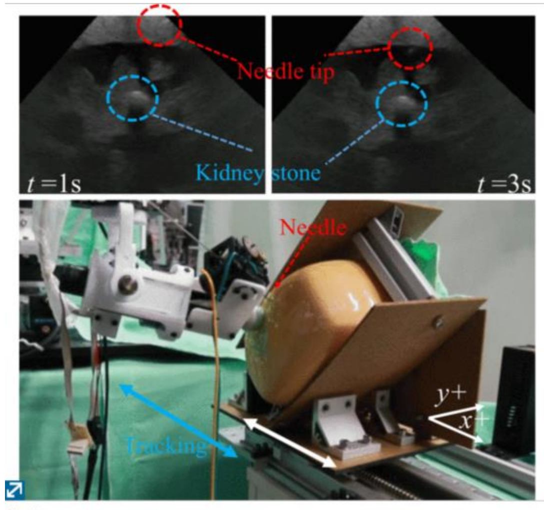
Why is understanding forces important?

- Case examples
  - Human uses force control during operation
  - Robot can be designed with this capability







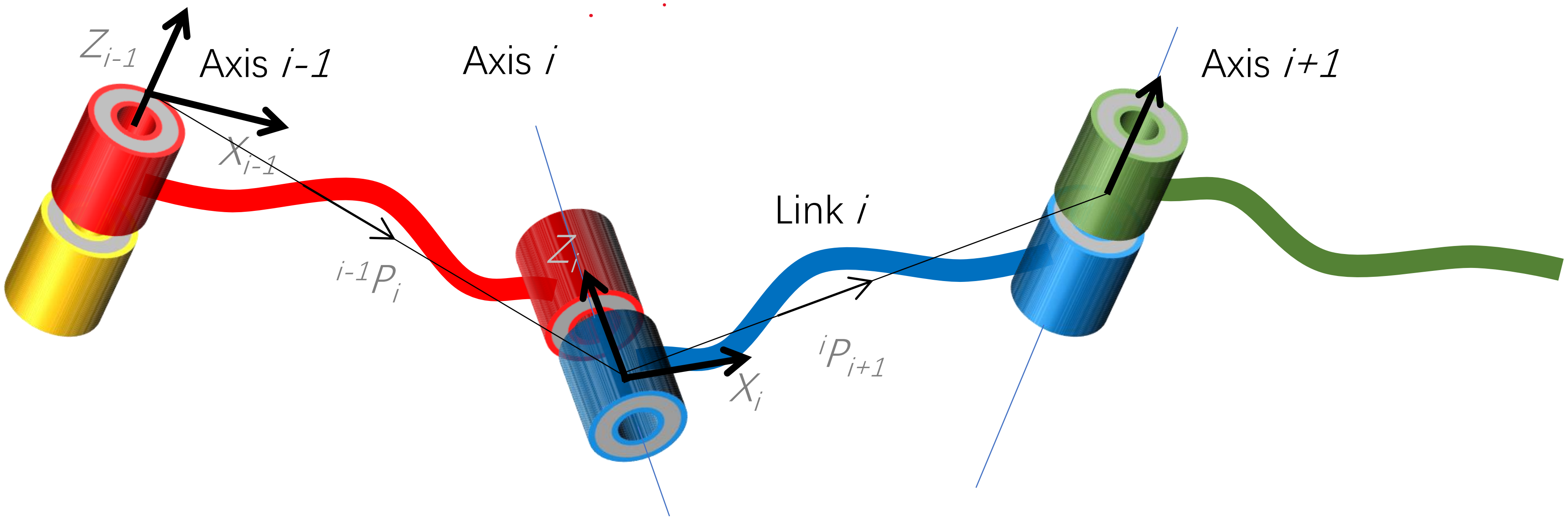


Static forces in manipulator

Let

f_i = force exerted on link *i* by link *i-1*

n_i = moment exerted on link *i*

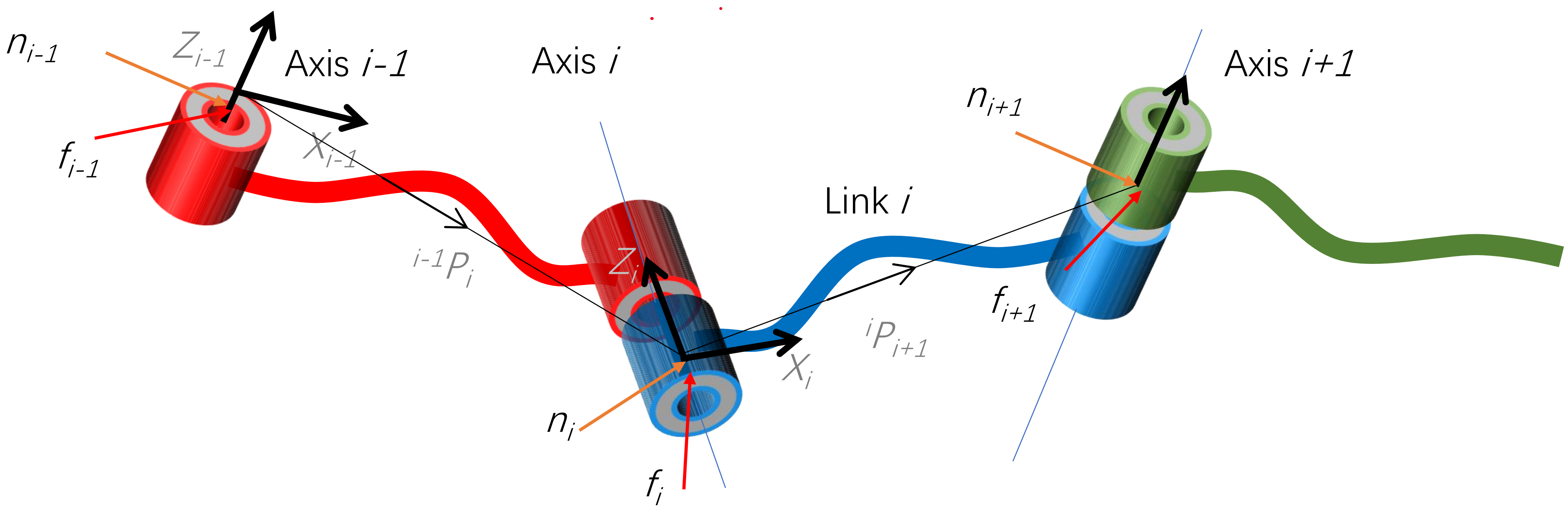


Static forces in manipulator

Let

f_i = force exerted on link i by link i-1

n_i = moment exerted on link i

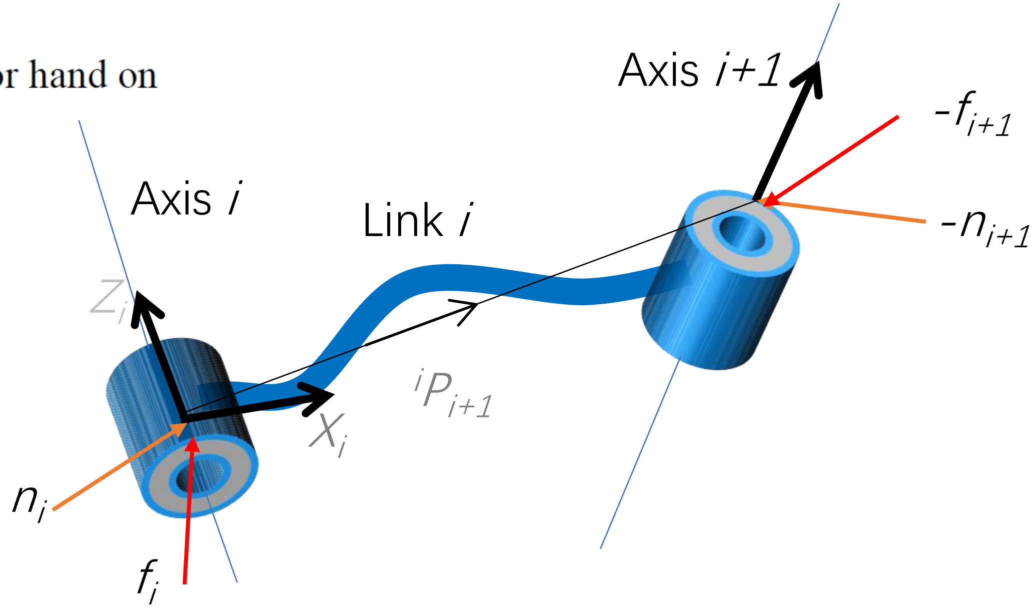


Static forces in manipulator

$$\begin{cases} \sum \mathbf{F} = 0 \quad \mathbf{f}_i - \mathbf{f}_{i+1} = 0 \\ \sum \text{Torques about origin of frame } i = 0 \\ \quad \mathbf{n}_i - \mathbf{n}_{i+1} + (\mathbf{p}_{i+1} - \mathbf{p}_i) \times (-\mathbf{f}_{i+1}) = 0 \end{cases}$$

If we start with a description of the force and moment applied by the hand, we can calculate the force and moment applied by each link working from the last link down to the base, link ϕ.

$\left.\begin{matrix} \mathbf{f}_{n+1} \\ \mathbf{n}_{n+1} \end{matrix}\right\}$ Force exerted by the manipulator hand on its environment.



Static forces in manipulator

Recursive Equations:

$$\left.\begin{aligned} \mathbf{f}_i &= \mathbf{f}_{i+1} \\ \mathbf{n}_i &= \mathbf{n}_{i+1} + (\mathbf{p}_{i+1} - \mathbf{p}_i) \times \mathbf{f}_{i+1} \end{aligned}\right\}$$

all vectors
expressed in
same frame
(e.g. base frame ϕ)

What forces are Needed at the Joints in order to
Balance the Reaction Forces & Moments acting in the link

$$\mathbf{T}_i = \begin{cases} \mathbf{n}_i^T \mathbf{z}_i & \text{for a rotational link i} \\ \mathbf{f}_i^T \mathbf{z}_i & \text{for a translational link i} \end{cases}$$

Jacobian of Force domains

- By Principle of virtual work, for static case:
- Amount of displacement to go to an infinitesimal
- Equate the **work done** in Cartesian terms with the work done in joint-space terms.
- In the multidimensional case, work is the **dot product** of a vector force/torque and a vector displacement. Thus, we have

6-by-1 Cartesian Force-Moment Vector

6-by-1 torque/force at joints

$$F \cdot \delta x = \tau . \delta\theta$$

6-by-1 virtual displacement in cartesian space

6-by-1 virtual joint displacement

Jacobian of Force domains

- By Principle of virtual work,

$$F \cdot \delta x = \tau . \delta\theta$$

$$F^{\mathrm{T}} \delta x = \tau^{\mathrm{T}} \delta\theta$$

Recall that, $\delta x = J\delta\theta$

$$F^{\mathrm{T}} (J\delta\theta) = \tau^{\mathrm{T}} \delta\theta$$

$$F^{\mathrm{T}} J = \tau^{\mathrm{T}}$$

6-by-1 Cartesian Force-Moment Vector

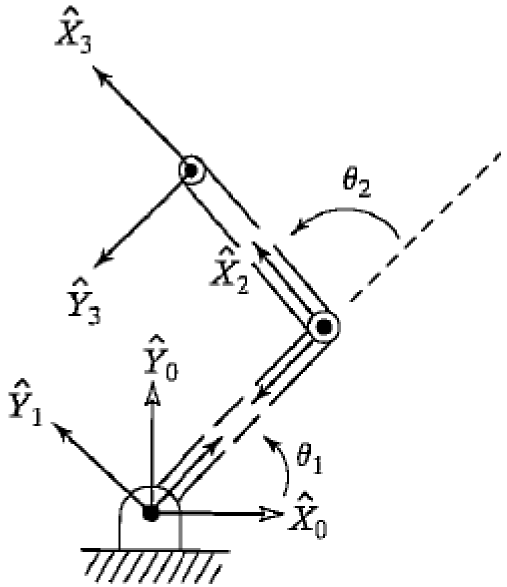
6-by-1 torque/force at joints $\rightarrow$ $\boxed{\tau = J^{\mathrm{T}} F}$

N-by-6 Jacobian Transposed

Q3.6 Example on Jacobian in Force Domain

Craig's Textbook Example 5.7

- For the 2-link manipulator example, find the required joint torque (i.e. actuator input) in order to apply a force vector ${}^3\mathbf{F}$ with its end-effector.



Q3.6 Example on Jacobian in Force Domain

Craig's Textbook Example 5.7

Deriving from the static equilibrium approach

$$^2f_2 = \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix},$$

$$^2n_2 = l_2\hat{X}_2 \times \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ l_2 f_y \end{bmatrix},$$

$$^1f_1 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} = \begin{bmatrix} c_2 f_x - s_2 f_y \\ s_2 f_x + c_2 f_y \\ 0 \end{bmatrix},$$

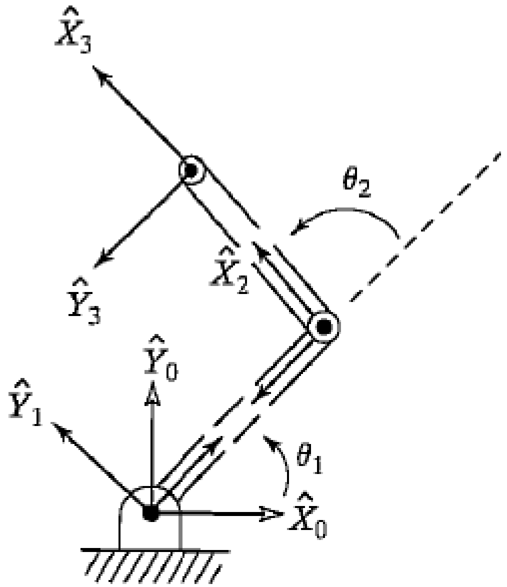
$$^1n_1 = \begin{bmatrix} 0 \\ 0 \\ l_2 f_y \end{bmatrix} + l_1\hat{X}_1 \times {}^1f_1 = \begin{bmatrix} 0 \\ 0 \\ l_1 s_2 f_x + l_1 c_2 f_y + l_2 f_y \end{bmatrix}.$$

Therefore, we have

$$\tau_1 = l_1 s_2 f_x + (l_2 + l_1 c_2) f_y,$$
$$\tau_2 = l_2 f_y.$$

This relationship can be written as a matrix operator:

$$\tau = \begin{bmatrix} l_1 s_2 & l_2 + l_1 c_2 \\ 0 & l_2 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix}.$$



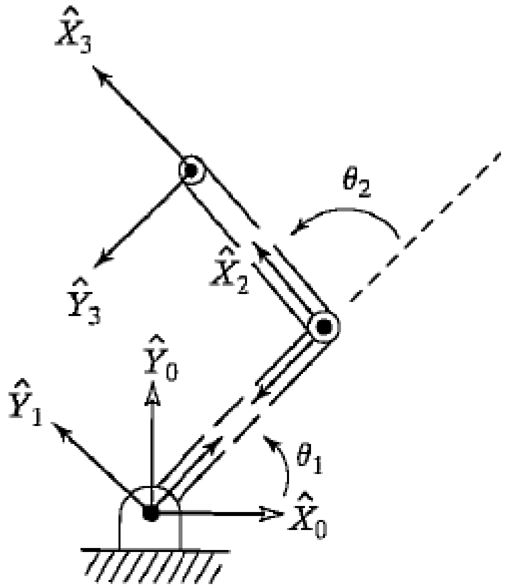
Q3.6 Example on Jacobian in Force Domain

Craig's Textbook Example 5.7

- In example Q3.5, we obtained the 2 x 2 Jacobian that relates joint rates to end-effector velocity in both frame {3}

$$ {}^{3}J(\Theta) = \begin{bmatrix} l_1 s_2 & 0 \\ l_1 c_2 + l_2 & l_2 \end{bmatrix} $$

It is no coincidence that this is the transpose of the matrix relating $\boldsymbol{\tau}$ and ${}^{3}\mathbf{F}$



Can you write down the matrix that can be pre-multiplied with ${}^{0}\mathbf{F}$ to obtain $\boldsymbol{\tau}$ based on the results in Q3.5?

Jacobian

- Note that a Cartesian space quantity can be converted into a joint space quantity without calculating inverse kinematic functions
- Recall that Jacobian maps Joint to Cartesian velocity coordinates as follows

$$v_N = [J_1 \quad \dots J_i \dots \quad J_N] \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_N \end{bmatrix}$$

- A useful expression for J_i can be written as

$$J_i = \begin{bmatrix} Z_{i-1} \\ 0 \end{bmatrix}$$ for a translational joint

$$J_i = \begin{bmatrix} Z_{i-1} \times (\mathrm{P}_N - \mathrm{P}_{i-1}) \\ Z_{i-1} \end{bmatrix}$$ for a rotational joint

Jacobian of Force domains

- When the Jacobian <u>loses full rank</u>, there are certain directions in which the end-effector cannot exert static forces (through joint actuation) as desired
- i.e. if <u>**J** is singular</u>, the equation is not valid
  - **F** could be increased or decreased in certain directions with no effect on the value calculated for τ
  - These directions are in the null-space of the Jacobian

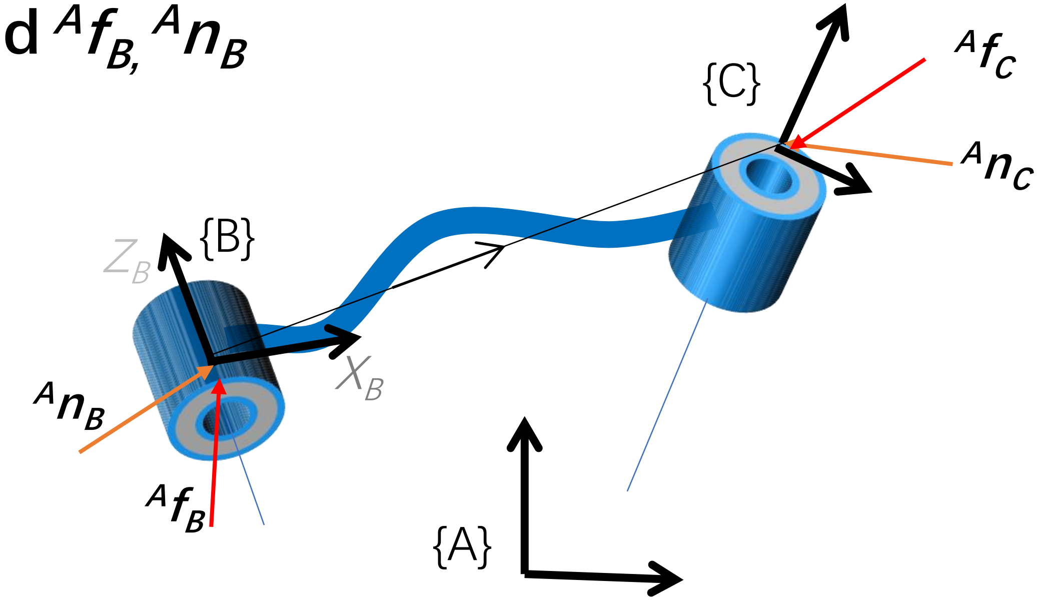
Static Force Transformation

Introduction to Robotics: Fundamentals

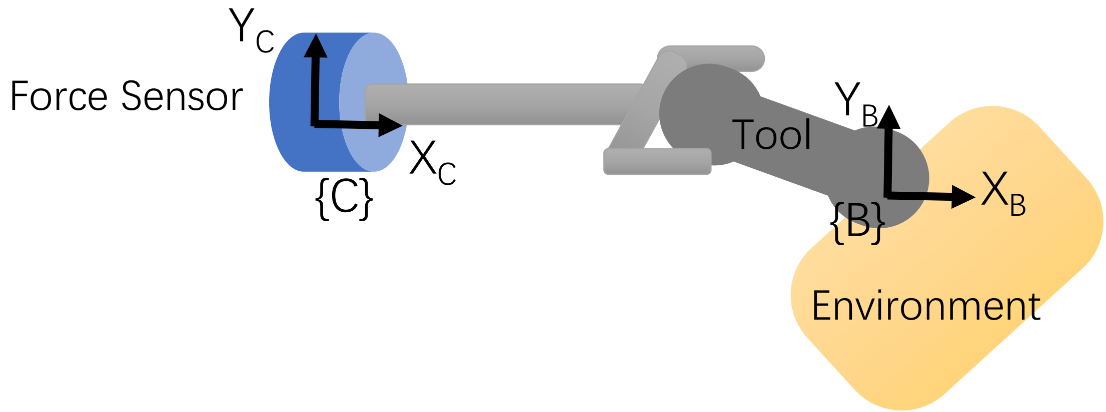
Cartesian Transformation of Static Force

Given ${}^{A}f_{C}$, ${}^{A}n_{C}$

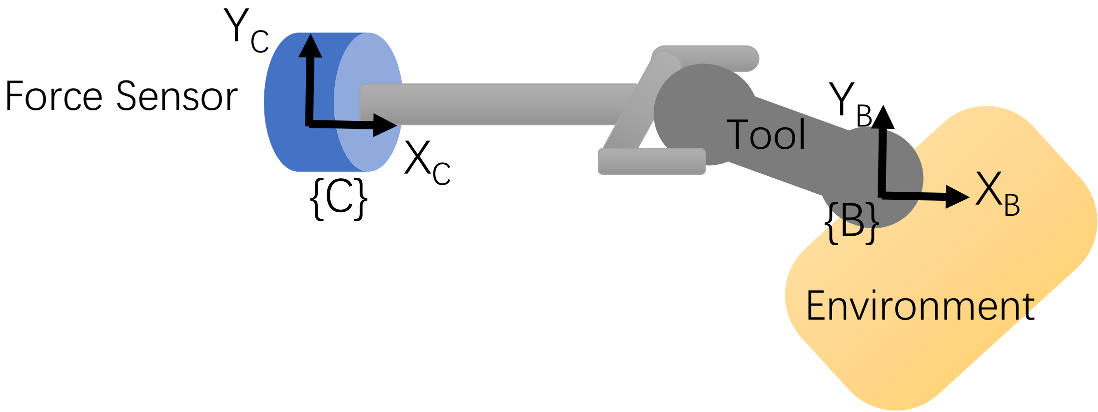
Find ${}^{A}f_{B}$, ${}^{A}n_{B}$



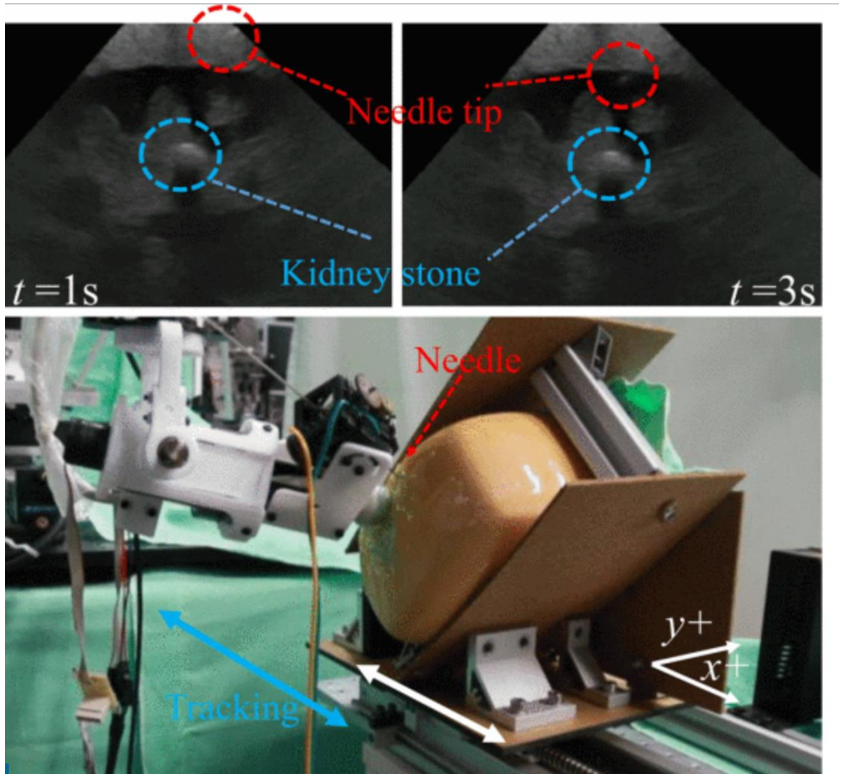
Cartesian Transformation of Static Force



Cartesian Transformation of Static Force



Example of a ultrasound transducer holding robot



Cartesian Transformation of Static Force

Velocity transformation

$$\begin{bmatrix} {}^B\upsilon_B \\ {}^B\omega_B \end{bmatrix} = \begin{bmatrix} {}^B_A R & -{}^B_A R\, {}^A P_{BORG}\times \\ 0 & {}^B_A R \end{bmatrix} \begin{bmatrix} {}^A\upsilon_A \\ {}^A\omega_A \end{bmatrix},$$

where the cross product is understood to be the matrix operator

$$P\times = \begin{bmatrix} 0 & -p_x & p_y \\ p_x & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix}.$$

Reversing the transformation,

$$\begin{bmatrix} {}^A\upsilon_A \\ {}^A\omega_A \end{bmatrix} = \begin{bmatrix} {}^A_B R & {}^A P_{BORG}\times {}^A_B R \\ 0 & {}^A_B R \end{bmatrix} \begin{bmatrix} {}^B\upsilon_B \\ {}^B\omega_B \end{bmatrix},$$

or

$${}^A\nu_A = {}^A_B T_\upsilon\, {}^B\nu_B.$$

Transpose of Jacobian

Force-moment transformation

$$\begin{bmatrix} {}^A F_A \\ {}^A N_A \end{bmatrix} = \begin{bmatrix} {}^A_B R & 0 \\ {}^A P_{BORG}\times {}^A_B R & {}^A_B R \end{bmatrix} \begin{bmatrix} {}^B F_B \\ {}^B N_B \end{bmatrix},$$

which may be written compactly as

$${}^A\mathcal{F}_A = {}^A_B T_f\, {}^B\mathcal{F}_B,$$

where T_f is used to denote a **force-moment transformation**.