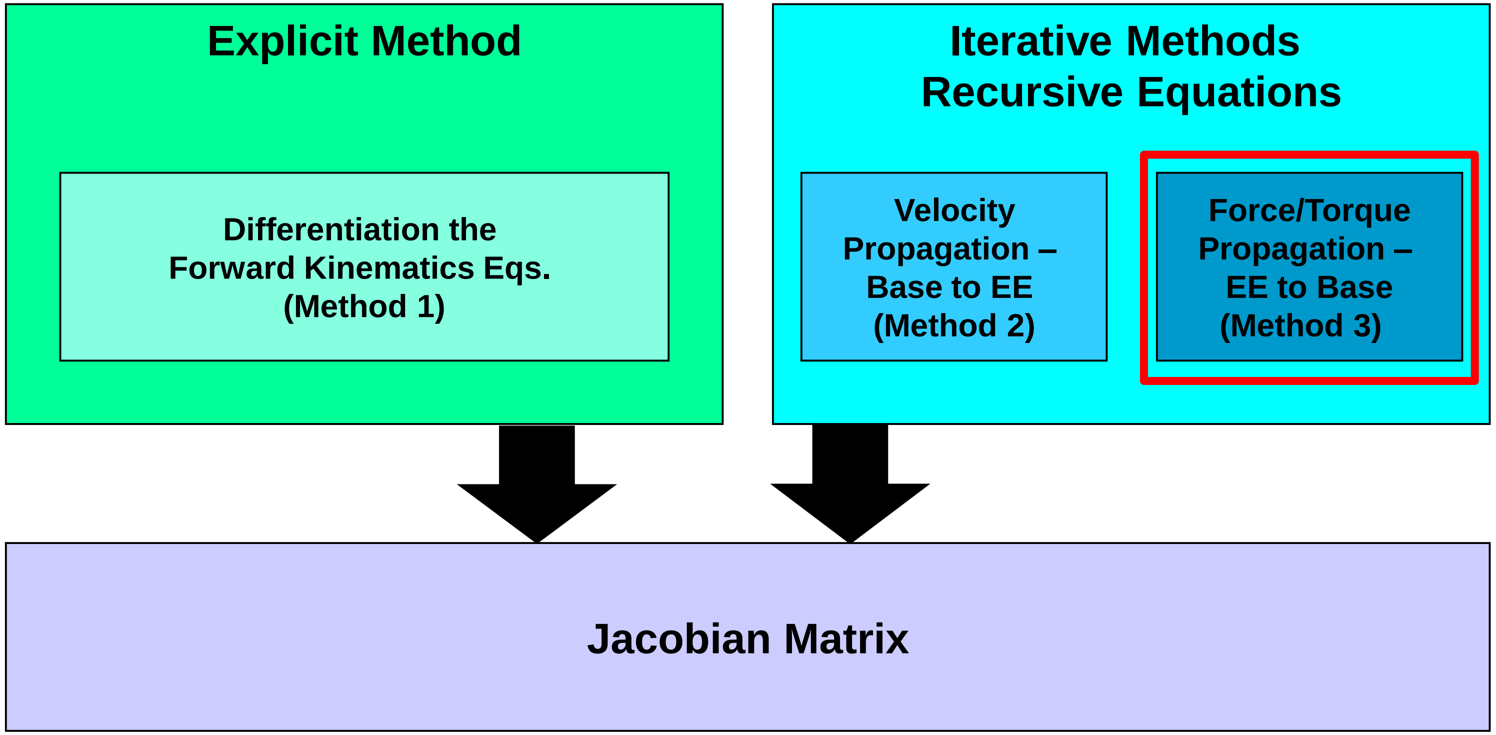
Jacobian Iterative Method - Force/Torque Propagation (Method No. 3)

Jacobian Matrix - Derivation Methods



Statics - Forces & Torques

Problem

Given: Typically the robot's end effector is applying forces and torques on an object in the environment or carrying an object (gravitational load).

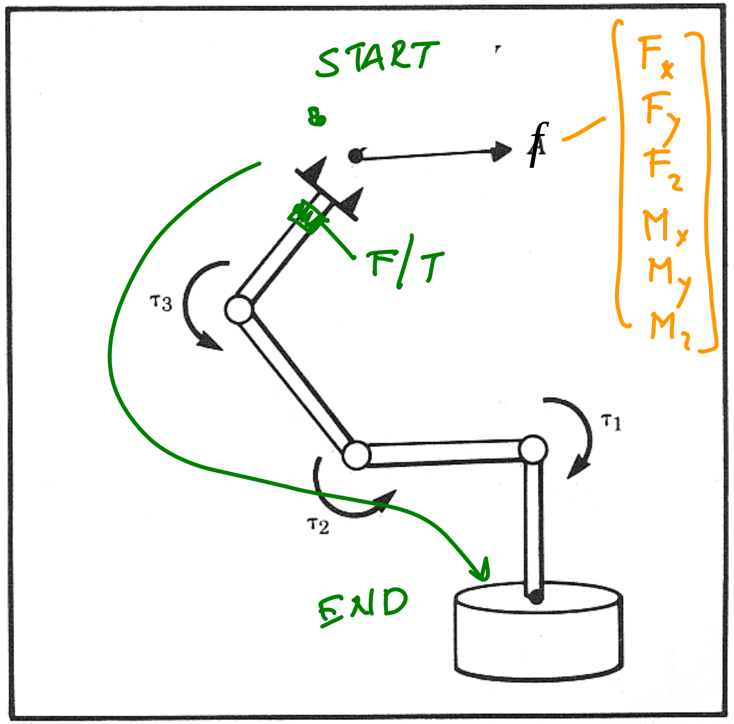
Compute: The joint torques which must be acting to keep the system in static equilibrium.

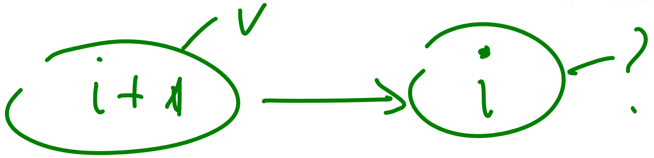
Solution

Jacobian - Mapping from the joint force/torques - τ to forces/torque in the Cartesian space applied on the end effector) - f .

$$\tau = \mathbf{J}^T f$$

Free Body Diagram - The chain like nature of a manipulator leads to decompose the chain into individual links and calculate how forces and moments propagate from one link to the next.





UCLA

Static Analysis Protocol - Free Body Diagram 1/

Step 1

Lock all the joints - Converting the manipulator (mechanism) to a structure

Step 2

Consider each link in the structure as a free body and write the force / moment equilibrium equations

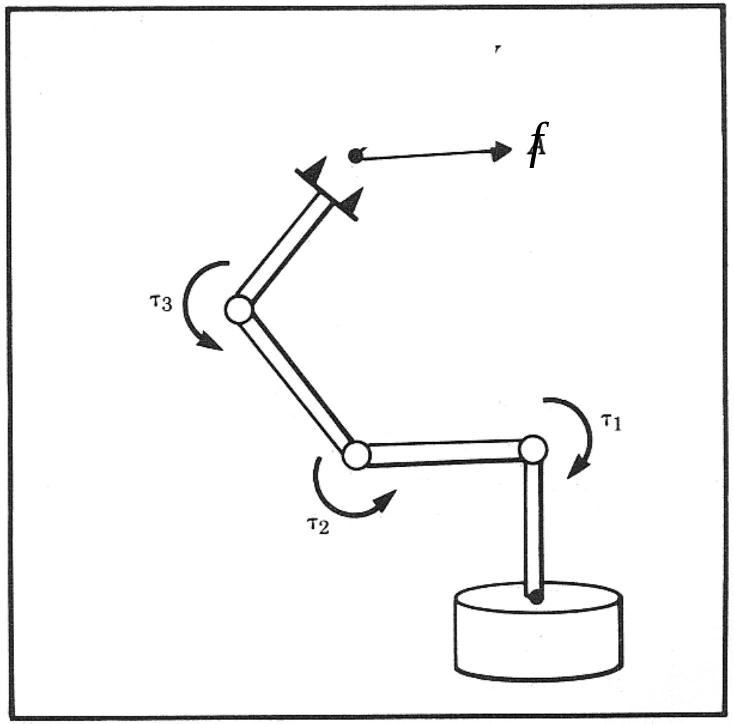
(3 Eqs.) $\sum F = 0$

(3 Eqs.) $\sum M = 0$

Step 3

Solve the equations - 6 Eq. for each link.

Apply backward solution starting from the last link (end effector) and end up at the first link (base)

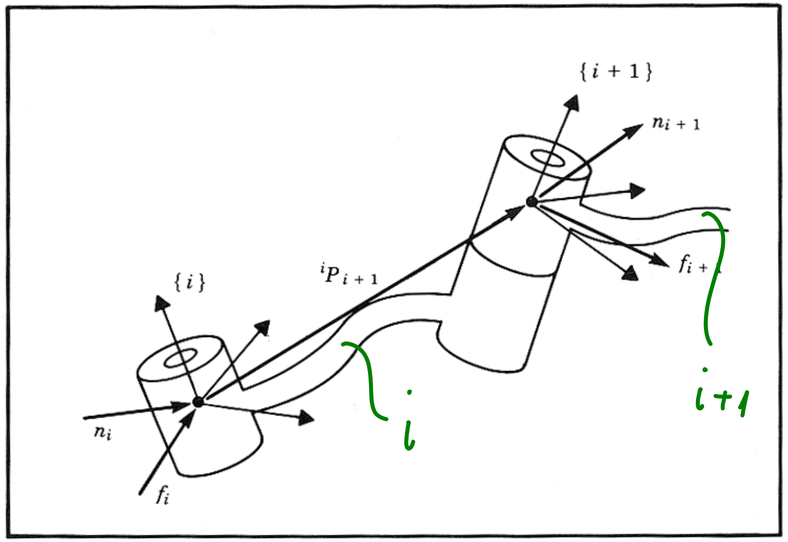


Static Analysis Protocol - Free Body Diagram 2/

- Special Symbols are defined for the force and torque exerted by the neighbor link

f_i - Force exerted on link $\boldsymbol{i}$ by link $\boldsymbol{i-1}$

n_i - Torque exerted on link $\boldsymbol{i}$ by link $\boldsymbol{i-1}$



Reference coordinate system {B}

Force $\boldsymbol{f}$ or torque $\boldsymbol{n}$

$${}^{B}C_{A}$$

${}^{B}f_{A}$

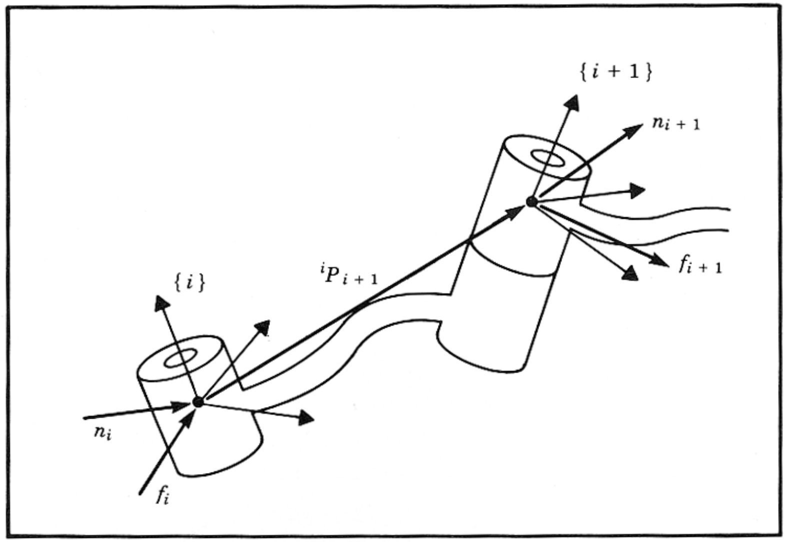
${}^{B}n_{A}$

Exerted on link A by link A-1

- For easy solution superscript index (A) should the same as the subscript (B)

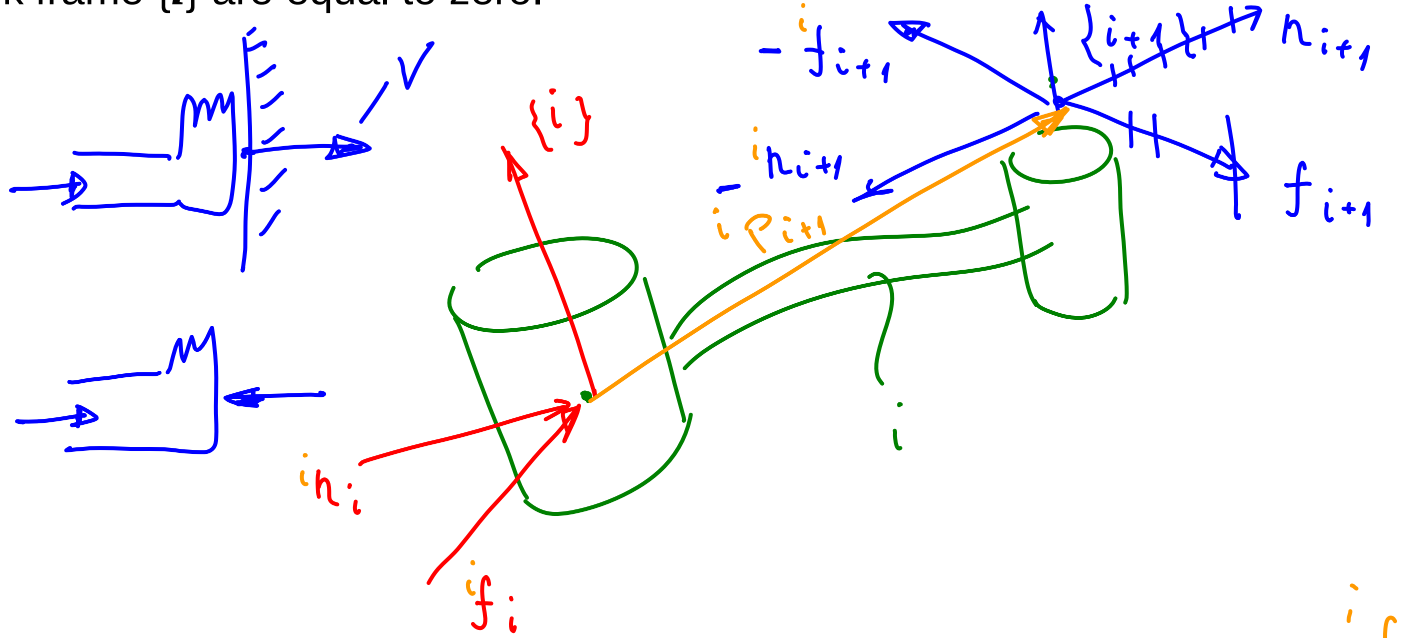
Static Analysis Protocol - Free Body Diagram 3/

- For serial manipulator in static equilibrium (joints locked), the sum the forces and torques acting on link i in the link frame $\{i\}$ are equal to zero.



Static Analysis Protocol - Free Body Diagram 3/

- For serial manipulator in static equilibrium (joints locked), the sum the forces and torques acting on link $\boldsymbol{i}$ in the link frame $\{\boldsymbol{i}\}$ are equal to zero.



$$\sum {}^iF = 0 \Rightarrow \sum {}^iF = {}^if_i - {}^if_{i+1} = 0 \longrightarrow \underbrace{{}^if_i = {}^if_{i+1}}$$

$$\sum {}^iM = 0 \Rightarrow \sum {}^iM = {}^in_i - {}^in_{i+1} - {}^iP_{i+1} \times {}^if_{i+1} = 0$$

ORIGIN of i

Static Analysis Protocol - Free Body Diagram 4/

- **Procedural Note**: The solution starts at the end effector and ends at the base
- Re-writing these equations in order such that the known forces (or torques) are on the right-hand side and the unknown forces (or torques) are on the left, we find

$$^{i}f_i = {}^{i}f_{i+1}$$

$$^{i}n_i = {}^{i}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_{i+1} = {}^{i}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_i$$

Static Analysis Protocol - Free Body Diagram 5/

- Changing the reference frame such that each force (and torque) is expressed upon their link's frame, we find the static force (and torque) propagation from link $\boldsymbol{i+1}$ to link $\boldsymbol{i}$

$$
{}^{i}f_i = {}^{i}f_{i+1} = {}^{i}_{i+1}R\;{}^{i+1}f_{i+1}
$$

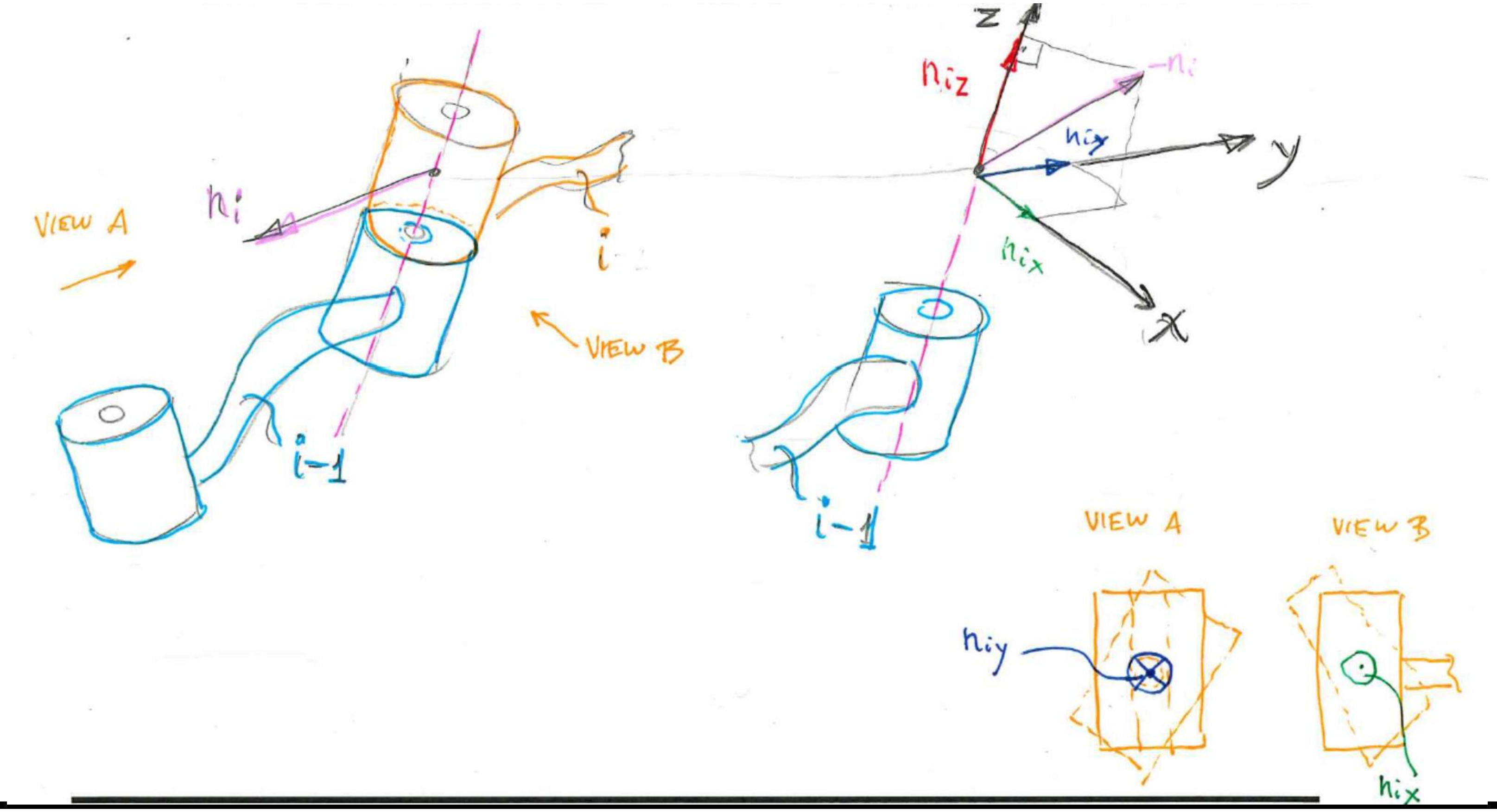
$$
{}^{i}n_i = {}^{i}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_{i+1} = {}^{i}_{i+1}R\;{}^{i+1}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_i
$$

$$
\boxed{\begin{aligned} {}^{i}f_i &= {}^{i}_{i+1}R\;{}^{i+1}f_{i+1} \\ {}^{i}n_i &= {}^{i}_{i+1}R\;{}^{i+1}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_i \end{aligned}}
$$

- These equations provide the static force (and torque) propagation from link to link. They allow us to start with the force and torque applied at the end effector, and calculate the force and torque at each joint all the way back to the robot base frame.

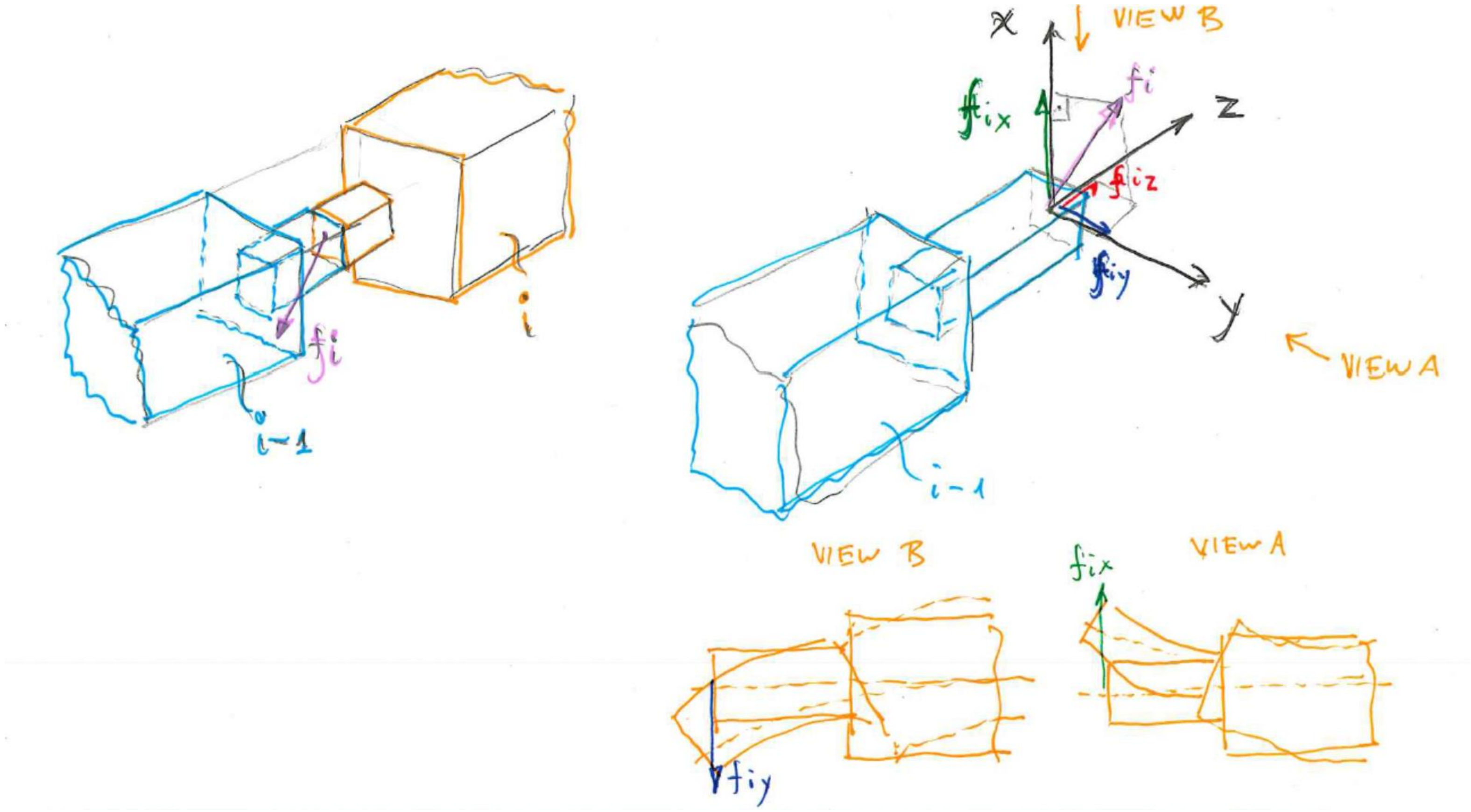
Static Analysis Protocol - Free Body Diagram 6/

- **Question:** What torques are needed at the joints in order to balance the reaction moments acting on the link (**Revolute Joint**).



Static Analysis Protocol - Free Body Diagram 7/

- **Question:** What forces are needed at the joints in order to balance the reaction forces acting on the link (**Prismatic Joint**).



Static Analysis Protocol - Free Body Diagram 8/

- **Answer:** All the components of the force and moment vectors are resisted by the structure of mechanism itself, except for the torque about the joint axis (revolute joint) or the force along the joint (prismatic joint).

- Therefore, to find the joint the torque or force required to maintain the static equilibrium, the dot product of the joint axis vector with the moment vector or force vector acting on the link is computed

Revolute Joint

$$\tau_i = {}^i n_i^T \hat{z}_i = [{}^i n_{ix} \quad {}^i n_{iy} \quad {}^i n_{iz}] \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Prismatic Joint

$$f_i = {}^i f_i^T \hat{z}_i = [{}^i f_{ix} \quad {}^i f_{iy} \quad {}^i f_{iz}] \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Example - 2R Robot - Static Analysis

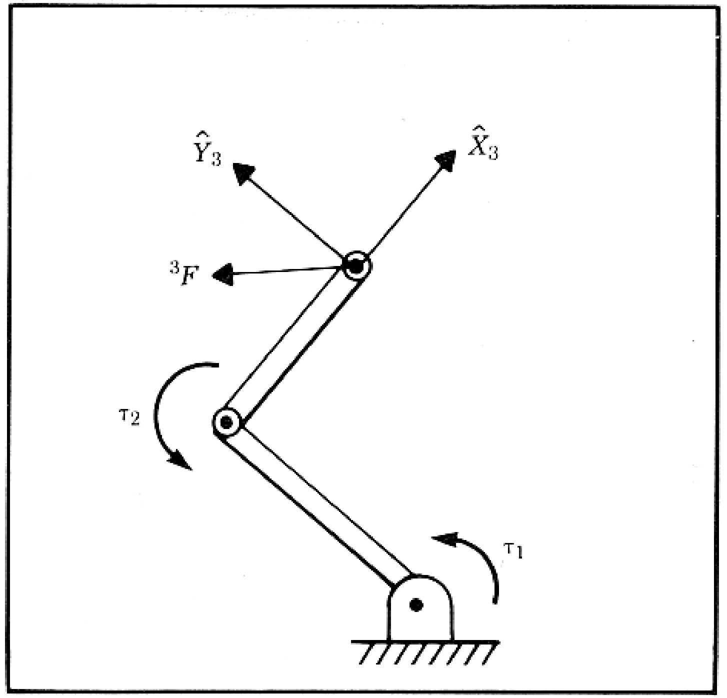
Problem

Given:

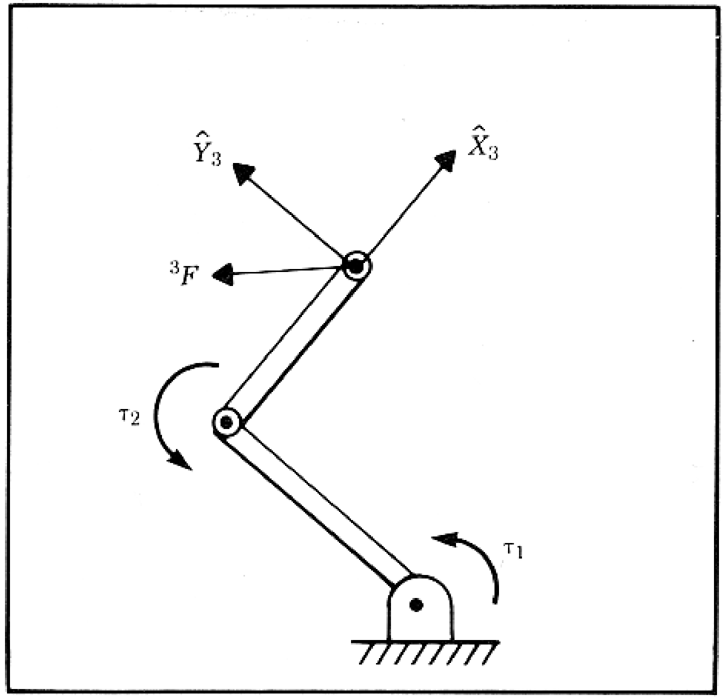
- 2R Robot
- A Force vector ${}^{3}f_3$ is applied **by** the end effector
- A torque vector ${}^{3}n_3 = 0$

Compute:

The required joint torque as a function of the robot configuration and the applied force



Example - 2R Robot - Static Analysis



Example - 2R Robot - Static Analysis

Solution

- Lock the revolute joints

- Apply the static equilibrium equations starting from the end effector and going toward the base

$$ {}^{i}f_{i} = {}^{i}_{i+1}R \; {}^{i+1}f_{i+1} $$

$$ {}^{i}n_{i} = {}^{i}_{i+1}R \; {}^{i+1}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_{i} $$

Example - 2R Robot - Static Analysis

- For i=2

$$ {}^{2}f_2 = {}^{2}_{3}R\ {}^{3}f_3 $$

$$ {}^{2}f_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} $$

$$ {}^{2}n_2 = {}^{2}_{3}R\ {}^{3}n_3 + {}^{2}P_3 \times {}^{2}f_2 $$

$$ {}^{2}n_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{X} & \hat{Y} & \hat{Z} \\ l_2 & 0 & 0 \\ f_x & f_y & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ l_2 f_y \end{bmatrix} $$

Example - 2R Robot - Static Analysis

- For i=1

$$^1f_1 = {}^1_2R\ {}^2f_2$$

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

$$^1n_1 = {}^1_2R\ {}^2n_2 + {}^1P_2 \times {}^1f_1$$

$$^1n_1 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ l_2 f_y \end{bmatrix} + \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} c_2 f_x - s_2 f_y \\ s_2 f_x + c_2 f_y \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ l_2 f_y \end{bmatrix} + \begin{bmatrix} \hat{X} & \hat{Y} & \hat{Z} \\ l_1 & 0 & 0 \\ c_2 f_x - s_2 f_y & s_2 f_x + c_2 f_y & 0 \end{bmatrix} =$$

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

Example - 2R Robot - Static Analysis

$$
{}^{1}n_{1} = \begin{bmatrix} 0 \\ 0 \\ l_1 s_2 f_x + l_1 c_2 f_y + l_2 f_y \end{bmatrix} \qquad {}^{2}n_{2} = \begin{bmatrix} 0 \\ 0 \\ l_2 f_y \end{bmatrix}
$$

$$
\tau_i = {}^{i}n_i^T \hat{z}_i = \begin{bmatrix} {}^{i}n_{i\,x} & {}^{i}n_{i\,y} & {}^{i}n_{i\,z} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
$$

$$
\tau_1 = l_1 s_2 f_x + l_1 c_2 f_y + l_2 f_y \qquad \tau_2 = l_2 f_y
$$

Example - 2R Robot - Static Analysis

- Re-writing the equations in a matrix form

$$\tau = \begin{bmatrix} l_1 s_2 & l_1 c_2 + l_2 \\ 0 & l_2 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix} = [{}^N J]^T f$$

Jacobian Methods of Derivation & the Corresponding Reference Frame – Summary

Method	Jacobian Matrix Reference Frame	Transformation to Base Frame (Frame 0)
Explicit (Diff. the Forward Kinematic Eq.)	${}^{0}J_{N}$	None
Iterative Velocity Eq.	${}^{N}J_{N}$	Transform Method 1: ${}^{0}v_{N}={}^{0}_{N}R\,{}^{N}v_{N}$ ${}^{0}\omega_{N}={}^{0}_{N}R\,{}^{N}\omega_{N}$ Transform Method 2: ${}^{0}J_{N}(\theta)=\begin{bmatrix} {}^{0}_{N}R & 0 \\ 0 & {}^{0}_{N}R \end{bmatrix}{}^{N}J_{N}(\theta)$
Iterative Force Eq.	${}^{N}J_{N}^{T}$	Transpose ${}^{N}J_{N}=[{}^{N}J_{N}^{T}]^{T}$ Transform ${}^{0}J_{N}(\theta)=\begin{bmatrix} {}^{0}_{N}R & 0 \\ 0 & {}^{0}_{N}R \end{bmatrix}{}^{N}J_{N}(\theta)$