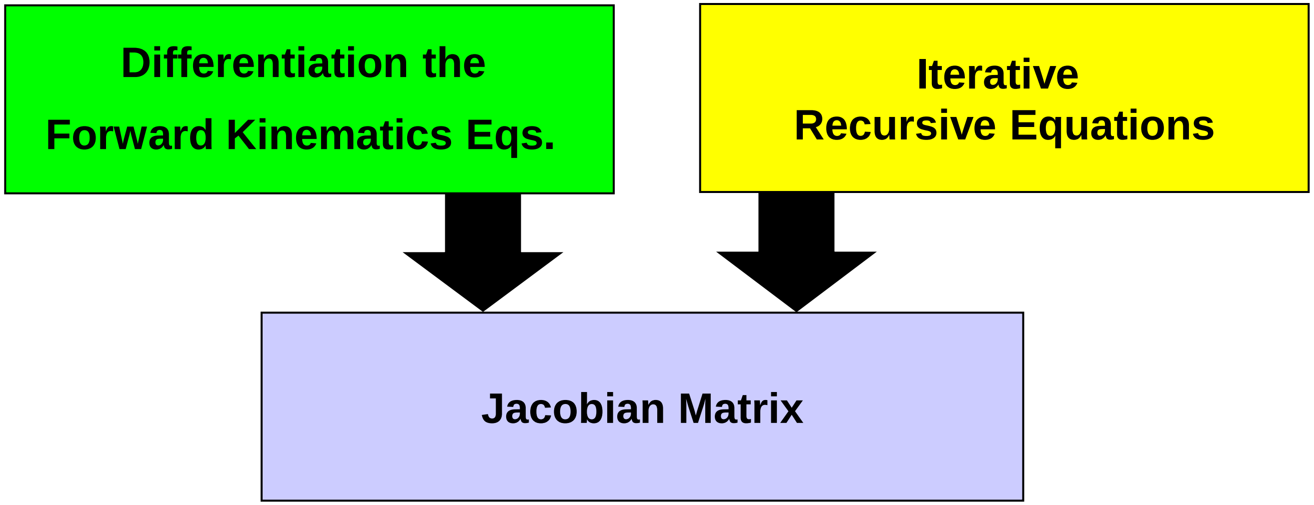
Jacobian: Velocity propagation 4/4

Jacobian Matrix - Calculation Methods



Jacobian: Velocity propagation

- The recursive expressions for the adjacent joint linear and angular velocities describe a relationship between the joint angle rates ($\dot{\theta}$) and the transnational and rotational velocities of the end effector ($\dot{X}$):

$$
{}^{i+1}\omega_{i+1} = {}^{i+1}_{\;\;i}R\,{}^{i}\omega_{i} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix}
$$

$$
{}^{i+1}v_{i+1} = {}^{i+1}_{\;\;i}R\left({}^{i}\omega \times {}^{i}P_{i+1} + {}^{i}v_{i}\right) + \begin{bmatrix} 0 \\ 0 \\ \dot{d}_{i+1} \end{bmatrix}
$$

Jacobian: Velocity propagation

- Therefore the recursive expressions for the adjacent joint linear and angular velocities can be used to determine the Jacobian in the end effector frame

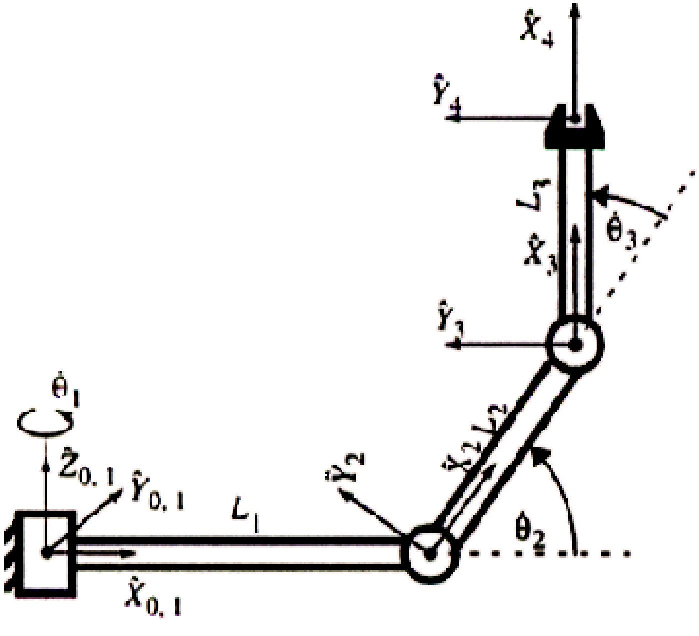
$$^{N}\dot{X} = {}^{N}J(\theta)\dot{\theta}$$

- This equation can be expanded to:

$$^{N}\dot{X} = {}^{N}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \Omega x \\ \Omega y \\ \Omega z \end{bmatrix} = \begin{bmatrix} {}^{N}v_{N} \\ {}^{N}\omega_{N} \end{bmatrix} = {}^{N}\begin{bmatrix} & J(\theta) & \end{bmatrix} \begin{bmatrix} \dot{\theta}1 \\ \dot{\theta}2 \\ \cdots \\ \dot{\theta}_{N} \end{bmatrix}$$

Jacobian - 3R - Example

- The linear angular velocities of the end effector (N=4)



$$
{}^4v_4 = \begin{bmatrix} L2s3\dot{\theta}_2 \\ (L2c3 + L3)\dot{\theta}_2 + L3\dot{\theta}_3 \\ (-L1 - L2c2 - L3c23)\dot{\theta}_1 \end{bmatrix}
$$

$$
{}^4\omega_4 = \begin{bmatrix} s23\dot{\theta}_1 \\ c23\dot{\theta}_1 \\ \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix}
$$

Jacobian - 3R - Example

- Re-arranged to previous two terms gives

$$
{}^{4}\dot{X} = {}^{4}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = \begin{bmatrix} {}^{4}v_4 \\ {}^{4}\omega_4 \end{bmatrix} = \begin{bmatrix} L2s3\dot{\theta}_2 \\ (L2c3 + L3)\dot{\theta}_2 + L3\dot{\theta}_3 \\ (-L1 - L2c2 - L3c23)\dot{\theta}_1 \\ s23\dot{\theta}_1 \\ c23\dot{\theta}_1 \\ \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix}
$$

- We can now factor out the joint velocities vector $\dot{\theta} = [\dot{\theta}_1 \dot{\theta}_2 \dot{\theta}_3]^T$ from the above matrix to give

Jacobian - 3R - Example

$$
{}^{4}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = {}^{4}\begin{bmatrix} 0 & L2s3 & 0 \\ 0 & L2c3+L3 & L3 \\ -L1-L2c2-L3c23 & 0 & 0 \\ s23 & 0 & 0 \\ c23 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}
$$

$$
{}^{4}\dot{X} = {}^{4}J(\theta)\dot{\theta}
$$

- The equations for ${}^{N}v_N$ and ${}^{N}\omega_N$ are always a linear combination of the joint velocities, so they can always be used to find the 6xN Jacobian matrix (${}^{N}J(\theta)$) for any robot manipulator.

- Note that the Jacobian matrix is expressed in frame {4}

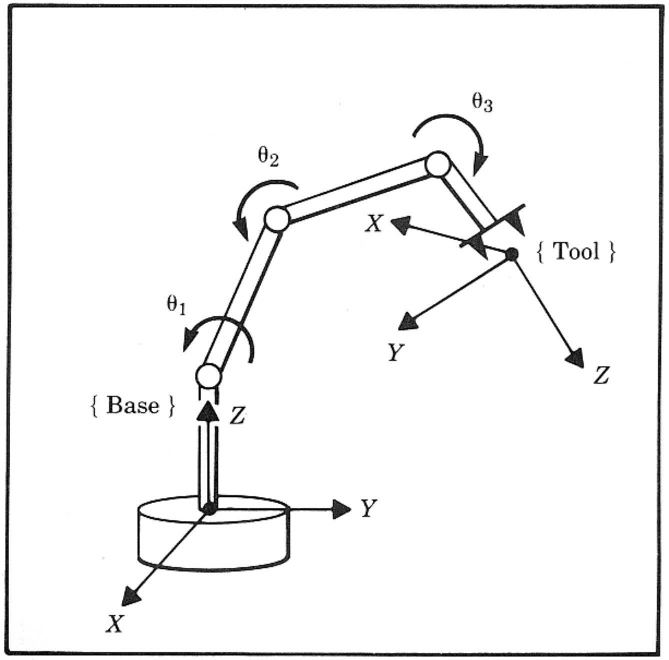
Jacobian: Frame of Representation

- Using the velocity propagation method we expressed the relationship between the velocity of the robot end effector measured relative to the robot base frame {0} and expressed in the end effector frame {N}.

$$^{N}\dot{X} = {}^{N}J(\theta)\dot{\theta}$$

- Occasionally, it may be desirable to express (represent) the end effector velocities in another frame (e.g. frame {0}, in which case we will need a method to provide the transformation.

$$^{0}\dot{X} = {}^{0}J(\theta)\dot{\theta}$$



Jacobian: Frame of Representation

- Consider the velocities in a different frame {B}

$$ {}^B\dot{X} = \begin{bmatrix} {}^B v_N \\ {}^B \omega_N \end{bmatrix} = {}^B J(\theta)\dot{\theta} $$

- We may use the rotation matrix to find the velocities in frame {A}:

$$ {}^A\dot{X} = \begin{bmatrix} {}^A v_N \\ {}^A \omega_N \end{bmatrix} = \begin{bmatrix} {}^A_B R \, {}^B v_N \\ {}^A_B R \, {}^B \omega_N \end{bmatrix} $$

- The Jacobian transformation is given by a rotation matrix

$$ {}^A\dot{X} = {}^A J(\theta)\dot{\theta} = {}^A_B R_J \, {}^B J(\theta)\dot{\theta} $$

Jacobian: Frame of Representation

- where ${}^{A}_{B}R_{J}$ is given by

$$
{}^{A}_{B}R_{J} = \begin{bmatrix} \left[{}^{A}_{B}R\right] & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \left[{}^{A}_{B}R\right] \end{bmatrix}
$$

or equivalently,

$$
{}^{A}J(\theta) = \begin{bmatrix} \left[{}^{A}_{B}R\right] & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \left[{}^{A}_{B}R\right] \end{bmatrix} {}^{B}J(\theta)
$$

Jacobian: Frame of Representation

- The Jacobian provides the relationship between the end effector's Cartesian velocity measured relative to the robot base frame {0}

- For velocity expressed in frame {N}

$$^{N}\dot{X} = {}^{N}J(\theta)\dot{\theta}$$

- For velocity expressed in frame {0}

$$^{0}\dot{X} = {}^{0}J(\theta)\dot{\theta}$$

Jacobian: Frame of Representation - 3R Example

$$
{}^{0}J(\theta)=\begin{bmatrix} \begin{bmatrix}{}^{0}_{4}R\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} \\ \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} & \begin{bmatrix}{}^{0}_{4}R\end{bmatrix} \end{bmatrix} {}^{4}J(\theta)=\begin{bmatrix} \begin{bmatrix} c1c23 & -c1s23 & s1 \\ s1c23 & -s1s23 & -c1 \\ s23 & c23 & 0 \end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} \\ \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} & \begin{bmatrix} c1c23 & -c1s23 & s1 \\ s1c23 & -s1s23 & -c1 \\ s23 & c23 & 0 \end{bmatrix} \end{bmatrix} {}^{4}\begin{bmatrix} 0 & L2s3 & 0 \\ 0 & L2c3+L3 & L3 \\ -L1-L2c2-L3c23 & 0 & 0 \\ s23 & 0 & 0 \\ c23 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}
$$

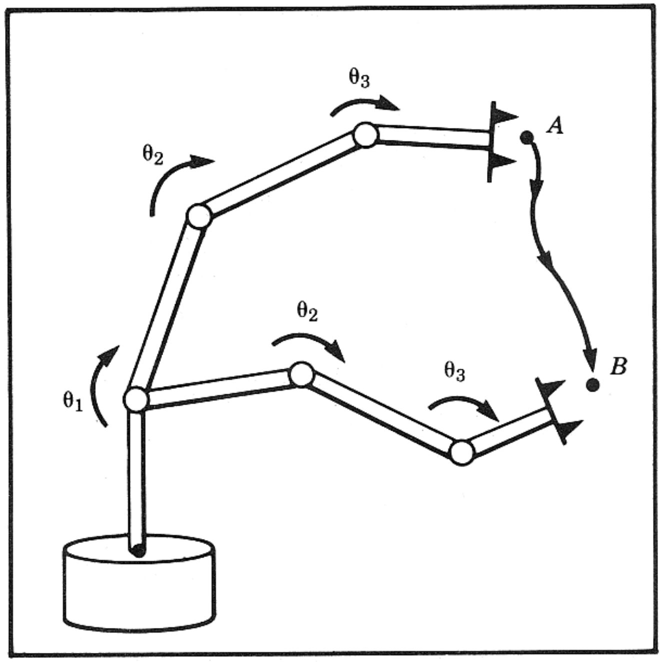
- The rotation matrix (${}^{0}_{4}R$) can be calculated base on the homogeneous transforms given by

$$
{}^{0}_{4}T={}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T
$$

Inverse Jacobian

- ***Given***
  - Tool tip path (defined mathematically)
  - Tool tip position/orientation
  - Tool tip velocity
  - Jacobian Matrix

$$\underline{\dot{x}} = J(\theta)\,\underline{\dot{\theta}}$$



- ***Problem:*** Calculate the joint velocities

- ***Solution:***
  - Compute the inverse Jacobian matrix
  - Use the following equation to compute the joint velocity

$$\underline{\dot{\theta}} = J(\theta)^{-1}\,\underline{\dot{x}}$$

Inverse Jacobian

- Cases in which the Jacobian matrix $J(\theta)$ is not inevitable ($J(\theta)^{-1}$ does not exists). Non invertible matrix is called singular matrix

  – **Case 1** - The Jacobian matrix is not squared

    In general the 6xN Jacobian matrix may be non-square in which case the inverse is not defined

  – **Case 2** - The determinant ($\det(J(\theta))$) is equal to zero

Inverse Jacobian - Reduced Jacobian

- ***Problem***
  - When the number of joints (N) is less than 6, the manipulator does not have the necessary degrees of freedom to achieve independent control of all six velocities components.

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \Omega x \\ \Omega y \\ \Omega z \end{bmatrix}$$

- ***Solution***
  - We can reduce the number of rows in the original Jacobian to describe a reduced Cartesian vector. For example, the full Cartesian velocity vector is given by

Jacobian: Reduced Jacobian - 3R Example

- Matrix Reduction - Option 1

$$
{}^{4}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \Omega x \\ \Omega y \\ \Omega z \end{bmatrix} = {}^{4}\begin{bmatrix} 0 & L2s3 & 0 \\ 0 & L2c3 + L3 & L3 \\ -L1 - L2c2 - L3c23 & 0 & 0 \\ s23 & 0 & 0 \\ c23 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}
$$

- Column of zeroes
- The determinate is equal to zero
- Only two out of the three variables can be independently specified

Jacobian: Reduced Jacobian - 3R Example

- Matrix Reduction - Option 2

$$
{}^{4}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \Omega x \\ \Omega y \\ \Omega z \end{bmatrix} = {}^{4}\begin{bmatrix} 0 & L2s3 & 0 \\ 0 & L2c3+L3 & L3 \\ -L1-L2c2-L3c23 & 0 & 0 \\ s23 & 0 & 0 \\ c23 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}
$$

- Two columns of zeroes
- The determinate is equal to zero
- Only one out of the three variables can be independently specified

Jacobian: Reduced Jacobian - 3R Example

- Matrix Reduction - Option 3

$$
{}^{4}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \Omega x \\ \Omega y \\ \Omega z \end{bmatrix} = {}^{4}\begin{bmatrix} 0 & L2s3 & 0 \\ 0 & L2c3 + L3 & L3 \\ -L1 - L2c2 - L3c23 & 0 & 0 \\ s23 & 0 & 0 \\ c23 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}
$$

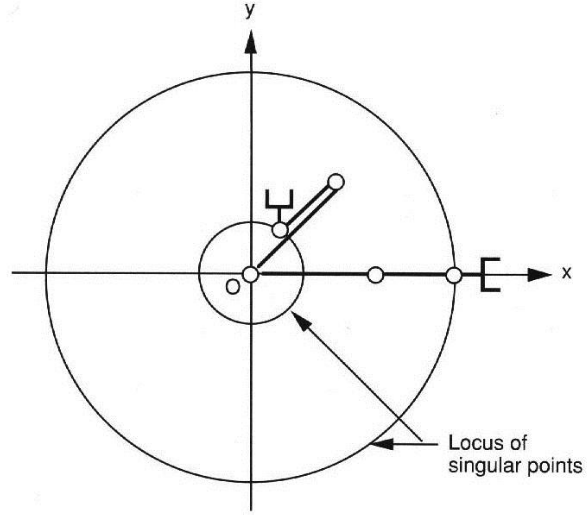
- The resulting reduced Jacobian will be square (the number of independent rows in the Jacobian are equal to the number of unknown variables) and can be inverted unless in a singular configuration.

Jacobian: Singularities

- To avoid singular configurations, the determinant of the Jacobian is often computed symbolically to find the set of joint values for which singularities will occur. Singularities often occur under two situations:

1. Workspace Boundary: the manipulator is fully extended or folded back upon itself.

2. Workspace Interior: generally caused by two or more axes intersecting.



Jacobian: Singular Configuration - 3R Example

- If we want to use the inverse Jacobian to compute the joint angular velocities we need to first find out at what points the inverse exists.

$$\underline{\dot{\theta}} = J(\theta)^{-1}\underline{\dot{x}}$$

- Considering the 3R robot

$$^{4}J_{r}(\theta)=\begin{bmatrix} 0 & L2s3 & 0 \\ 0 & L2c3+L3 & L3 \\ -L1-L2c2-L3c23 & 0 & 0 \end{bmatrix}$$

- The determinate of the Jacobian is defined as follows

$$\left|{}^{4}J_{r}(\theta)\right| = -(L1+L2c2+L3c23)(L2s3)L3$$

Jacobian: Singular Configuration - 3R Example

$$\left|{}^{4}J_{r}(\theta)\right| = -(L1 + L2c2 + L3c23)(L2s3)L3$$

- The reduced Jacbian matrix is singular when it determinate is equal to zero

$$-(L1 + L2c2 + L3c23)(L2s3)L3 = 0$$

- The singular condition occur when either of the following are true

$$s3 = 0$$

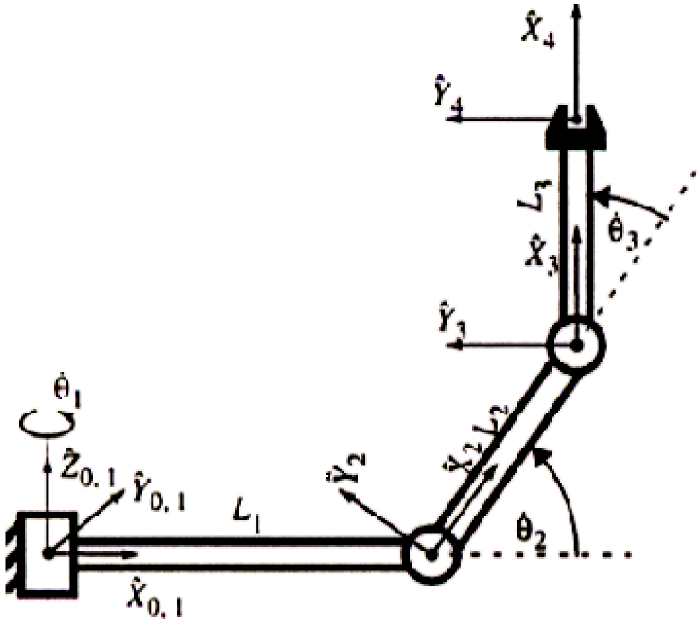
$$-L1 - L2c2 - L3c23 = 0$$

Jacobian: Singular Configuration - 3R Example

- Case 1: $s3 = 0$

$$s3 = 0 \Rightarrow \begin{cases} \theta_3 = 0^o \\ \theta_3 = 180^o \end{cases}$$

$${}^4J_r(\theta) = \begin{bmatrix} 0 & L2s3 & 0 \\ 0 & L2c3 + L3 & L3 \\ -L1 - L2c2 - L3c23 & 0 & 0 \end{bmatrix}$$



- The first row of the Jacobian is zero
- The 3R robot is loosing one DOF.
- The robot can no longer move along the X-axis of frame {4}

Jacobian: Singular Configuration - 3R Example

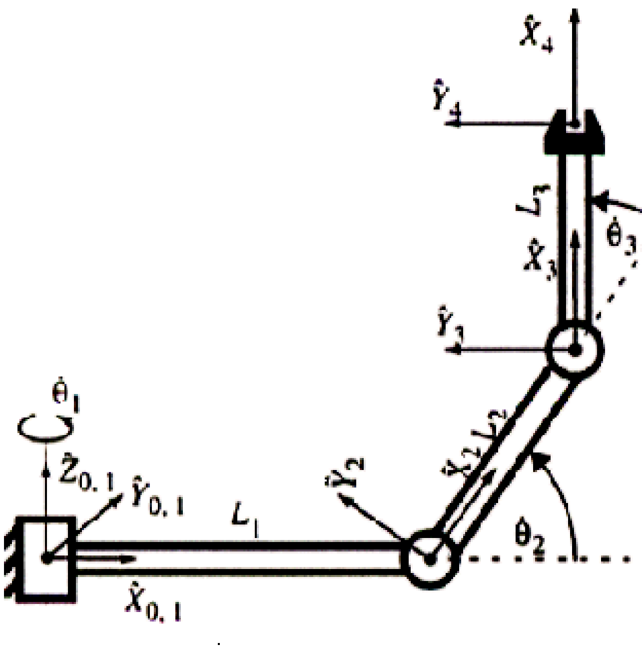
- Case 2: $-L1-L2c2-L3c23=0$

$$L1=-L2c2-L3c23$$

- Occur only if $L2+L3\geq L1$

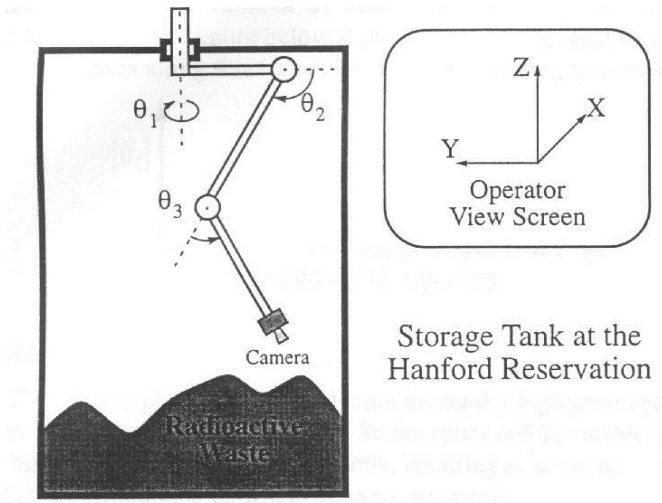
$$^{4}J_r(\theta)=\begin{bmatrix} 0 & L2s3 & 0 \\ 0 & L2c3+L3 & L3 \\ -L1-L2c2-L3c23 & 0 & 0 \end{bmatrix}$$

- The third row of the Jacobian is zero
- The origin of frame {4} intersects the Z-axis of frame {1}
- The 3R robot is loosing one DOF.
- The robot can no longer move along the Z-axis of frame {4}



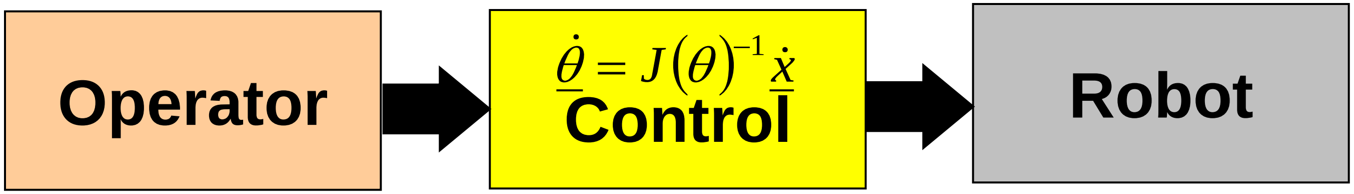
Joint Velocity Near Singular Position - 3R Example

- Robot : 3R robot
- Task: Visual inspection



Storage Tank at the
Hanford Reservation

- Control



Joint Velocity Near Singular Position - 3R Example

- ***Singularity (Case 2)-*** The origin of frame {4} intersects the Z-axis of frame {1}

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & L2s3 & 0 \\ 0 & L2c3 + L3 & L3 \\ -L1 - L2c2 - L3c23 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

- Solve for $\dot{\theta}_1$ in terms of $\dot{z}$ we find

$$\dot{\theta}_1 = \frac{\dot{z}}{-L1 - L2c2 - L3c23}$$

$$-L1 - L2c2 - L3c23 = 0$$

$$\dot{\theta}_1 \rightarrow \infty$$

Joint Velocity Near Singular Position - 3R Example

- Singularity - $$\det\left(J\left(\theta\right)\right)=0$$

- Problems:

  - **Motor Constrains** - The robot is physically limited from moving in unusual high joint velocities by motor power constrains. Therefore, the robot will be unable to track the required joint velocity trajectory exactly resulting in some perturbation to the commanded Cartesian velocity trajectory.
  - **Gears and Shafts** - The derivative of the angular velocity is the angular acceleration. The high acceleration of the joint resulting form approaching too close to a singularity may cause damage to the gear/shafts.
  - **DOF** - At a singular configuration (specific point in space) the manipulator loses one or more DOF.

- Consequences – Certain tasks can not be performed at a singular configuration

Designing Well Conditioned Workspace

- Difficulty in operating at
  - Workspace Boundaries
  - Neighborhood of singular point inside the workspace

- The further the manipulator is away from singularities the better it moves uniformly and apply forces in all directions

- **Manipulability Measure** - How far / close the manipulator is from singularity
  - Range $0 \le w < \infty$
  - For Non redundant manipulators $w = \det\left(J(\theta)\right)$
  - For redundant manipulators $w = \sqrt{\det\left(J(\theta)J^T(\theta)\right)}$
  - A good manipulator design has large area of characterized by high value of the manipulability (w)

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).

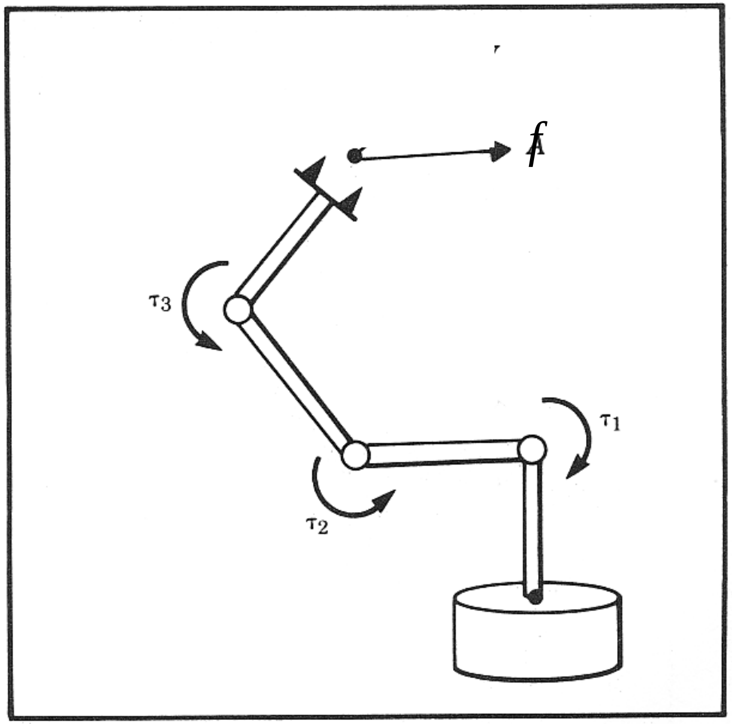
Compute: The joint torques which must be acting to keep the system in static equilibrium.

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.



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

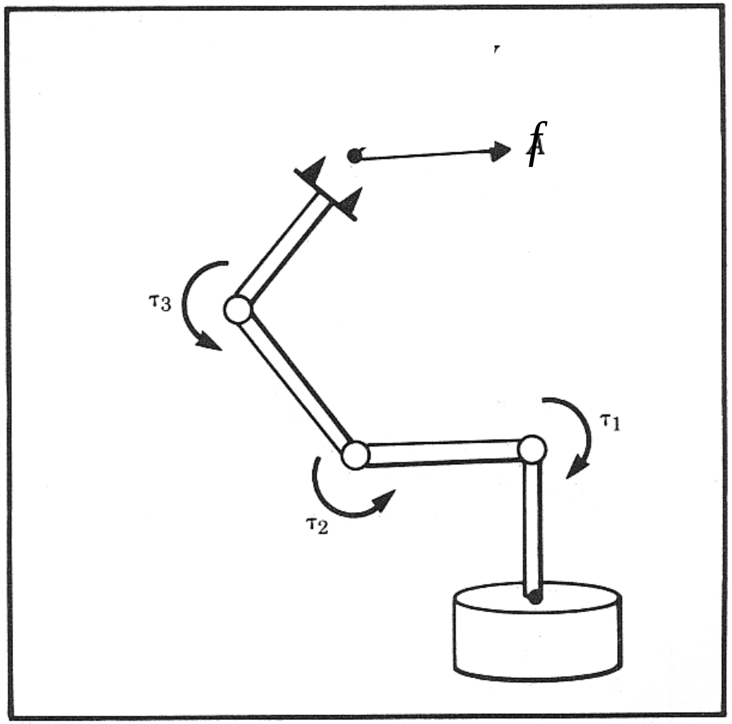
(3 Eqs.) $\sum F = 0$

(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)

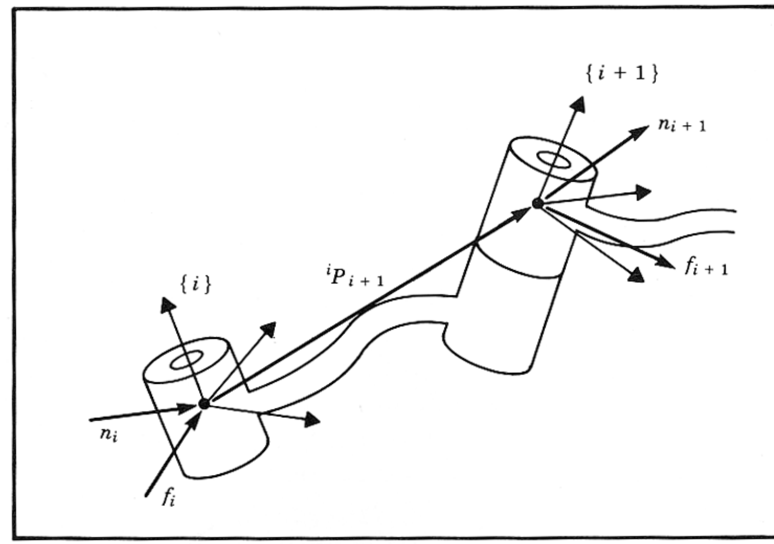


Static Analysis Protocol - Free Body Diagram 2/

- 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 $\boldsymbol{n}$

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

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.

$$\sum F = 0 \quad \Rightarrow \quad \sum F = {}^{i}f_i - {}^{i}f_{i+1} = 0$$

$$\sum M = 0 \quad \Rightarrow \quad \sum M = {}^{i}n_i - {}^{i}n_{i+1} - {}^{i}P_{i+1} \times {}^{i}f_{i+1} = 0$$

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 ***i+1*** to link ***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

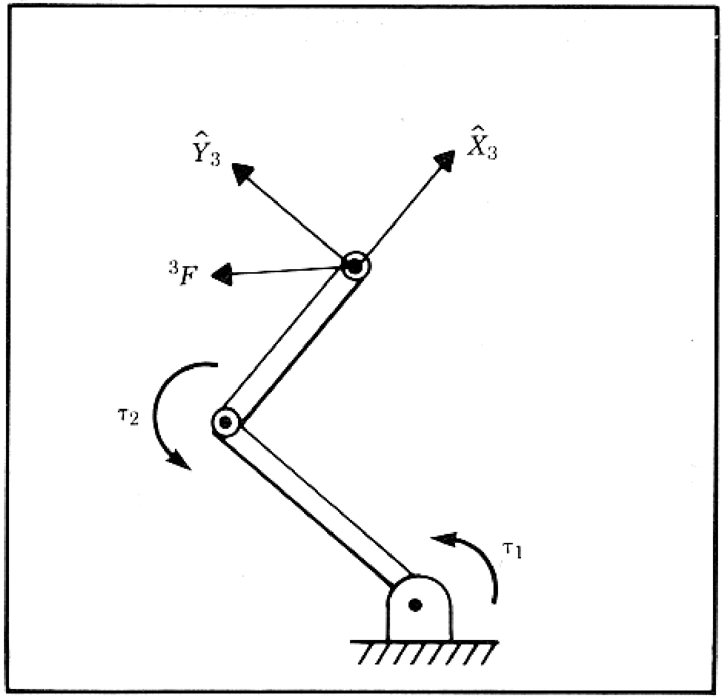
Problem

Given:

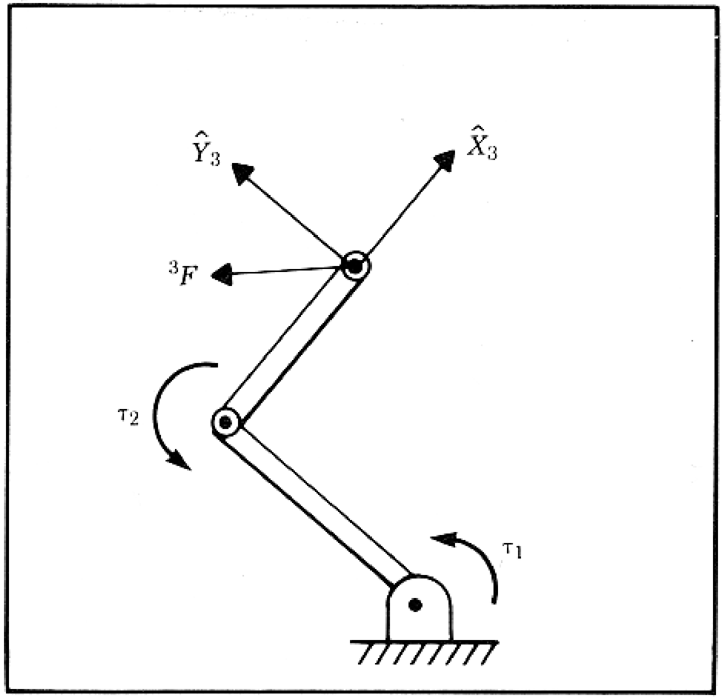
- 2R Robot
- A Force vector $^{3}f_{3}$ is applied **by** the end effector
- A torque vector $^{3}n_{3} = 0$

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

$$ {}^{1}f_1 = {}^{1}_{2}R \; {}^{2}f_2 $$

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

$$ {}^{1}n_1 = {}^{1}_{2}R \; {}^{2}n_2 + {}^{1}P_2 \times {}^{1}f_1 $$

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

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

Example - 2R Robot - Static Analysis

$$
{}^1n_1 = \begin{bmatrix} 0 \\ 0 \\ l_1 s_2 f_x + l_1 c_2 f_y + l_2 f_y \end{bmatrix} \qquad {}^2n_2 = \begin{bmatrix} 0 \\ 0 \\ l_2 f_y \end{bmatrix}
$$

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

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

Example - 2R Robot - Static Analysis

- Re-writing the equations in a matrix form

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

Jacobian – Duality