Week 5 – Jacobians: Velocities and Static Forces

Advanced Robotic Systems – MANU2453

Dr Ehsan Asadi, School of Engineering
RMIT University, Victoria, Australia
Email: ehsan.asadi@rmit.edu.au

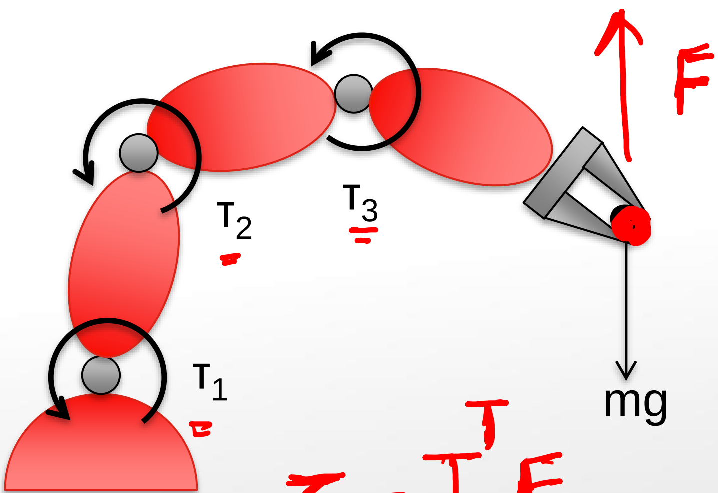
RMIT UNIVERSITY

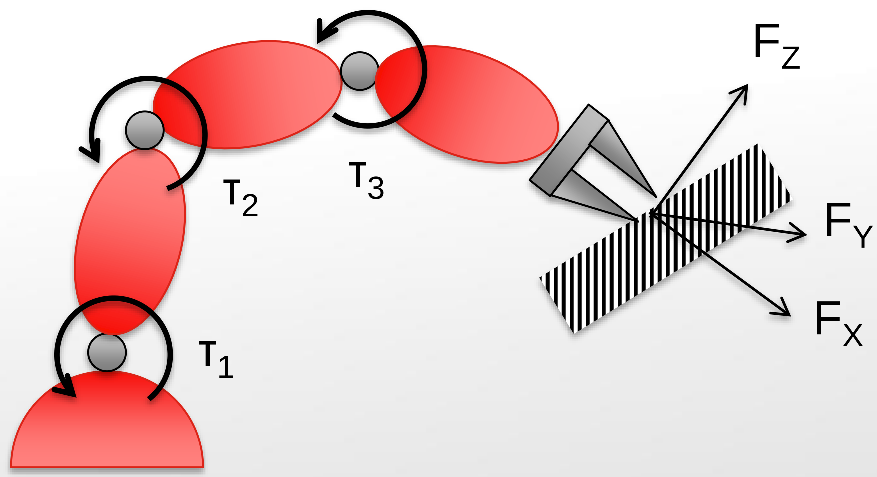
Content

- Introduction - Jacobian
- Method 1 - Direct differentiation (for Linear Jacobian)
- Method 2 - Velocity Propagation from Link to Link
- Method 3 - Explicit Form (for your study, not included in exam)
- Static Forces in Manipulators
- Singularities

Static Forces in Manipulator

- The question we would like to answer in this section is as follows:
  - The robot is holding an object with mass m (left), or
  - The robot is pushing the environment with force F (right).
  - What would be the joint torques needed to keep the system in static https://rmit.instructure.com/courses/51269/external_tools/23547?



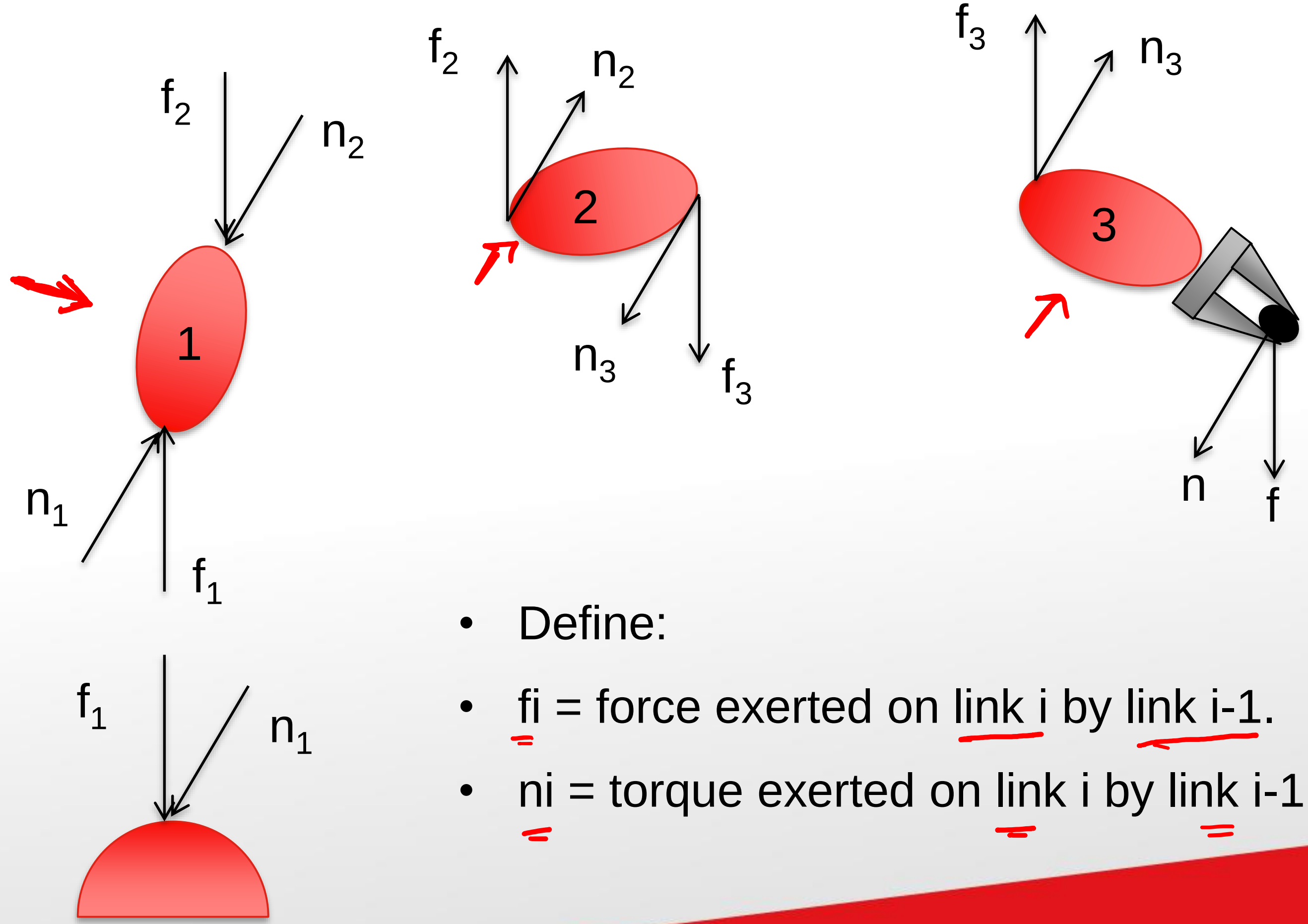


$$\tau = J^T F$$

$$\dot{X} = J\dot{\theta}$$

Static Forces in Manipulator

- This can be solved by separating each link, and find a force-moment balance relationship in terms of the link frames.

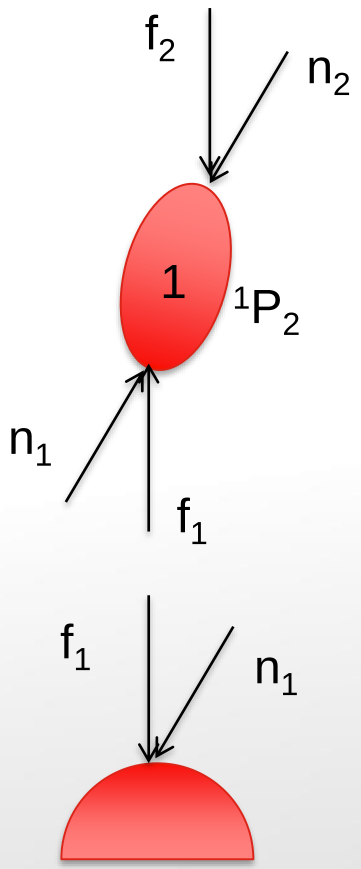


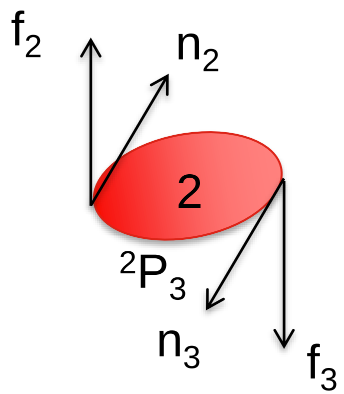
- Define:
- fi = force exerted on link i by link i-1.
- ni = torque exerted on link i by link i-1.

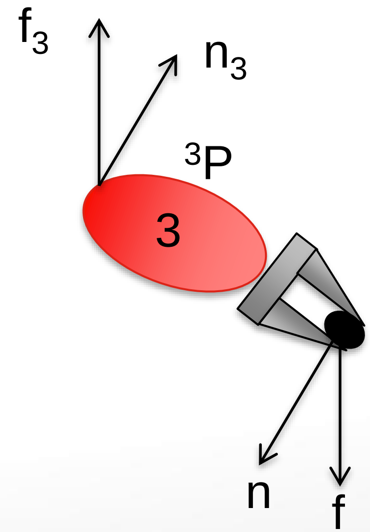
Static Forces in Manipulator

- For static equilibrium: $\sum f = 0 \ \& \ \sum n = 0$

Here: about frame origin







- Therefore:

$$^1f_1 - {}^1f_2 = 0; \quad {}^2f_2 - {}^2f_3 = 0; \quad {}^3f_3 - {}^3f = 0$$

$$^1n_1 - {}^1n_2 - {}^1P_2 \times {}^1f_2 = 0 \qquad {}^2n_2 - {}^2n_3 - {}^2P_3 \times {}^2f_3 = 0$$

$$^3n_3 - {}^3n - {}^3P \times {}^3f = 0$$

Static Forces in Manipulator

- We see that: $\begin{cases} {}^{i}f_i = {}^{i}f_{i+1} \\ {}^{i}n_i = {}^{i}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_{i+1} \end{cases}$ length or $\begin{matrix} {}^{i}f_i = {}^{i}_{i+1}R \cdot {}^{i+1}f_{i+1} \\ {}^{i}n_i = {}^{i}_{i+1}R \cdot {}^{i+1}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_i \end{matrix}$ same
- The equations on the right use only forces and moments described within their own link frames.
- Hence we have the algorithm:

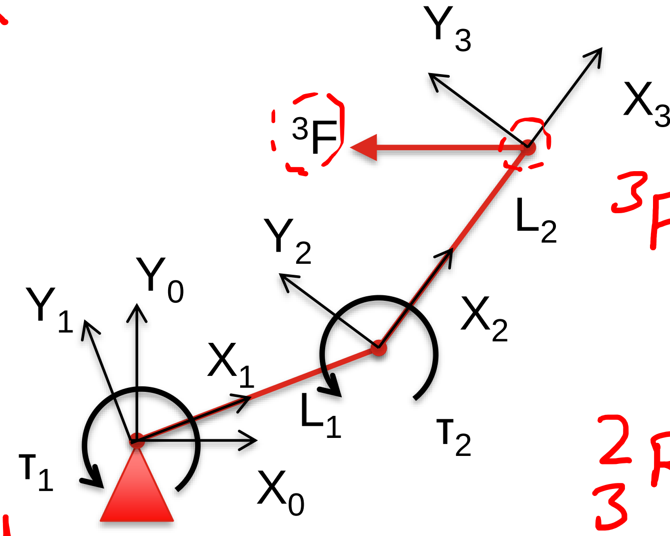
1 • Start from last link. n

2 • Calculate iteratively, down to the base: $\begin{cases} {}^{i}f_i = {}^{i}_{i+1}R \cdot {}^{i+1}f_{i+1} \\ {}^{i}n_i = {}^{i}_{i+1}R \cdot {}^{i+1}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_i \end{cases}$

3 • The joint torques required to maintain the static equilibrium are then calculated as dot-product of joint-axis vector and the moment vector acting on the link:
  - R Revolute: $\tau_i = {}^{i}n_i^T \cdot {}^{i}\hat{Z}_i$ P Prismatic: $\tau_i = {}^{i}f_i^T \cdot {}^{i}\hat{Z}_i$

Example

2-Link



$${}^3F = \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix}$$

- A force F acts horizontally (wrt {0}) on the origin of {3}, of the 2-link robot.
- What are the joint torques needed to hold the robot in equilibrium?
- We apply the recursive algorithm:

$${}^2_3R = I$$

1
  - Start from: $n = 2$

$$\begin{bmatrix} L_2 \\ 0 \\ 0 \end{bmatrix} \times$$

$${}^2f_2 = \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} \qquad {}^2n_2 = L_2\hat{X}_2 \times \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ L_2 f_y \end{bmatrix}$$

- where f_x and f_y are the x and y components of ${}^3F$ in {3}, and they are the same for {2}.

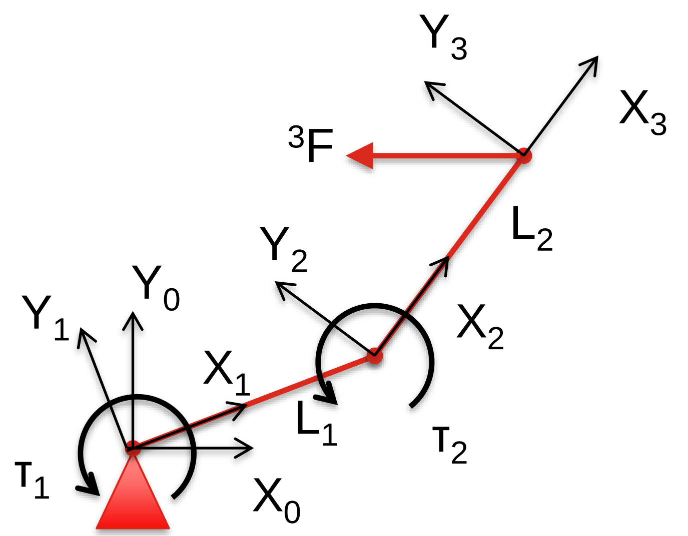
$$\begin{vmatrix} i & j & k \\ L_2 & 0 & 0 \\ f_x & f_y & 0 \end{vmatrix}$$

$0 \quad 0$

Example

Step 2

- Next:



$1f_1$

$${}^1f_1 = {}^1_2R \cdot {}^2f_2 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} = \begin{bmatrix} c_2 f_x - s_2 f_y \\ s_2 f_x + c_2 f_y \\ 0 \end{bmatrix}$$

Last link

$$\begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix}$$

a

b

$${}^1n_1 = {}^1_2R \cdot {}^2n_2 + {}^1P_2 \times {}^1f_1 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ L_2 f_y \end{bmatrix} + \underline{L_1 \hat{X}_1} \times \begin{bmatrix} c_2 f_x - s_2 f_y \\ s_2 f_x - c_2 f_y \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ L_2 f_y \end{bmatrix} + \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} c_2 f_x - s_2 f_y \\ s_2 f_x + c_2 f_y \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ L_2 f_y + L_1 s_2 f_x + L_1 c_2 f_y \end{bmatrix}$$

$$\begin{vmatrix} i & j & K \\ L_1 & 0 & 0 \\ a & b & 0 \end{vmatrix}$$

Previous Steps → $^1n_1$ & $^2n_2$ -

Example

$\begin{bmatrix}a_1\\a_2\\\vdots\end{bmatrix}\begin{bmatrix}b_1\\b_2\\\vdots\end{bmatrix}$ $a\cdot b = a^T b$

Step 3

- Finally, we calculate the torques:

J1

$$\tau_1 = {}^1n_1^T \cdot {}^1\hat{Z}_1$$

$^1n_1^T$

$$= \begin{bmatrix}0 & 0 & L_2 f_y + L_1 s_2 f_x + L_1 c_2 f_y\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix}$$

1x3 3x1

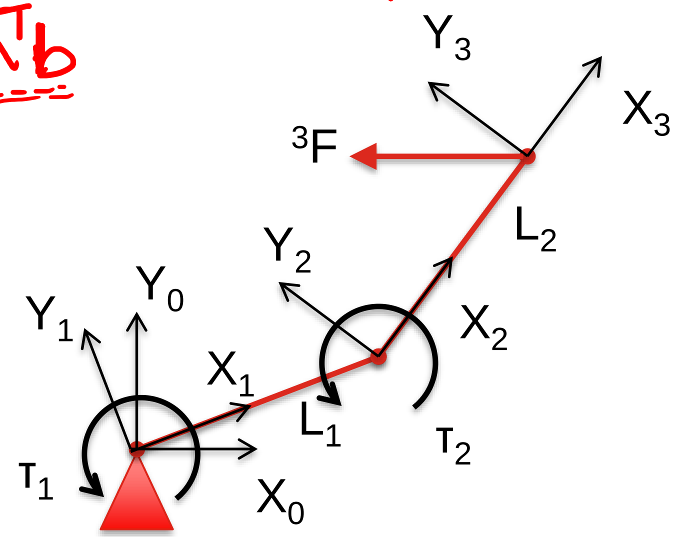
$$\tau_1 = L_1 s_2 f_x + \left(L_2 + L_1 c_2\right) f_y$$

J2

$$\tau_2 = {}^2n_2^T \cdot {}^2\hat{Z}_2$$

$$= \begin{bmatrix}0 & 0 & L_2 f_y\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix}$$

$$\tau_2 = L_2 f_y$$



$^3\tau = {}^3J^T \cdot {}^3F$

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = \begin{bmatrix}L_1 s_2 & L_2 + L_1 c_2\\ 0 & L_2\end{bmatrix}\begin{bmatrix}f_x\\ f_y\end{bmatrix}$$

Remember that this is in frame {3}

This is exactly $^3J^T$ → we had this expression before:

^3J

$$^3\upsilon_3 = \begin{bmatrix}L_1 s_2 \dot{\theta}_1\\ L_1 c_2 \dot{\theta}_1 + L_2\left(\dot{\theta}_1 + \dot{\theta}_2\right)\end{bmatrix} = \begin{bmatrix}L_1 s_2 & 0\\ L_2 + L_1 c_2 & L_2\end{bmatrix}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}$$

$^0J \to ?$

Example

^0_3R

- Therefore: $\tau = {}^3J^T \cdot {}^3F$
- We can also express everything in frame {0}:

$$\tau = {}^0J^T \cdot {}^0F$$

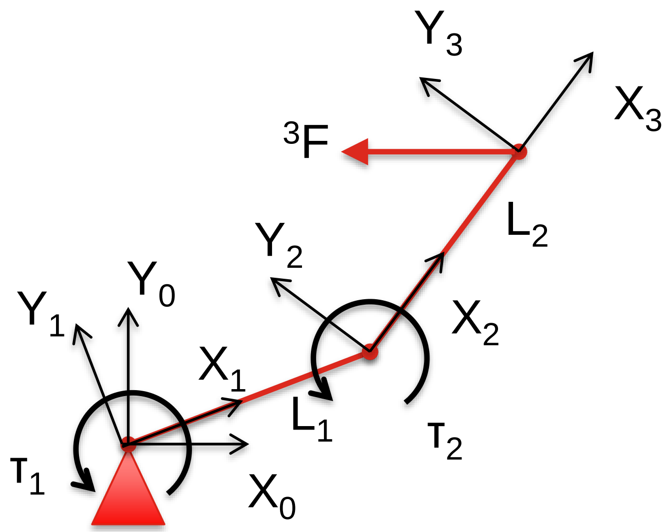
$$= \left({}^0_3R\,{}^3J\right)^T \cdot {}^0F$$

$${}^0\tau = \left(\begin{bmatrix} c_{12} & -s_{12} \\ s_{12} & c_{12} \end{bmatrix}\begin{bmatrix} L_1 s_2 & 0 \\ L_2 + L_1 c_2 & L_2 \end{bmatrix}\right)^T \cdot \begin{bmatrix} {}^0f_x \\ {}^0f_y \end{bmatrix}$$

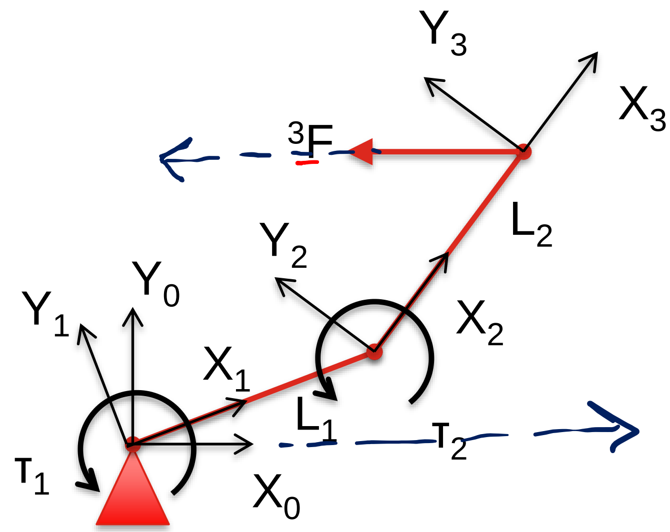
$$= \begin{bmatrix} -L_1 s_1 - L_2 s_{12} & -L_2 s_{12} \\ L_1 c_1 + L_2 c_{12} & L_2 c_{12} \end{bmatrix}^T \cdot \begin{bmatrix} {}^0f_x \\ {}^0f_y \end{bmatrix}$$

$$= \begin{bmatrix} -L_1 s_1 - L_2 s_{12} & L_1 c_1 + L_2 c_{12} \\ -L_2 s_{12} & L_2 c_{12} \end{bmatrix} \cdot \begin{bmatrix} {}^0f_x \\ {}^0f_y \end{bmatrix}$$

$^0J^T$ $\quad$ $^0J^T$ $\quad$ ^0F



Example



- With numerical values:
  - $L_1 = 1$, $L_2 = 1$
  - $\theta_1 = 30\text{deg}$, $\theta_2 = 30\text{deg}$
  - $|{}^3F| = 10\text{N}$
- With these, we have:

$$f_x = {}^3 f_x = -5 \qquad {}^0 f_x = \boxed{-10}$$
$$f_y = {}^3 f_y = 8.66 \qquad {}^0 f_y = \boxed{0}$$

- Substituting the values into either

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} L_1 s_2 & L_2 + L_1 c_2 \\ 0 & L_2 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

or

$$\tau = \begin{bmatrix} -L_1 s_1 - L_2 s_{12} & L_1 c_1 + L_2 c_{12} \\ -L_2 s_{12} & L_2 c_{12} \end{bmatrix} \cdot \begin{bmatrix} {}^0 f_x \\ {}^0 f_y \end{bmatrix}$$

gives

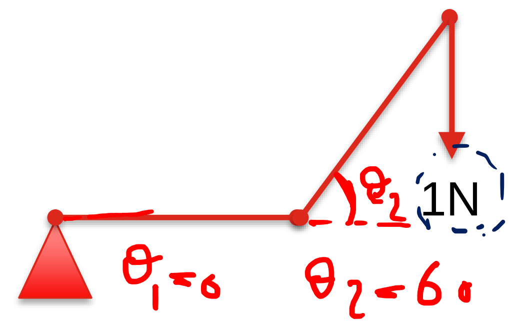
$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} 13.66 \\ 8.66 \end{bmatrix}$$

Another Example

- Still the same two link robot, but at different configuration and force.
- Case 1: $\theta_1 = 0$, $\theta_2 = 60$ deg, F = 1N.

$$\tau = \begin{bmatrix} -L_1s_1 - L_2s_{12} & L_1c_1 + L_2c_{12} \\ -L_2s_{12} & L_2c_{12} \end{bmatrix} \cdot \begin{bmatrix} {}^0f_x \\ {}^0f_y \end{bmatrix}$$

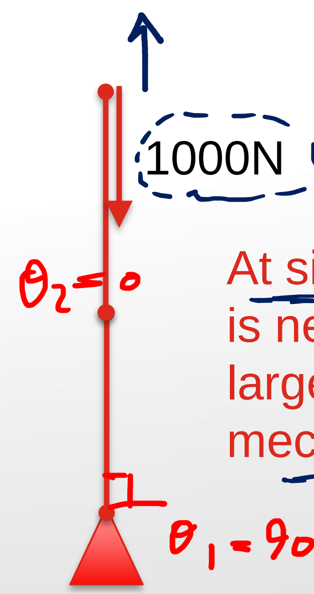
$$= \begin{bmatrix} -0.866 & 1.5 \\ -0.866 & 0.5 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} -1.5 \\ -0.5 \end{bmatrix} \begin{array}{l} \tau_1 \\ \tau_2 \end{array}$$



- Case 2: $\theta_1 = 90$ deg, $\theta_2 = 0$, F = 1000N.

$$\tau = \begin{bmatrix} -L_1s_1 - L_2s_{12} & L_1c_1 + L_2c_{12} \\ -L_2s_{12} & L_2c_{12} \end{bmatrix} \cdot \begin{bmatrix} {}^0f_x \\ {}^0f_y \end{bmatrix}$$

$$= \begin{bmatrix} -2 & 0 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ -1000 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{array}{l} \tau_1 \\ \tau_2 \end{array}$$



At singularity, no torque is needed to withstand large forces → mechanical advantage!

Content

- Introduction - Jacobian
- Method 1 - Direct differentiation (for Linear Jacobian)
- Method 2 - Velocity Propagation from Link to Link
- Method 3 - Explicit Form (for your study, not included in exam)
- Static Forces in Manipulators
- Singularities

Kinematic Singularities

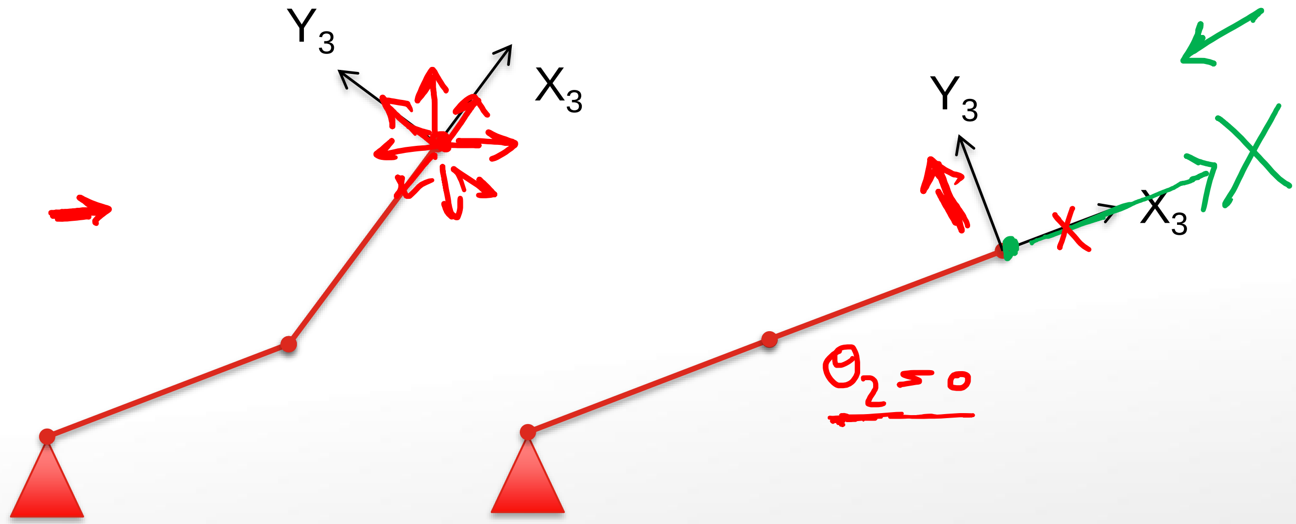
The robot loses some DoF

Rotations R
& Express the Rot $\begin{cases} \alpha \\ \beta \\ \gamma \end{cases}$

- Kinematic singularity happens when the end-effector loses the ability to move in a direction, or to rotate about a direction.
  - THE direction is called the "Singular Direction".

$\beta = 0, -180$
α, γ

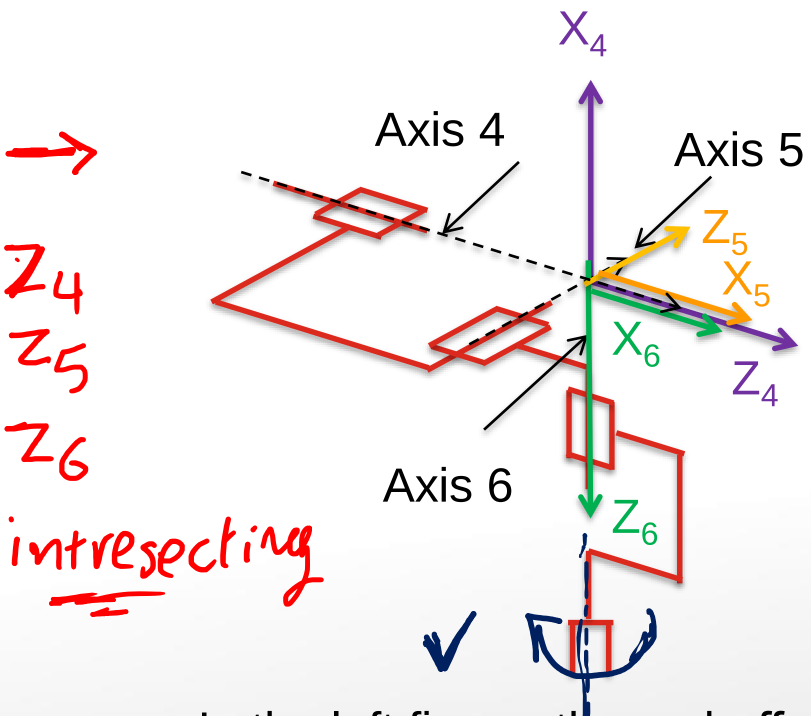
- E.g. Two-link robot:

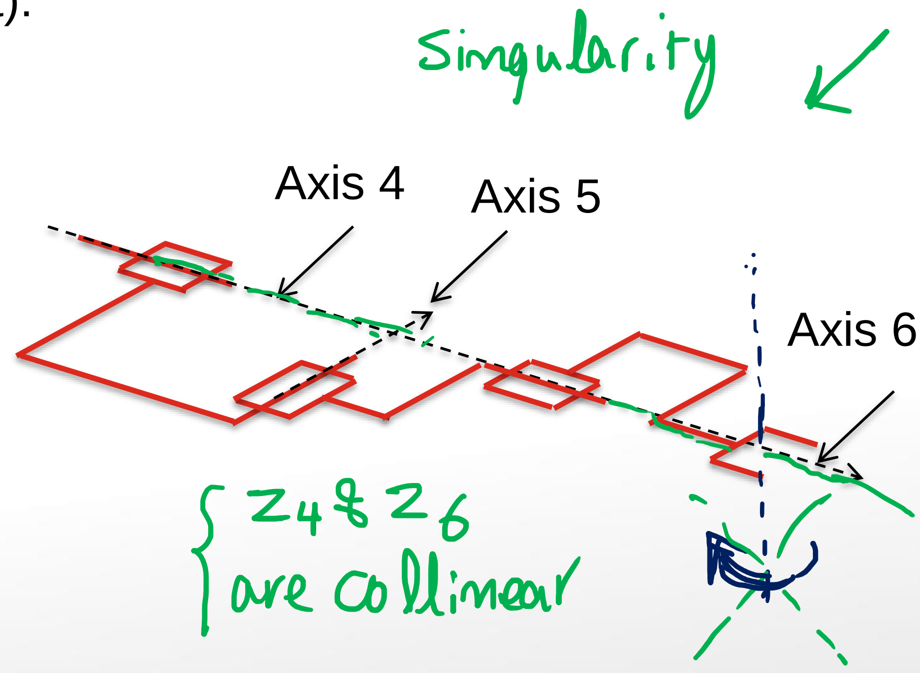


- In the left figure, the end-effector can move in any direction instantly.
- In the right figure, the end-effector loses the ability to move in the x-direction.

Kinematic Singularities

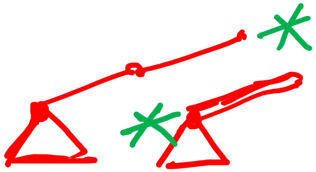
- Another example: 6-link robot, with the last 3 axes intersecting.
- Looking at the last 3 joints (the wrist):





- In the left figure, the end-effector can rotate about a "vertical axis".
- In the right figure, when axes 4 and 6 are collinear, the end-effector loses ability to rotate about the axis.

Kinematic Singularities



- The first example is a case of workspace-boundary singularities.
  - Manipulator is fully stretched or folded back on itself.
- The second example is a case of workspace-interior singularities.
  - Generally caused by lining up two or more axes.

Collinear

6 Link Z_4 & Z_6

- Mathematically, these singularities happen when the Jacobian matrix becomes non-invertible / singular.
  - We knew: $v = J(q)\dot{q}$

$v = J\dot{\theta} \rightarrow \dot{\theta} = J^{-1} v$ ≈0 ∞ ∞

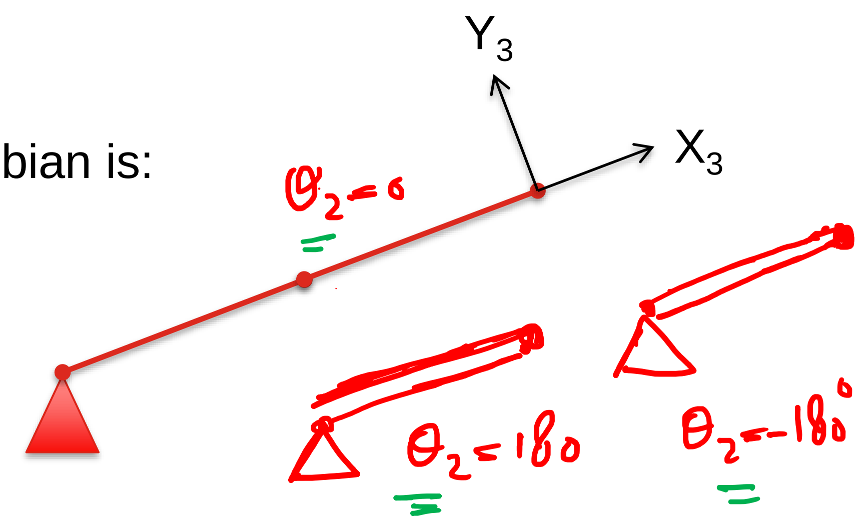
  - The inverse question is: If we “want” a certain v, what should the joint rate be?
  - This can be obtained via: $\dot{q} = J^{-1}(q)v$

$|J| \rightarrow 0 \rightarrow$ Singularity

  - If the Jabobian matrix is not invertible or ill-conditioned (determinant close to zero), then we need an infinitely big joint rate to obtain the v. (just imagine q_dot = 1/J * v with J = 0 or J ≈ 0, for scalar cases)
    - Not possible or practical.

Example

2-Link



- For the two-link robot example, the Jacobian is:

$$\underbrace{\begin{bmatrix} {}^0\upsilon_{3x} \\ {}^0\upsilon_{3y} \end{bmatrix}}_{v} = \underbrace{\underbrace{\begin{bmatrix} -L_1 s_1 - L_2 s_{12} & -L_2 s_{12} \\ L_1 c_1 + L_2 c_{12} & L_2 c_{12} \end{bmatrix}}_{{}^0J_v}}_{J} \underbrace{\begin{bmatrix} \dot\theta_1 \\ \dot\theta_2 \end{bmatrix}}_{\dot\theta}$$

- The determinant of the Jacobian is: $|J| = 0$

$$\det\begin{bmatrix} -L_1 s_1 - L_2 s_{12} & -L_2 s_{12} \\ L_1 c_1 + L_2 c_{12} & L_2 c_{12} \end{bmatrix} = -L_1 L_2 s_1 c_{12} - L_2^2 s_{12} c_{12} + L_1 L_2 c_1 s_{12} + L_2^2 s_{12} c_{12}$$

$$= \boxed{L_1 L_2 s_2} \qquad L_1 \,\&\, L_2$$

- The determinant is zero, i.e. the Jacobian becomes singular, when $\boxed{s_2 = 0}$ i.e.

Sin θ_2

$$\theta_2 = k\pi \qquad \begin{cases} \theta_2 = 0 \\ \theta_2 = \pm 180 \end{cases}$$

Resolved Motion Rate Control

- Here, we would like to show another useful application of the Jacobians, apart from calculating velocity and force.
- The Jacobian is widely used to control the robots:
  - At the current moment, the joint angles are q=(q1,…,qn), and we know the tip's position (from forward kinematics) at x.
  - Next, we want the tip's position to move somewhere else. How should the joint angles change? A • $\dot{\theta}$? • B
    - Last week, we saw how difficult it is to solve this inverse kinematics question, because it is a nonlinear problem. $\dot{x} = J\dot{q}$
    - However, we now know that for small changes: $\delta x = J(q)\delta q$ $\delta x = J\,\delta q$
    - Outside of singularity, we have: $\delta q = J^{-1}(q)\delta x$ $\delta q = J^{-1}\delta x$
    - Hence: Split the path into many small paths δx
    - And at each step, calculate $\delta q = J^{-1}(q)\delta x$ and set $q_{\text{new}} = q + \delta q$

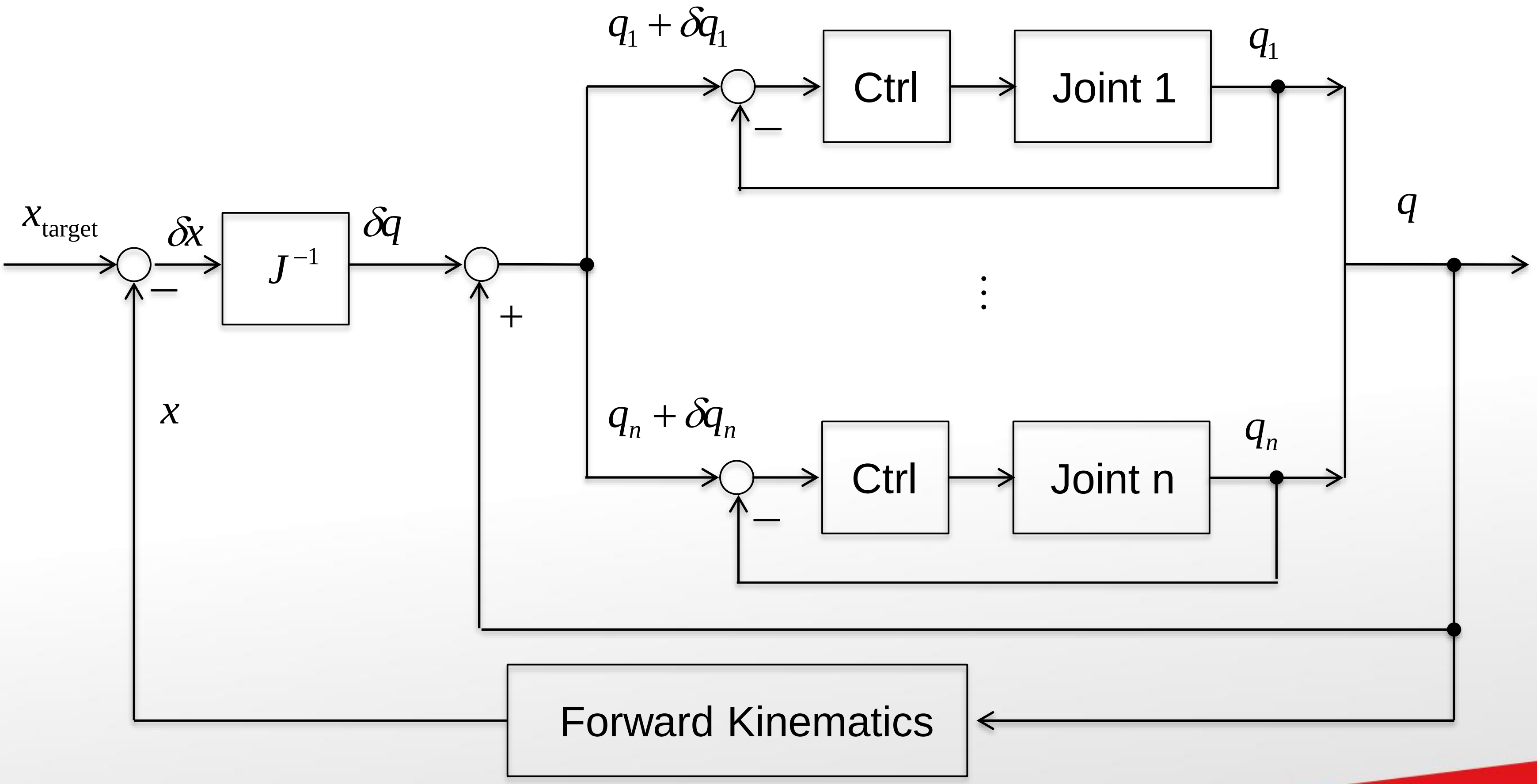
δx

A B

δq

Resolved Motion Rate Control

- The block diagram is as follows:



Thank you!

Have a good evening.