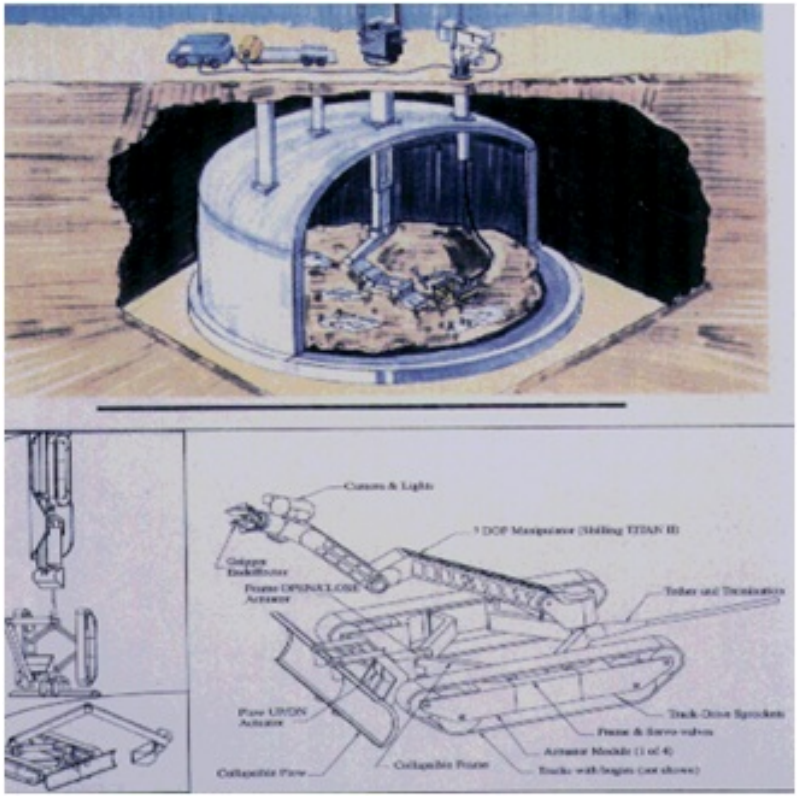
Kinematic Design of Redundant Robotic Manipulators that are Optimally Fault Tolerant

Presented by:
Khaled M. Ben-Gharbia

January 15th, 2013

ARticulated
Motion
LABoratory

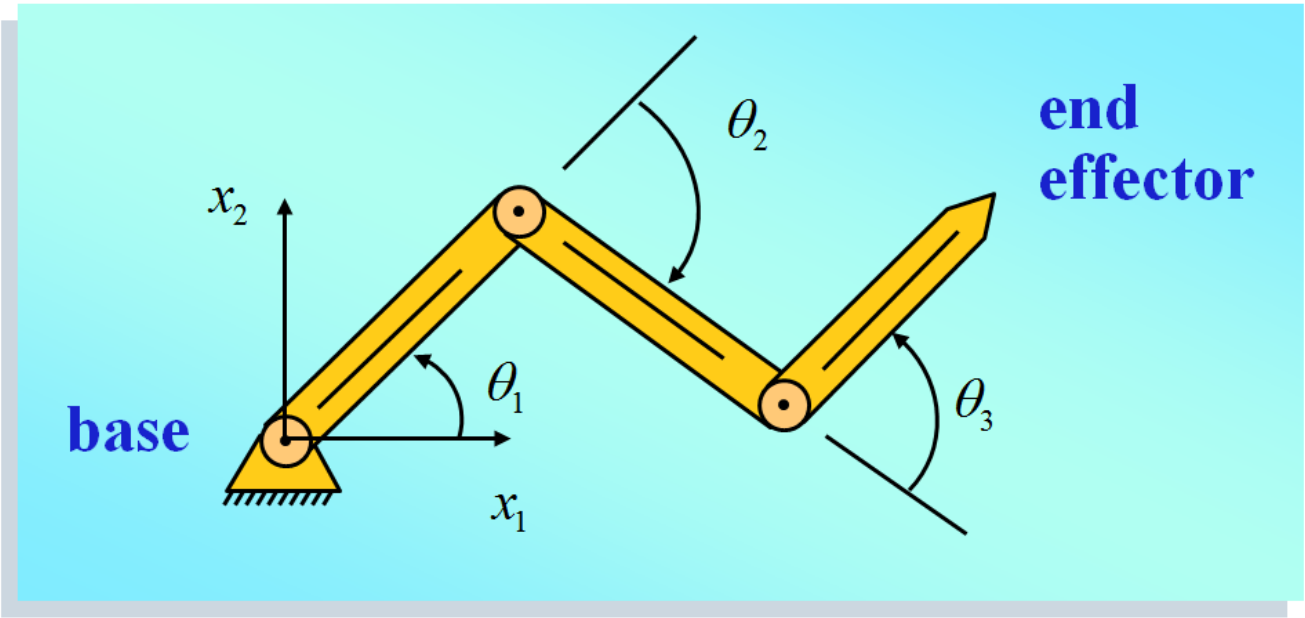
Why Fault Tolerance?



- Applications
  - hazardous waste cleanup
  - space/underwater exploration
  - anywhere failures are likely or intervention is costly
- Common failure mode
  - locked actuators

ARticulated
Motion
LABoratory

Simple Redundant Robot Planar 3DOF



$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = J(\theta) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

Jacobian (J) Matrix

- 3R planar manipulator
  - Forward Kinematics:

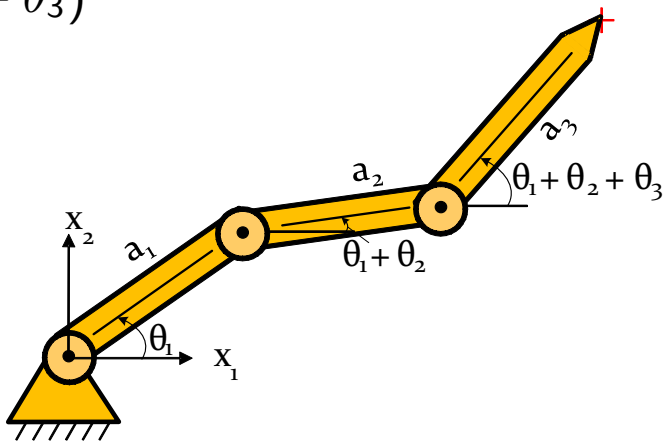
$$x = f(\theta)$$
$$x_1 = a_1 \cos(\theta_1) + a_2 \cos(\theta_1 + \theta_2) + a_3 \cos(\theta_1 + \theta_2 + \theta_3)$$
$$x_2 = a_1 \sin(\theta_1) + a_2 \sin(\theta_1 + \theta_2) + a_3 \sin(\theta_1 + \theta_2 + \theta_3)$$

$\implies$ taking the derivative w.r.t time
$\implies$ $\dot{x} = J(\theta)\dot{\theta}$

- Geometrically

$$J = [j_1 \;\; j_2 \;\; j_3] = [z_1 \times p_1 \;\; z_2 \times p_2 \;\; z_3 \times p_3]$$

where z_i is the rotation axis, and here it is $[0 \;\; 0 \;\; 1]^T$



- Spatial manipulator

$$j_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}$$

Jacobian (J) Matrix

- 3R planar manipulator
  - Forward Kinematics:

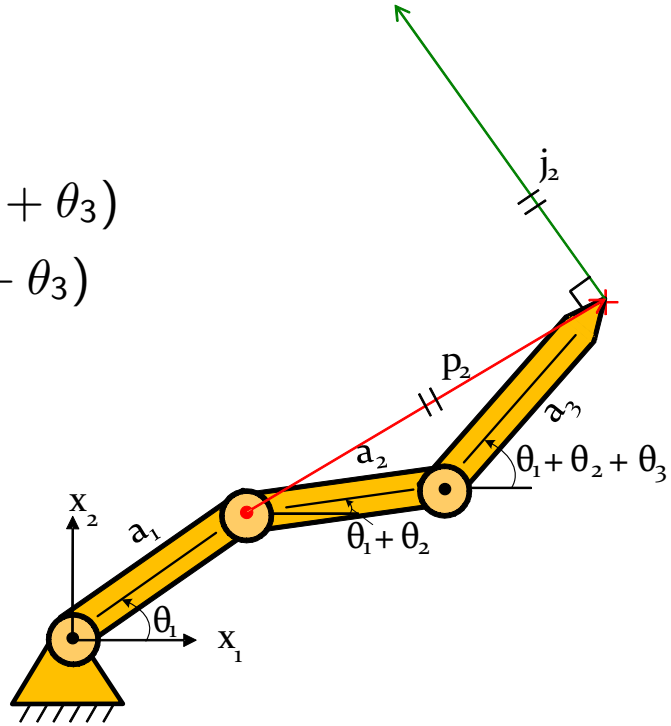
$$x = f(\theta)$$
$$x_1 = a_1 \cos(\theta_1) + a_2 \cos(\theta_1 + \theta_2) + a_3 \cos(\theta_1 + \theta_2 + \theta_3)$$
$$x_2 = a_1 \sin(\theta_1) + a_2 \sin(\theta_1 + \theta_2) + a_3 \sin(\theta_1 + \theta_2 + \theta_3)$$

$\Longrightarrow$ taking the derivative w.r.t time
$\Longrightarrow$ $\dot{x} = J(\theta)\dot{\theta}$

  - Geometrically

$$J = [j_1 \;\; j_2 \;\; j_3] = [z_1 \times p_1 \;\; z_2 \times p_2 \;\; z_3 \times p_3]$$

where z_i is the rotation axis, and here it is $[0 \;\; 0 \;\; 1]^T$



- Spatial manipulator

$$j_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}$$

Jacobian (J) Matrix

- 3R planar manipulator
  - Forward Kinematics:

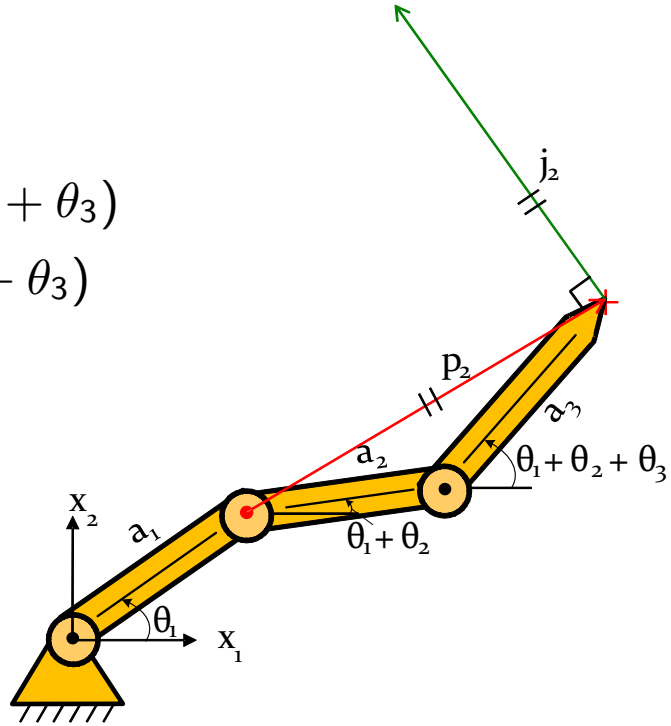
$$x = f(\theta)$$
$$x_1 = a_1\cos(\theta_1) + a_2\cos(\theta_1+\theta_2) + a_3\cos(\theta_1+\theta_2+\theta_3)$$
$$x_2 = a_1\sin(\theta_1) + a_2\sin(\theta_1+\theta_2) + a_3\sin(\theta_1+\theta_2+\theta_3)$$

$\Longrightarrow$ taking the derivative w.r.t time
$\Longrightarrow$ $\dot{x} = J(\theta)\dot{\theta}$

  - Geometrically

$$J = [j_1 \;\; j_2 \;\; j_3] = [z_1 \times p_1 \;\; z_2 \times p_2 \;\; z_3 \times p_3]$$

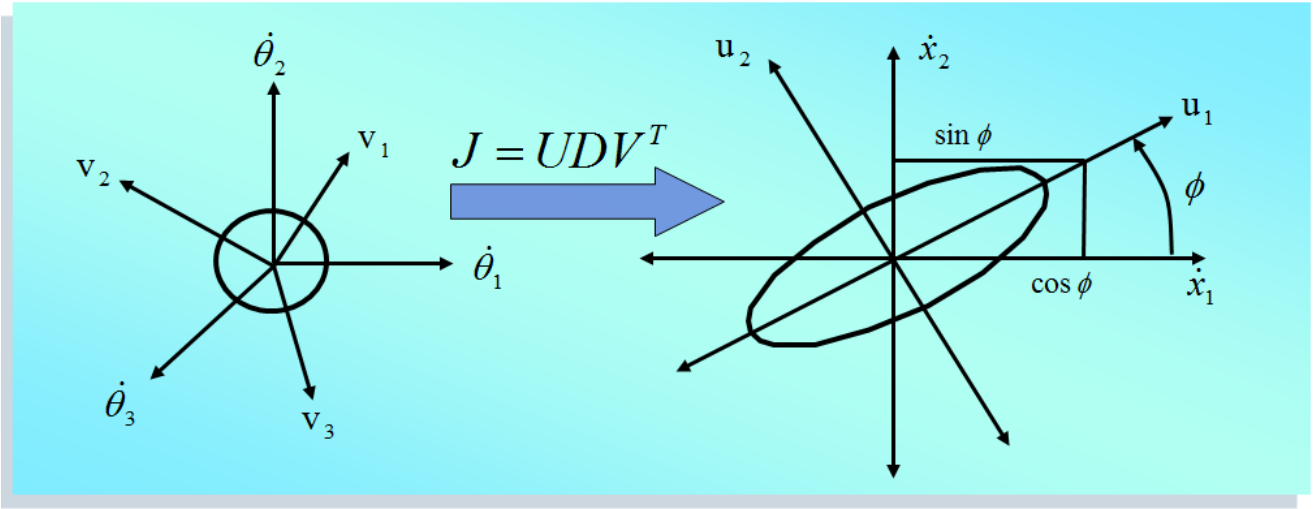
where z_i is the rotation axis, and here it is $[0 \;\; 0 \;\; 1]^T$



- Spatial manipulator

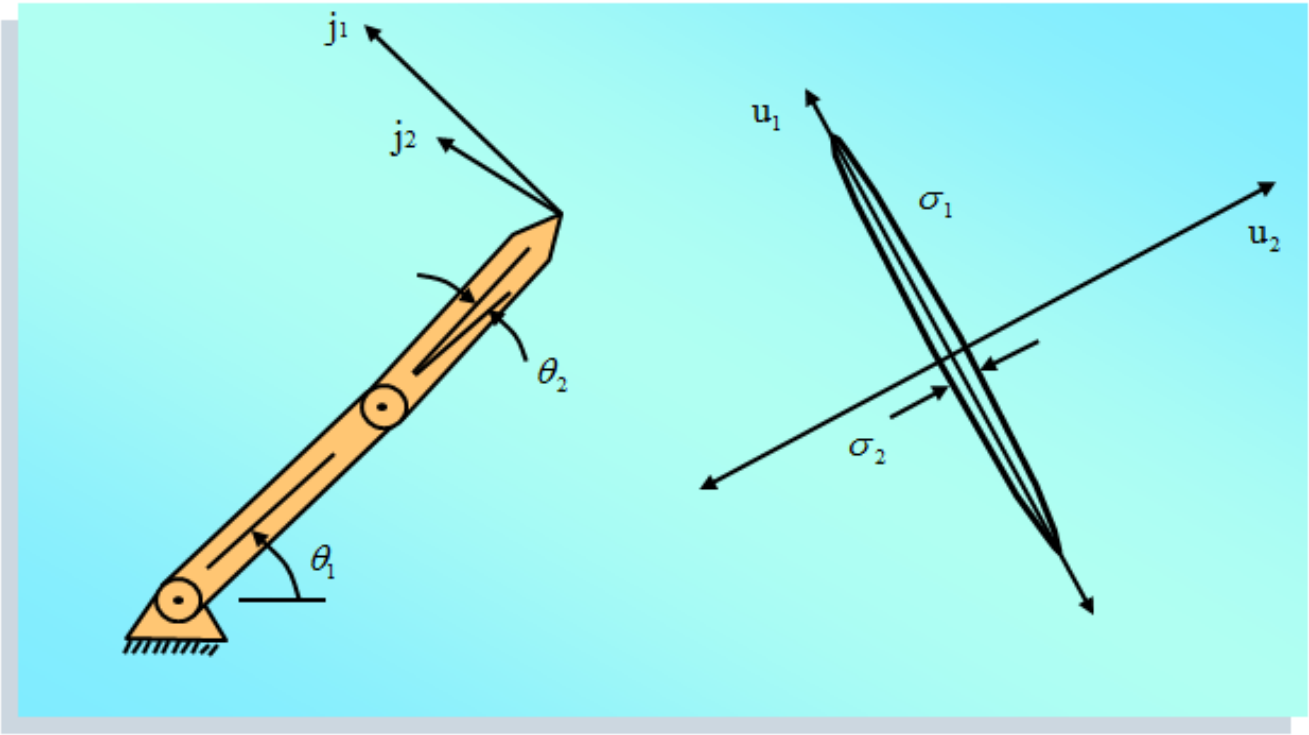
$$j_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}$$

Kinematic Dexterity Measures: Function of Singular Values of the Jacobian



$$V^T V = I \quad D = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \end{bmatrix} \quad U = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

Approaching Kinematic Singularities



- Singularity: $\sigma_2 = 0$
- Value of σ_m is a common dexterity measure

Optimal Fault Tolerant Configurations for Locked Joints Failures

Fault-free Jacobian: $J_{m\times n} = [j_1 \; j_2 \; \ldots \; j_n]$

Failure at joint f: ${}^f J_{m\times n-1} = [j_1 \; j_2 \; \ldots \; j_{f-1} \; j_{f+1} \; \ldots \; j_n]$

$$ {}^f J = \sum_{i=1}^{m} {}^f\sigma_i \; {}^f\hat{u}_i \; {}^f\hat{v}_i^T $$

Worst-case remaining dexterity: $\mathcal{K} = \min_{f=1}^{n} \; {}^f\sigma_m$

Isotropic & Optimal J: $\mathcal{K} = \sigma\sqrt{\frac{n-m}{n}}$ for all f

Isotropic and Optimally Fault Tolerant Jacobian

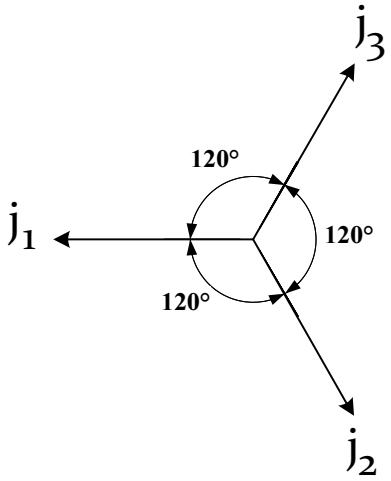
- equal σ_i's
- equal ${}^f\sigma_m$ for all f
- equal $\|j_i\|$ for all i

- Example: 2×3 isotropic and optimally fault tolerant J:

$$J = [\ j_1 \ \ j_2 \ \ j_3\ ] = \begin{bmatrix} -\sqrt{\frac{2}{3}} & \sqrt{\frac{1}{6}} & \sqrt{\frac{1}{6}} \\ 0 & -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix}$$

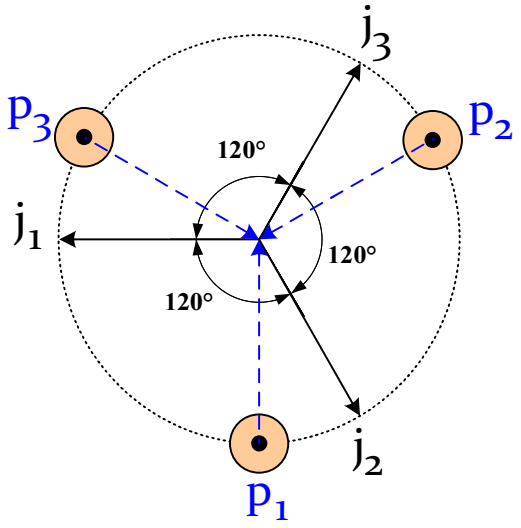
Null space equally distributed: $n_J = \left[\ \sqrt{\frac{1}{3}}\ \sqrt{\frac{1}{3}}\ \sqrt{\frac{1}{3}}\ \right]$

Example: 2×3 Optimally Fault Tolerant Jacobian



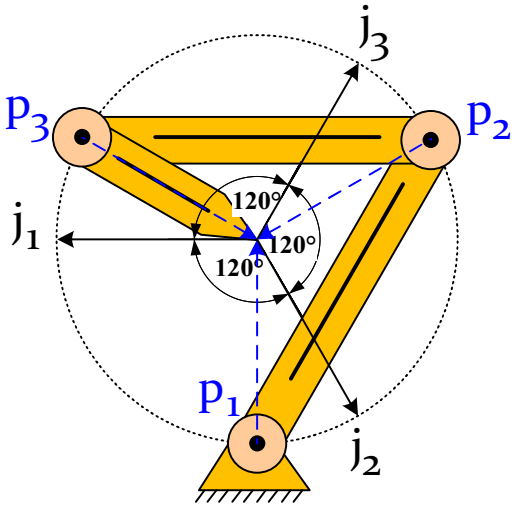
Remaining dexterity: $\mathcal{K} = {}^{f}\sigma_2 = \sqrt{\frac{1}{3}}$

Example: 2×3 Optimally Fault Tolerant Jacobian



Remaining dexterity: $\mathcal{K} = {}^{f}\sigma_2 = \sqrt{\frac{1}{3}}$

Example: 2×3 Optimally Fault Tolerant Jacobian



Remaining dexterity: $\mathcal{K} = {}^{f}\sigma_2 = \sqrt{\frac{1}{3}}$

Different Optimally Fault Tolerant Jacobians

- Sign change and permuting columns do not affect the isotropy and the optimally fault tolerant properties of J, e.g.

$$\begin{bmatrix} \pm j_1 & \pm j_2 & \pm j_3 \end{bmatrix}$$

- But each new J may belong to a different manipulator

Goal of This Research

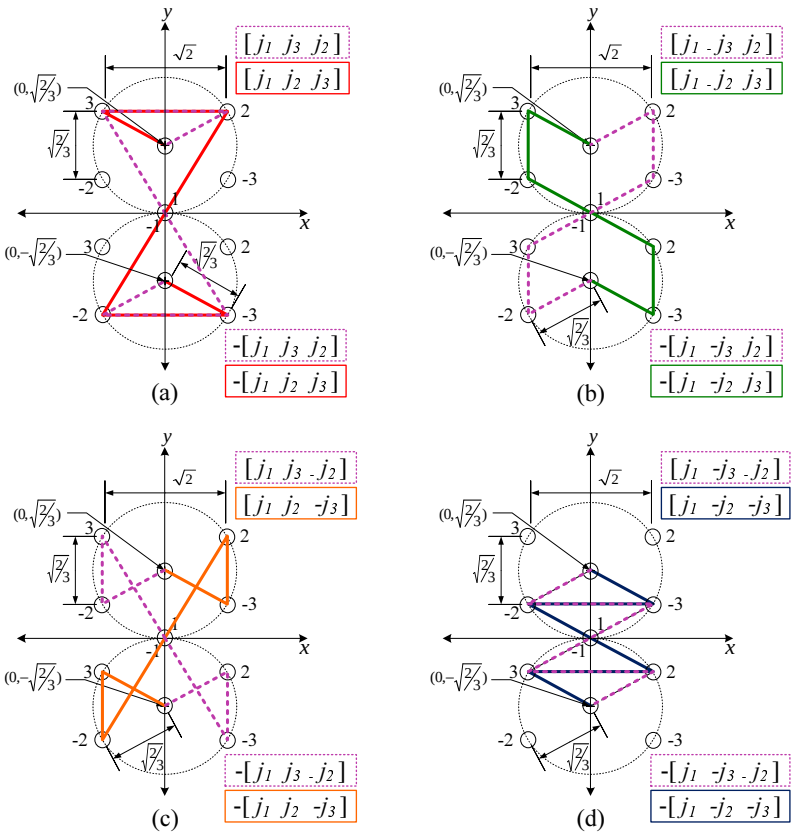
- Show that multiple different manipulators possess the same desired local properties described by a Jacobian (designed to be optimally fault-tolerant)
- Study optimally fault tolerant Jacobians for different task space dimensions
- Illustrate the difference between these manipulators in terms of their global fault tolerant properties

3DOF Planar Manipulators

- The total possible Jacobians are 48 ($2^3 \times 3!$)
- All of these Jacobians corresponds to only 4 different manipulators
- Link lengths:

Robot	a_1	a_2	a_3
1	L_s	L_s	L_s
2	L_s	L_l	L_s
3	L_l	L_s	L_s
4	L_l	L_l	L_s

$L_s = \sqrt{2/3}$ and $L_l = \sqrt{2}$



Why 4?

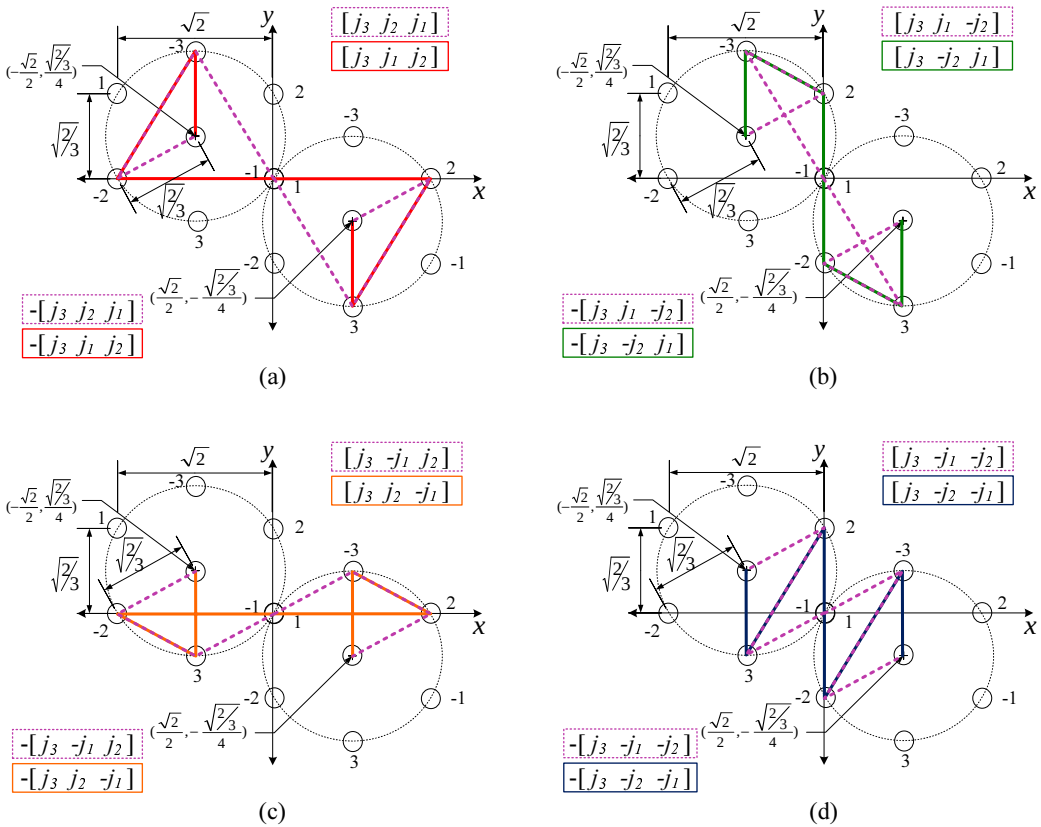
ARticulated Motion LABoratory

3DOF Planar Manipulators

- The total possible Jacobians are 48 ($2^3 \times 3!$)
- All of these Jacobians corresponds to only 4 different manipulators
- Link lengths:

Robot	a_1	a_2	a_3
1	L_s	L_s	L_s
2	L_s	L_l	L_s
3	L_l	L_s	L_s
4	L_l	L_l	L_s

$L_s = \sqrt{2/3}$ and $L_l = \sqrt{2}$



Why 4?

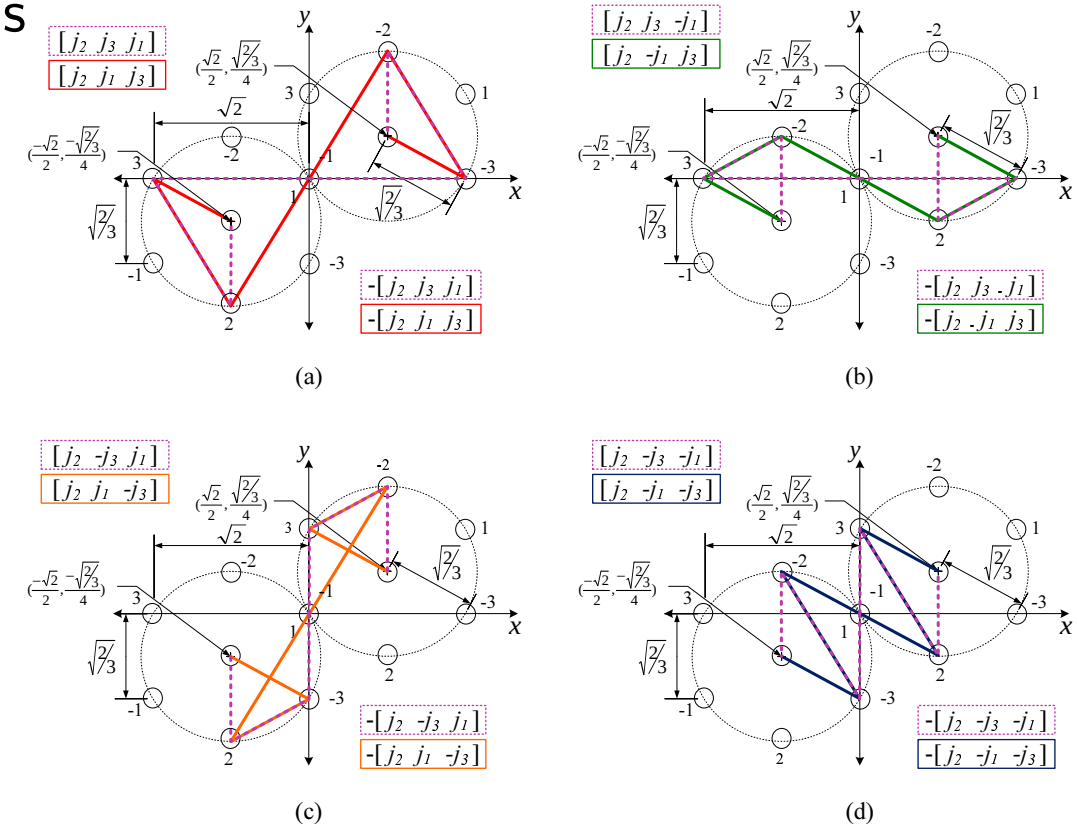
ARticulated Motion LABoratory

3DOF Planar Manipulators

- The total possible Jacobians are 48 ($2^3 \times 3!$)
- All of these Jacobians corresponds to only 4 different manipulators
- Link lengths:

Robot	a_1	a_2	a_3
1	L_s	L_s	L_s
2	L_s	L_l	L_s
3	L_l	L_s	L_s
4	L_l	L_l	L_s

$L_s = \sqrt{2/3}$ and $L_l = \sqrt{2}$



Why 4?

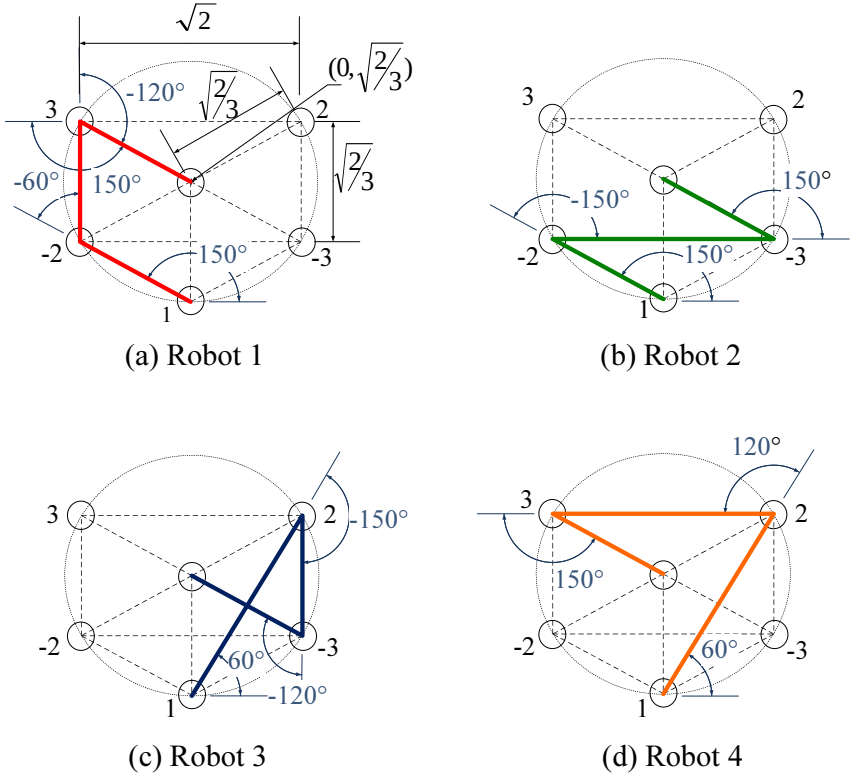
ARticulated Motion LABoratory

3DOF Planar Manipulators

- The total possible Jacobians are 48 ($2^3 \times 3!$)
- All of these Jacobians corresponds to only 4 different manipulators
- Link lengths:

Robot	a_1	a_2	a_3
1	L_s	L_s	L_s
2	L_s	L_l	L_s
3	L_l	L_s	L_s
4	L_l	L_l	L_s

$L_s = \sqrt{2/3}$ and $L_l = \sqrt{2}$



(a) Robot 1 (b) Robot 2 (c) Robot 3 (d) Robot 4

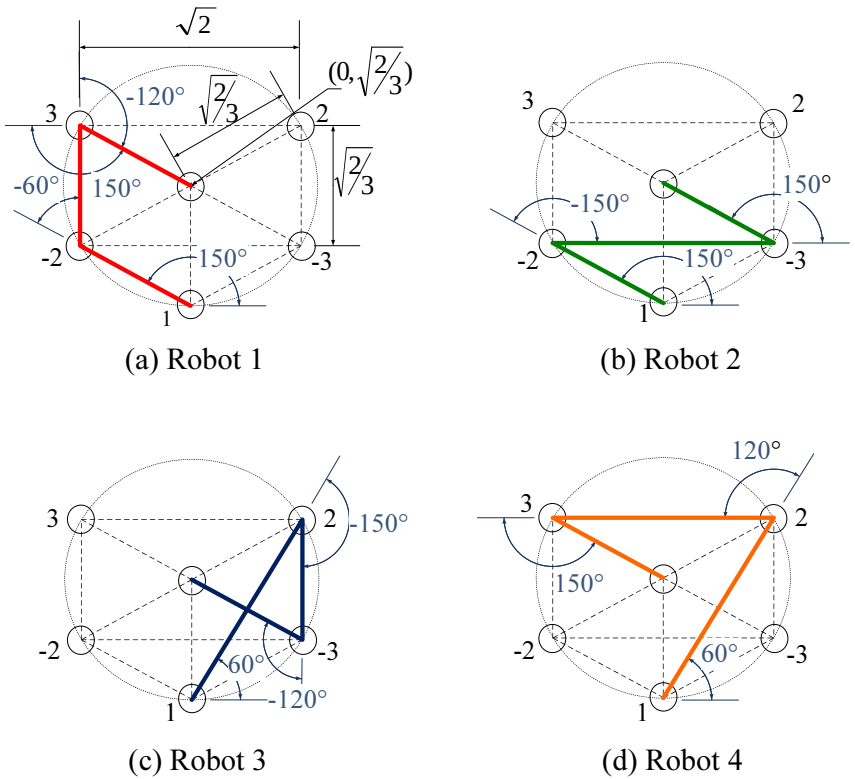
Why 4?

3DOF Planar Manipulators

- The total possible Jacobians are 48 ($2^3 \times 3!$)
- All of these Jacobians corresponds to only 4 different manipulators
- Link lengths:

Robot	a_1	a_2	a_3
1	L_s	L_s	L_s
2	L_s	L_l	L_s
3	L_l	L_s	L_s
4	L_l	L_l	L_s

$L_s = \sqrt{2/3}$ and $L_l = \sqrt{2}$



(a) Robot 1 (b) Robot 2 (c) Robot 3 (d) Robot 4

Why 4?

ARticulated Motion LABoratory

Different Robots Using the Gram Matrix

- Gram matrix is:

$$G = J^T J = I - NN^T \quad \text{when } J \text{ is isotropic}$$

  - N is a matrix consisting of orthonormal null vectors of J
  - $N = \hat{n}_J$ when $n - m = 1$
  - If J' is a rotation and/or reflection of J, then $J'^T J' = J^T J$; they both belong to the same manipulator

- Replacing $\hat{n}_J$ with $-\hat{n}_J$ doesn't affect G
- Only the 4 cases $\hat{n}_J = \sqrt{1/3}[1, \ \pm 1, \ \pm 1]$ determine four families of non-equivalent Jacobians
- With $\hat{n}_J = \sqrt{1/3}[1, \ 1, \ 1]$ the Gram matrix is:

$$G = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{bmatrix}$$

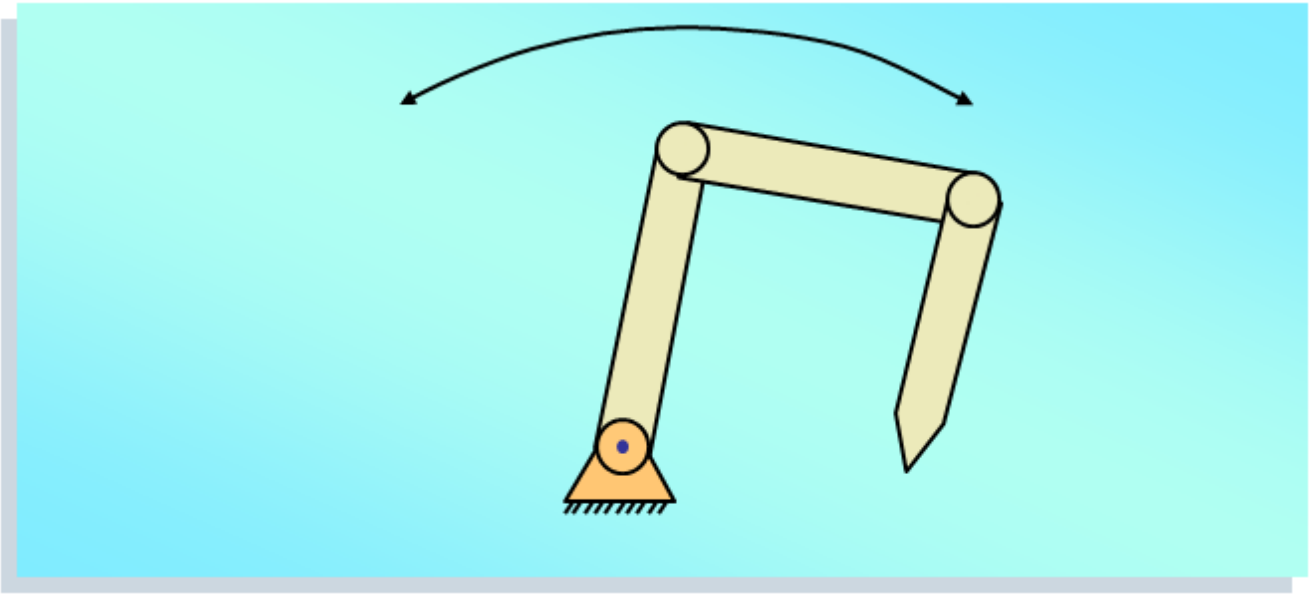
- $a_i^2 = g_{ii} + g_{i+1,i+1} - g_{i,i+1}$ (and $a_n^2 = g_{nn}$)

Global Workspace properties

- There different workspace boundaries:
  - $(2L_l + L_s)$
  - $(L_l + 2L_s)$
  - $3L_s$
- Determine the maximum value of $\mathcal{K}$ as a function of distance from the base
  - $\mathcal{K}$ is not a function of θ_1
  - There is a symmetry while rotating around the base
  - x-axis trajectory was selected

Finding Maximum $\mathcal{K}$

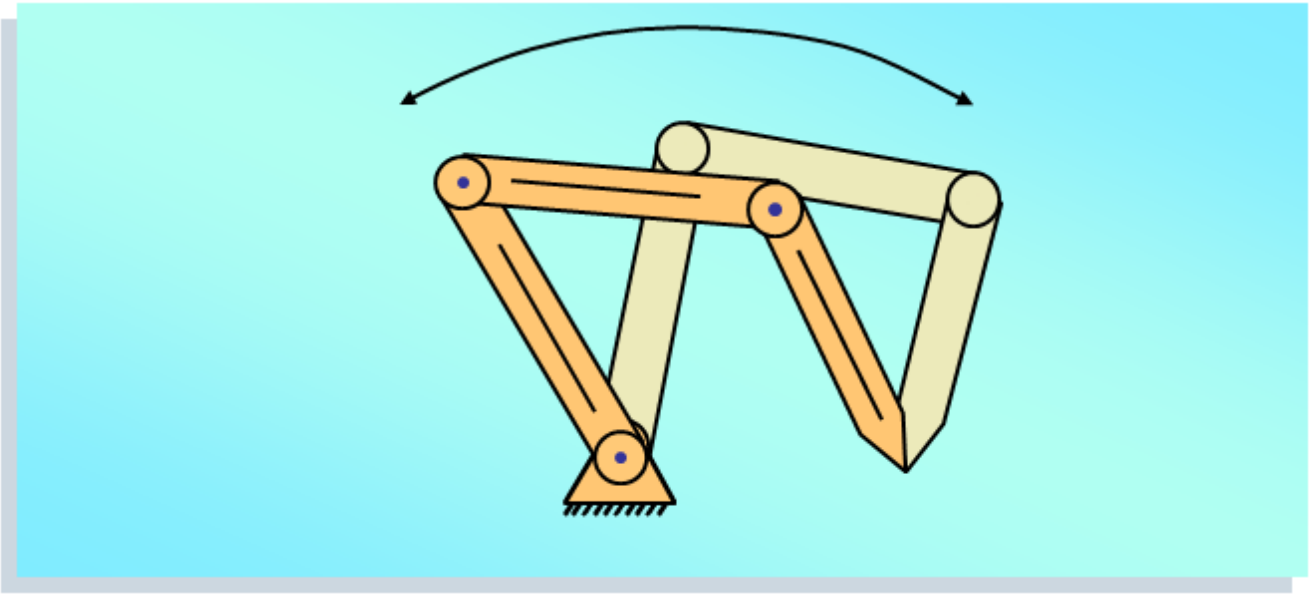
Use homogenous solution (Null Motion)



$$\dot{\theta} = J^{+}\dot{x} + \alpha n_J$$

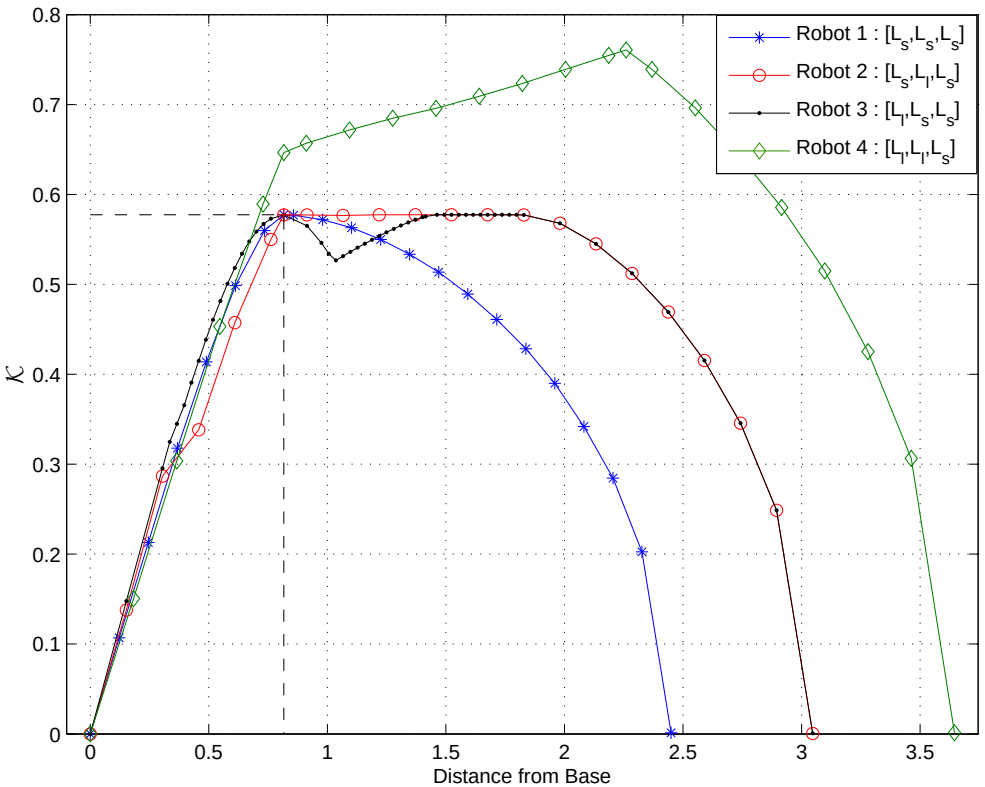
Finding Maximum $\mathcal{K}$

Use homogenous solution (Null Motion)



$$\dot{\theta} = J^{+}\dot{x} + \alpha n_J$$

Results



- Robot4 has a wide range of $\mathcal{K}$ larger than the optimal value
- Robot1 has only a peak at the optimal design point
- Robot2 has a flat region in the middle of its workspace.
- **Robot3** has a significant dip in the maximum value of $\mathcal{K}$ at a distance near one unit from the base before it returns to a comparable value to that of Robot2.

ARticulated Motion LABoratory

Cited Papers

K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, "An illustration of generating robots from optimal fault-tolerant Jacobians," 15th IASTED International Conference on Robotics and Applications, pp. 453–460, Cambridge, MA, Nov. 1–3, 2010.

K. M. Ben-Gharbia, R. G. Roberts, and A. A. Maciejewski, "Examples of planar robot kinematic designs from optimally fault-tolerant Jacobians," IEEE International Conference on Robotics and Automation , pp. 4710–4715, Shanghai, China, May 9–13, 2011.

K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, "A kinematic analysis and evaluation of planar robots designed from optimally fault-tolerant Jacobians," IEEE Trans. Robot., (under review).

Optimally 2×4 Fault Tolerant Jacobian (Two Failures)

- A 2×4 optimally fault tolerant Jacobian is:

$$J = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{\sqrt{2}} & \frac{1}{2} \end{bmatrix}.$$

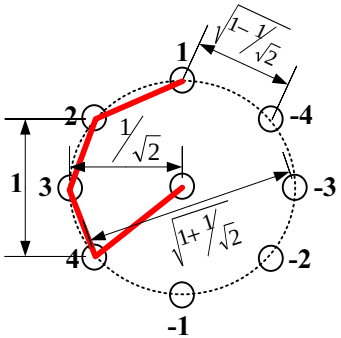
- N has two orthonormal vectors, and each row norm is $\sqrt{1/2}$
- The corresponding Gram matrix:

$$G = \begin{bmatrix} \frac{1}{2} & \frac{1}{2\sqrt{2}} & 0 & \frac{-1}{2\sqrt{2}} \\ \frac{1}{2\sqrt{2}} & \frac{1}{2} & \frac{1}{2\sqrt{2}} & 0 \\ 0 & \frac{1}{2\sqrt{2}} & \frac{1}{2} & \frac{1}{2\sqrt{2}} \\ \frac{-1}{2\sqrt{2}} & 0 & \frac{1}{2\sqrt{2}} & \frac{1}{2} \end{bmatrix}.$$

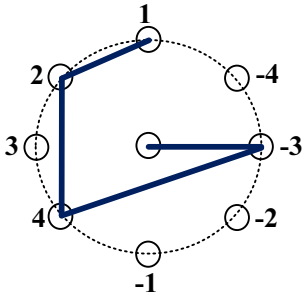
- The superdiagonal can be exactly one of three forms: $(\pm\frac{1}{2\sqrt{2}}, \pm\frac{1}{2\sqrt{2}}, \pm\frac{1}{2\sqrt{2}})$, $(\pm\frac{1}{2\sqrt{2}}, 0, \pm\frac{1}{2\sqrt{2}})$, or $(0, \pm\frac{1}{2\sqrt{2}}, 0)$
- Thus the total number of different manipulators is $2^3 + 2^2 + 2 = 14$

14 Optimal Fault Tolerant Planar 4R Manipulators

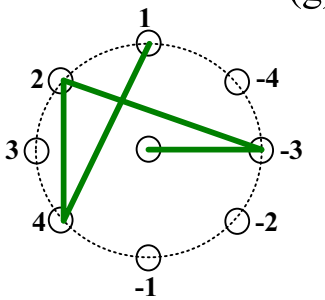
Robot	a_1	a_2	a_3	a_4
1/9	L_a/L_d	L_a	L_a	L_b
2/10	L_a/L_d	L_a	L_d	L_b
3/11	L_a/L_d	L_c	L_a	L_b
4/12	L_a/L_d	L_c	L_d	L_b
5/13	L_a/L_d	L_d	L_a	L_b
6/14	L_a/L_d	L_d	L_d	L_b
7	L_c	L_a	L_c	L_b
8	L_c	L_d	L_c	L_b



(a) Robot 1



(g) Robot 7



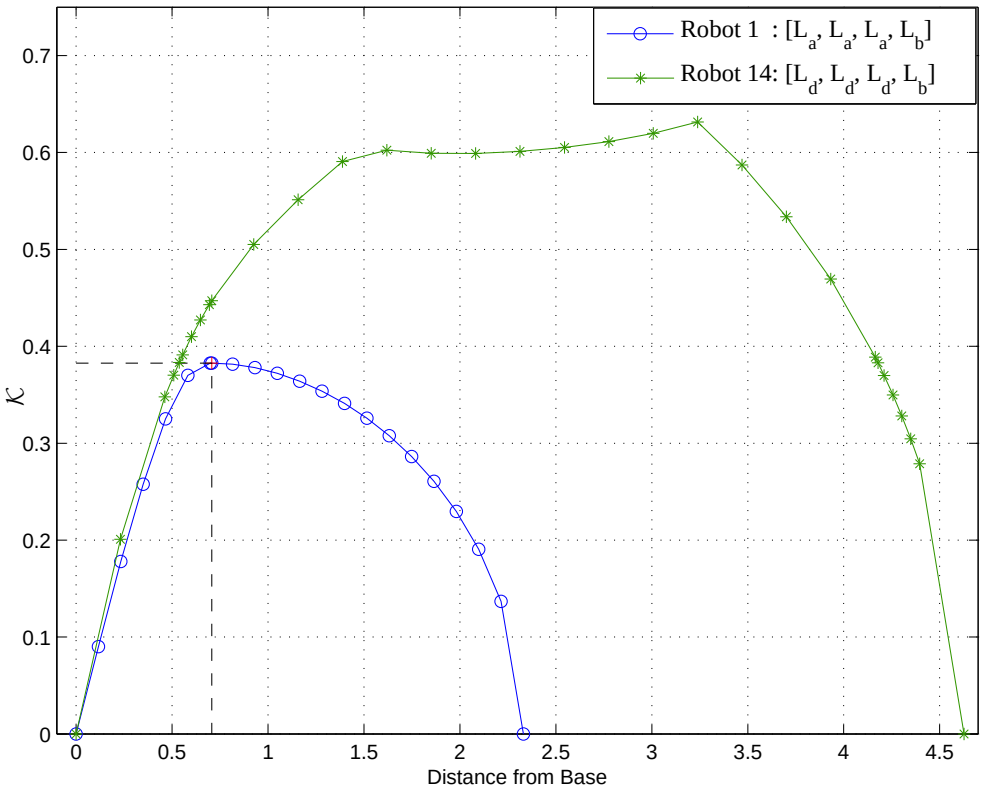
(m) Robot 13

$$L_a = \sqrt{1 - \frac{1}{\sqrt{2}}},\ L_b = \frac{1}{\sqrt{2}},\ L_c = 1,$$

and $L_d = \sqrt{1 + \frac{1}{\sqrt{2}}}$

ARticulated
Motion
LABoratory

Results



- Robot1 has only a peak at the optimal design point
- Robot14 has a wide range of $\mathcal{K}$ larger than the optimal value for 80% of its total workspace

Comparison

Robot	Reach $[m]$	Fault tolerant workspace [%]	Joint motion $[^\circ/m]$
1	2.33	0.00	–
2	2.79	15.14/(0.25)	107.3
3	3.10	9.99/(0.07)	117.0
4	3.10	22.16/(1.31)	98.6
5	3.10	26.96/(0.73)	372.8
6	3.25	47.76	94.1
7	3.56	46.21/(0.04)	190.2
8	3.56	49.06/(0.10)	72.2
9	3.86	57.68	79.7
10	3.86	58.02	109.8
11	3.86	58.78	71.0
12	4.01	73.03	76.5
13	4.32	77.42	74.1
14	4.63	80.23	70.7

Cited Papers

K. M. Ben-Gharbia, R. G. Roberts, and A. A. Maciejewski, "Examples of planar robot kinematic designs from optimally fault-tolerant Jacobians," IEEE International Conference on Robotics and Automation , pp. 4710–4715, Shanghai, China, May 9–13, 2011.

K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, "A kinematic analysis and evaluation of planar robots designed from optimally fault-tolerant Jacobians," IEEE Trans. Robot., (under review).

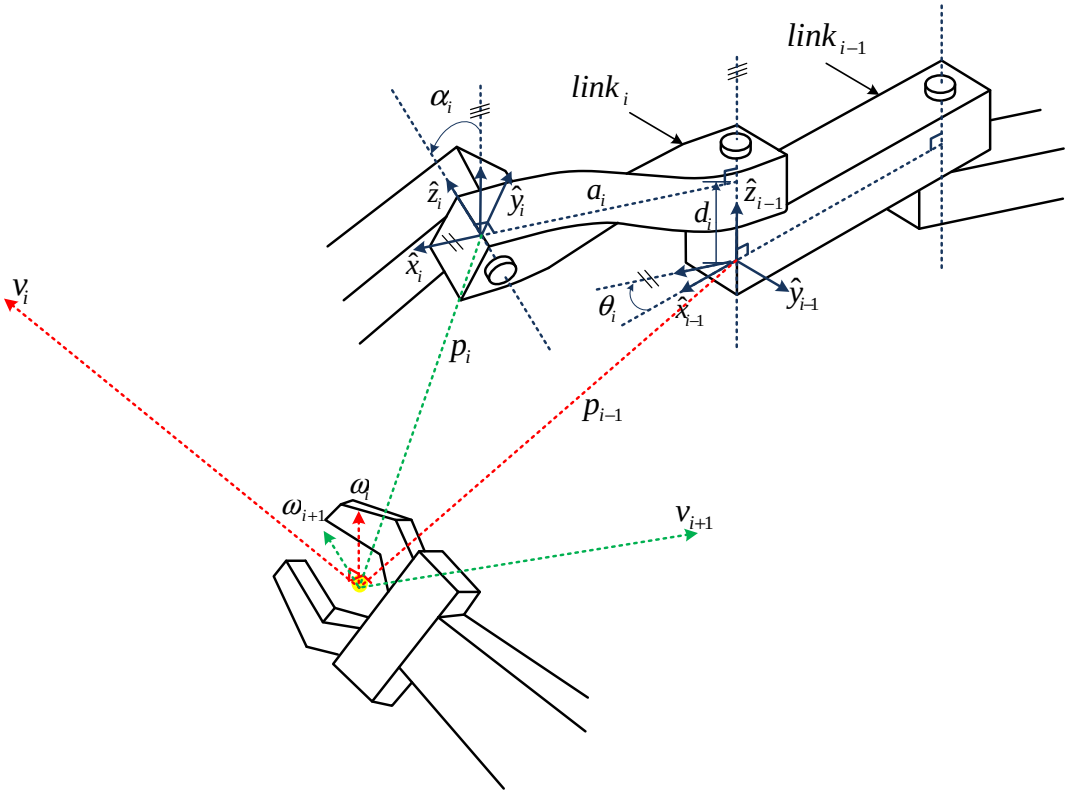
Spatial Manipulators

- The geometry becomes more complicated
- Denavit and Hartenberg (DH) parameters are used to describe the kinematic, parameters including link lengths
  - a_i: Link length
  - α_i: Link twist
  - d_i: Joint offset
  - θ_i: Joint value
- Any spatial Jacobian is represented by a $6 \times n$ matrix, s.t.

$$j_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} z_{i-1} \times p_{i-1} \\ z_{i-1} \end{bmatrix}$$

- Even for a spatial positioning task, we still need to define the orientational part of Jacobian
- Computing a maximum reach and a total workspace volume is harder than the case for a planar manipulator
- Moreover, it is harder to determine the DH parameters from a Jacobian

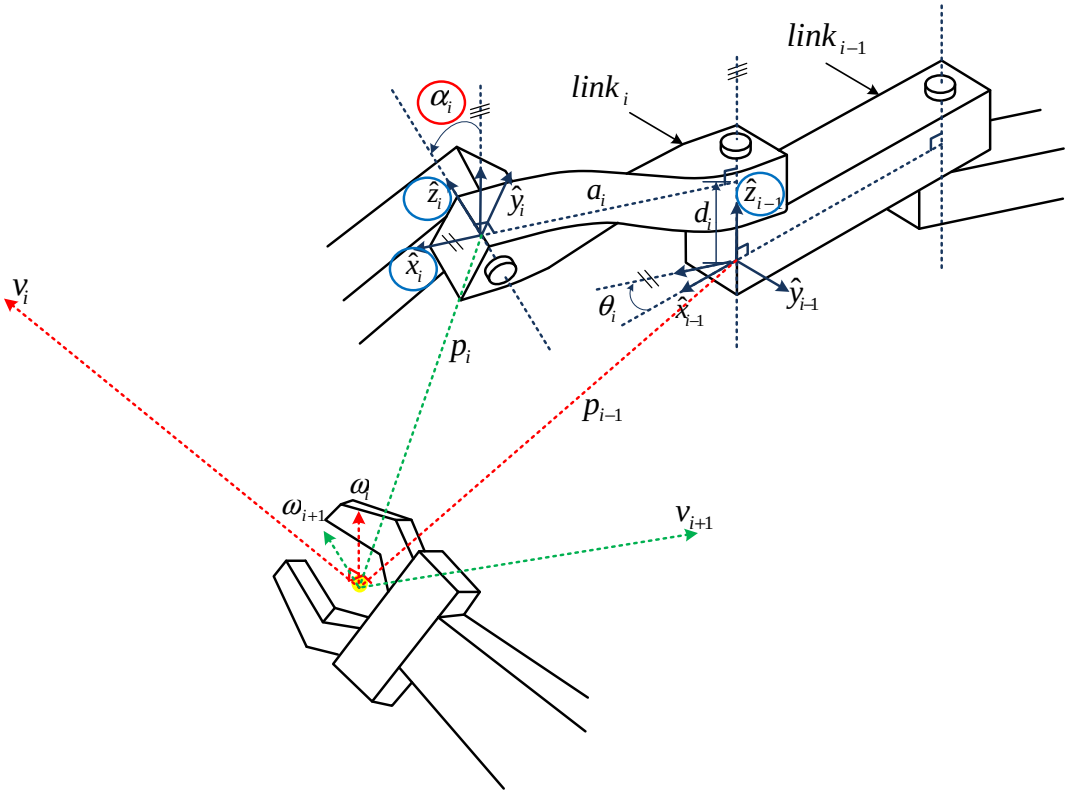
Calculating DH Parameters from a Given Desired Jacobian



- $\hat{x}_i = \pm \frac{\omega_i \times \omega_{i+1}}{\|\omega_i \times \omega_{i+1}\|}$
- $\hat{x}_i$ is pointing away from $\hat{z}_{i-1}$
- p_{i-1} can't be determined directly so
- Use p'_{i-1} instead: $p'_{i-1} = \omega_i \times v_i$
- Only the origins are required to determine d_i

$$j_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} z_{i-1} \times p_{i-1} \\ z_{i-1} \end{bmatrix}$$

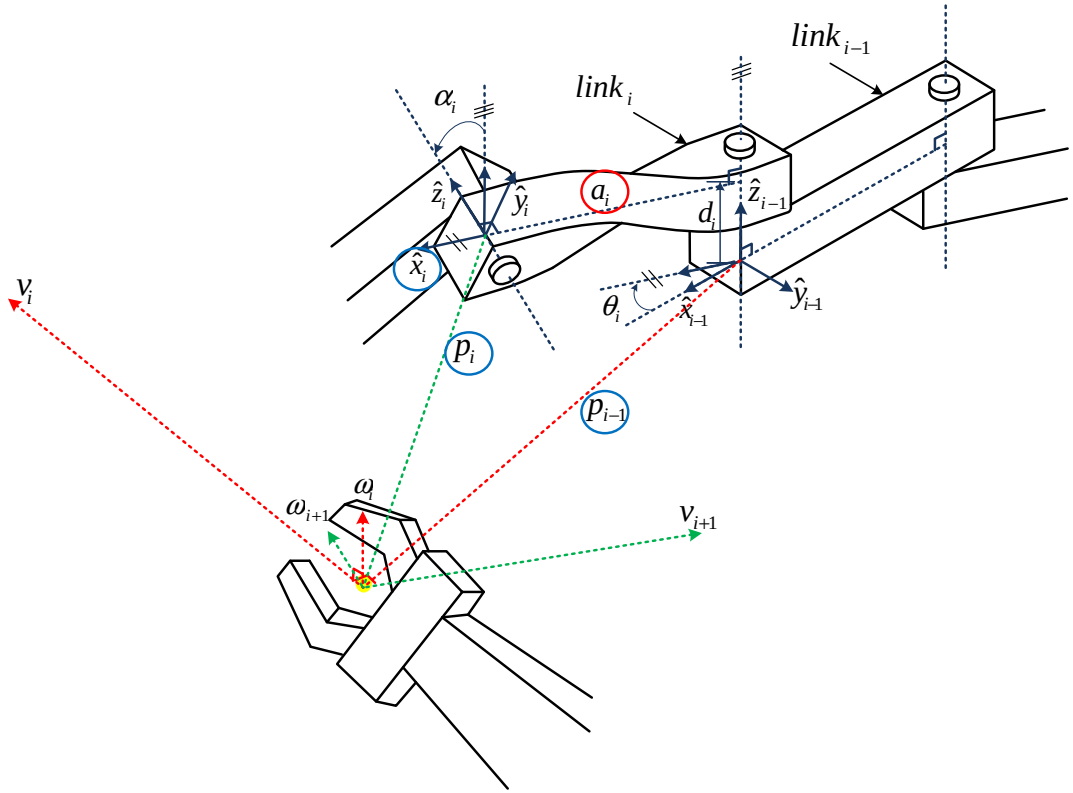
Calculating DH Parameters from a Given Desired Jacobian



- $\hat{x}_i = \pm \frac{\omega_i \times \omega_{i+1}}{\|\omega_i \times \omega_{i+1}\|}$
- $\hat{x}_i$ is pointing away from $\hat{z}_{i-1}$
- p_{i-1} can't be determined directly so
- Use p'_{i-1} instead: $p'_{i-1} = \omega_i \times v_i$
- Only the origins are required to determine d_i

$$j_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} z_{i-1} \times p_{i-1} \\ z_{i-1} \end{bmatrix}$$

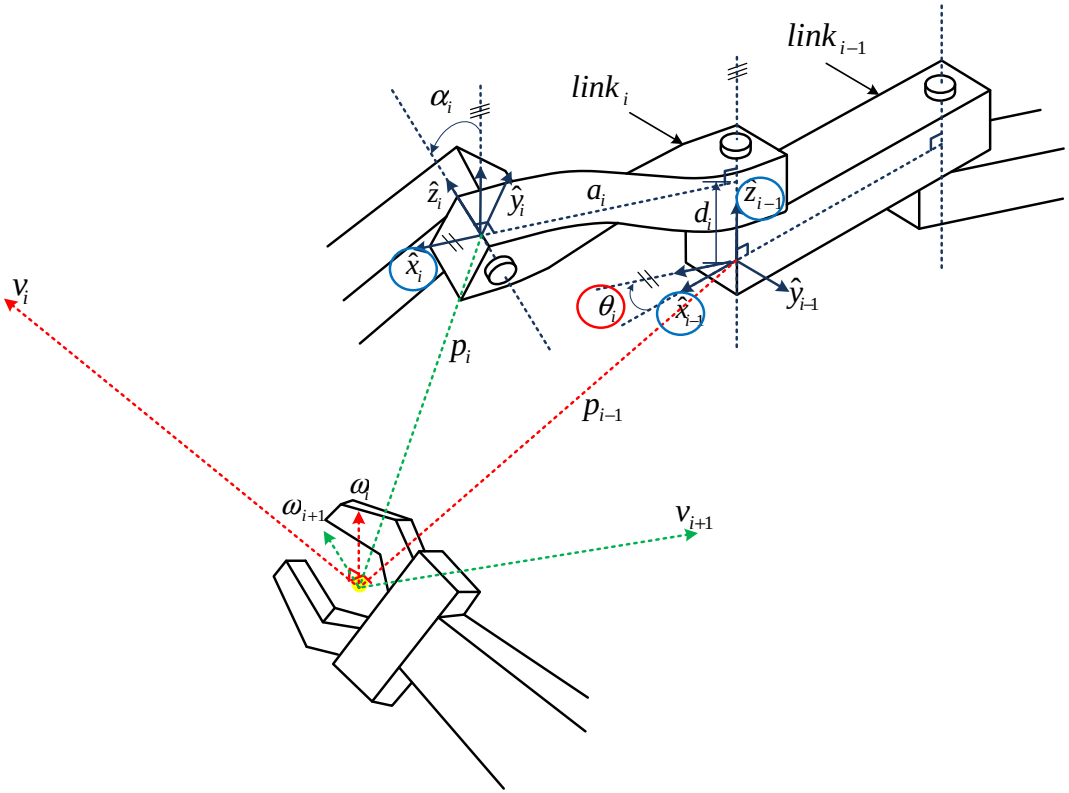
Calculating DH Parameters from a Given Desired Jacobian



- $\hat{x}_i = \pm \frac{\omega_i \times \omega_{i+1}}{\|\omega_i \times \omega_{i+1}\|}$
- $\hat{x}_i$ is pointing away from $\hat{z}_{i-1}$
- p_{i-1} can't be determined directly so
- Use p'_{i-1} instead: $p'_{i-1} = \omega_i \times v_i$
- Only the origins are required to determine d_i

$$j_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} z_{i-1} \times p_{i-1} \\ z_{i-1} \end{bmatrix}$$

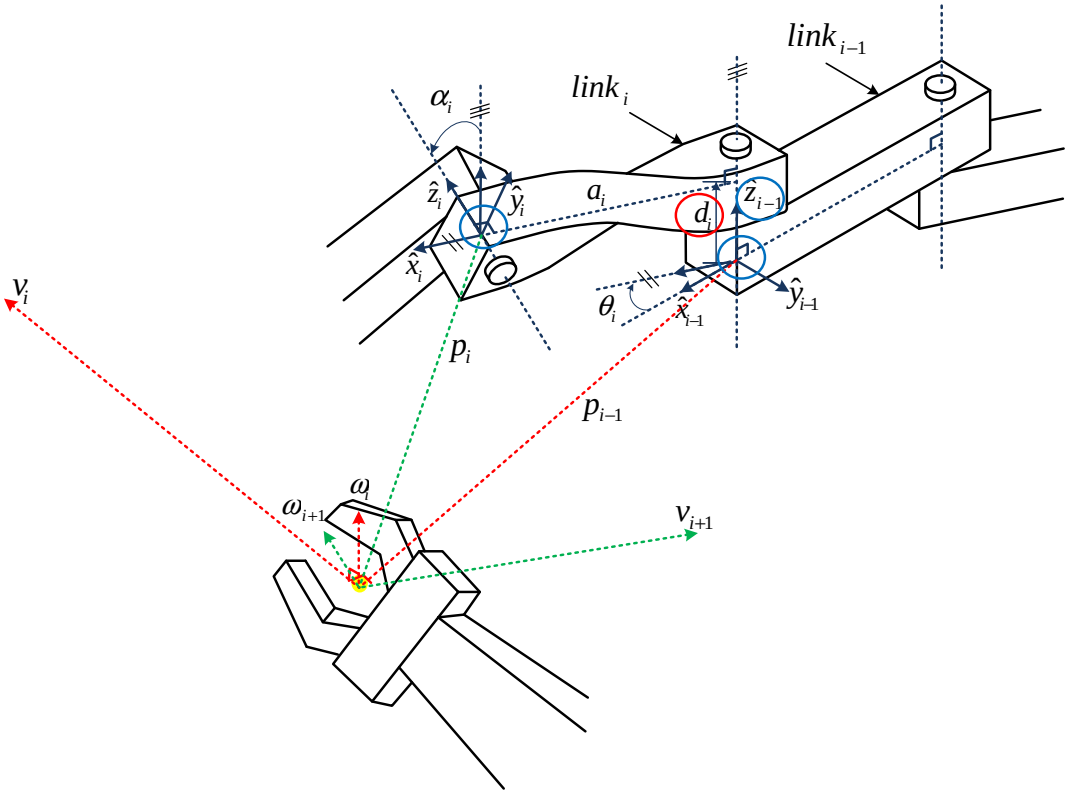
Calculating DH Parameters from a Given Desired Jacobian



- $\hat{x}_i = \pm \frac{\omega_i \times \omega_{i+1}}{\|\omega_i \times \omega_{i+1}\|}$
- $\hat{x}_i$ is pointing away from $\hat{z}_{i-1}$
- p_{i-1} can't be determined directly so
- Use p'_{i-1} instead: $p'_{i-1} = \omega_i \times v_i$
- Only the origins are required to determine d_i

$$j_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} z_{i-1} \times p_{i-1} \\ z_{i-1} \end{bmatrix}$$

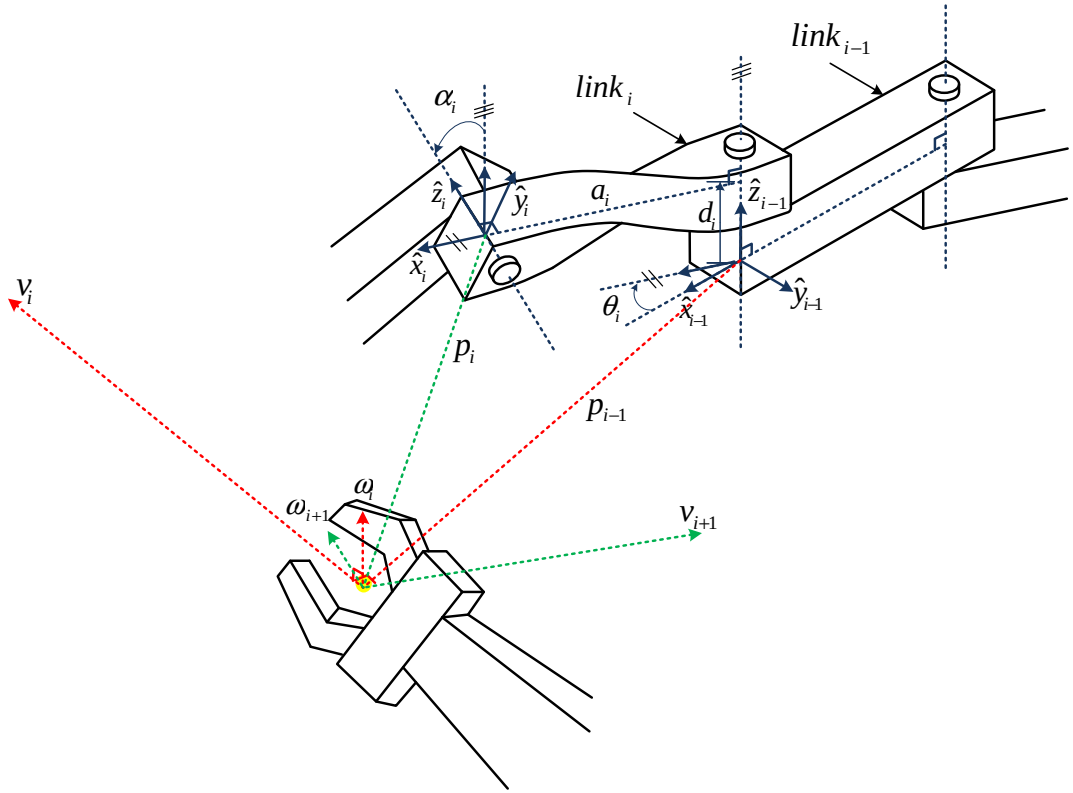
Calculating DH Parameters from a Given Desired Jacobian



- $\hat{x}_i = \pm \frac{\omega_i \times \omega_{i+1}}{\|\omega_i \times \omega_{i+1}\|}$
- $\hat{x}_i$ is pointing away from $\hat{z}_{i-1}$
- p_{i-1} can't be determined directly so
- Use p'_{i-1} instead: $p'_{i-1} = \omega_i \times v_i$
- Only the origins are required to determine d_i

$$j_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} z_{i-1} \times p_{i-1} \\ z_{i-1} \end{bmatrix}$$

Calculating DH Parameters from a Given Desired Jacobian



- $\hat{x}_i = \pm \frac{\omega_i \times \omega_{i+1}}{\|\omega_i \times \omega_{i+1}\|}$
- $\hat{x}_i$ is pointing away from $\hat{z}_{i-1}$
- p_{i-1} can't be determined directly so
- Use p'_{i-1} instead: $p'_{i-1} = \omega_i \times v_i$
- Only the origins are required to determine d_i

$$j_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} z_{i-1} \times p_{i-1} \\ z_{i-1} \end{bmatrix}$$

ARticulated Motion LABoratory

3×4 Optimally Fault Tolerant Jacobian

$$J = [\, j_1 \;\; j_2 \;\; j_3 \;\; j_4 \,] = \begin{bmatrix} -\sqrt{\frac{3}{4}} & \sqrt{\frac{1}{12}} & \sqrt{\frac{1}{12}} & \sqrt{\frac{1}{12}} \\ 0 & -\sqrt{\frac{2}{3}} & \sqrt{\frac{1}{6}} & \sqrt{\frac{1}{6}} \\ 0 & 0 & -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \end{bmatrix}$$

Null space equally distributed: $n_J = \frac{1}{2}[\, 1 \;\; 1 \;\; 1 \;\; 1 \,]$

Remaining dexterity: $\mathcal{K} = {}^{f}\sigma_3 = \frac{1}{2}$

Generating Physical Robots

- Rotation axes are not defined by only the 3×4 J
- By presenting J as

$$J_{6\times 4} = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix}$$

There is some freedom in selecting $J_\omega \implies$ thus, having different robots

Characterizing All 6×4 Fault Tolerant Jacobians

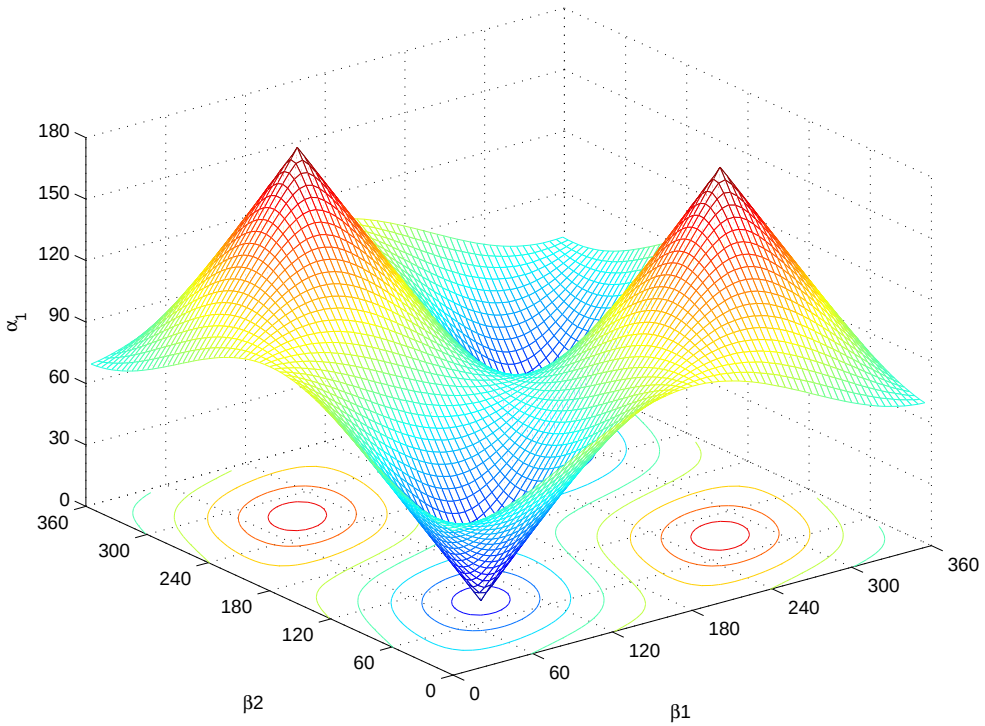
- Each $j_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}$ has
  - ω_i is orthogonal to v_i
  - ω_i is normalized
  - $\implies \omega_i = f(\beta_i)$ s.t.

$$\omega_1 = \begin{bmatrix} 0 \\ \cos(\beta_1) \\ \sin(\beta_1) \end{bmatrix}, \omega_2 = \begin{bmatrix} \frac{2\sqrt{2}}{3}\cos(\beta_2) \\ \frac{1}{3}\cos(\beta_2) \\ \sin(\beta_2) \end{bmatrix}, \omega_3 = \begin{bmatrix} \frac{2\sqrt{2}}{3}\sin(\beta_3) \\ -\left(\frac{\sqrt{3}}{2}\cos(\beta_3) + \frac{1}{6}\sin(\beta_3)\right) \\ -\frac{1}{2}\cos(\beta_3) + \frac{\sqrt{3}}{6}\sin(\beta_3) \end{bmatrix},$$

$$\omega_4 = \begin{bmatrix} \frac{2\sqrt{2}}{3}\sin(\beta_4) \\ -\left(\frac{\sqrt{3}}{2}\cos(\beta_4) + \frac{1}{6}\sin(\beta_4)\right) \\ -\left(-\frac{1}{2}\cos(\beta_4) + \frac{\sqrt{3}}{6}\sin(\beta_4)\right) \end{bmatrix}$$

- $0^\circ \leq \beta_i \leq 360^\circ$
- All rotate within a circle centered at origin

The Relationship Between DH Parameters and β_i's



- DH parameters are parameterized as:

$$\begin{aligned} \alpha_i &= f_{\alpha_i}(\beta_i, \beta_{i+1}) \\ a_i &= f_{a_i}(\beta_i, \beta_{i+1}) \\ \theta_i &= f_{\theta_i}(\beta_{i-1}, \beta_i, \beta_{i+1}) \\ d_i &= f_{d_i}(\beta_{i-1}, \beta_i, \beta_{i+1}) \end{aligned}$$

- Because the 5th coordinate frame (tool) is arbitrary,

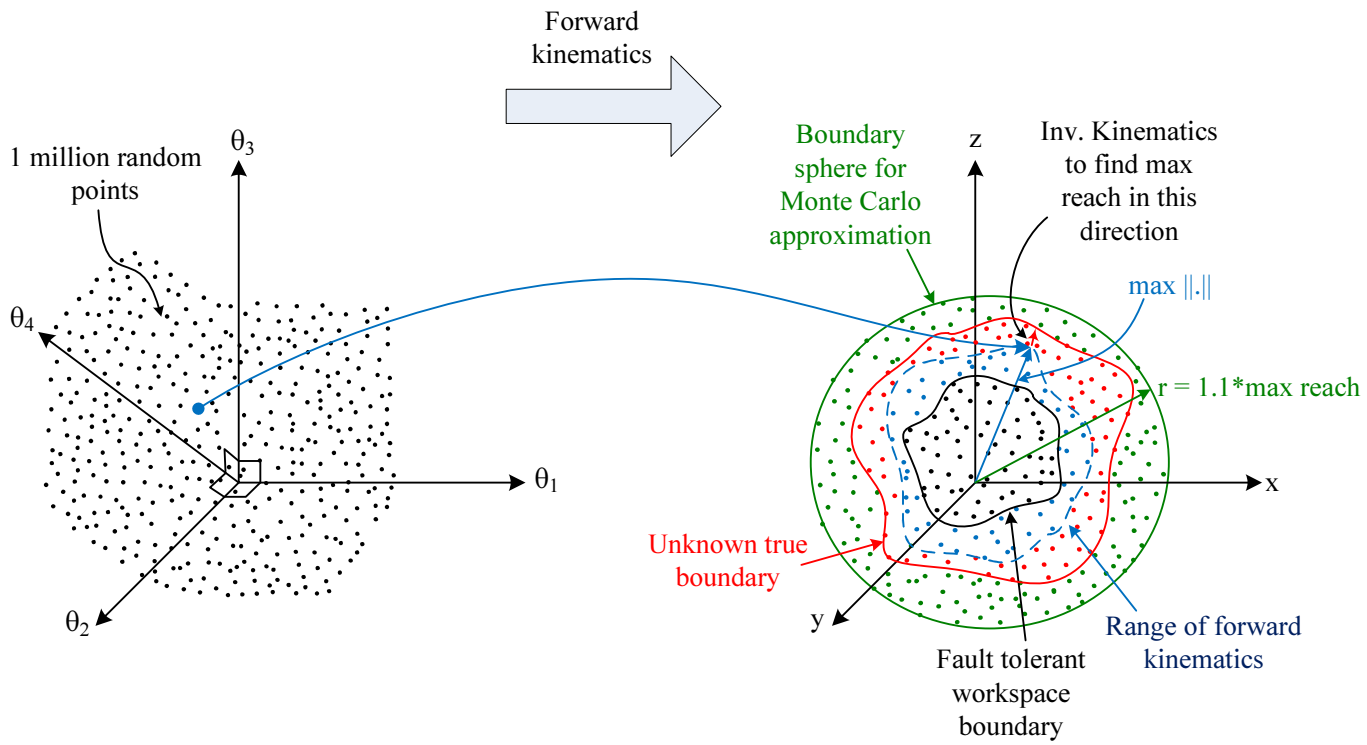
$$\begin{aligned} \alpha_4 &= 0 \\ a_4 &= \sqrt{3}/2 \\ \theta_4 &= f_{\theta_4}(\beta_3, \beta_4) \\ d_4 &= f_{d_4}(\beta_3, \beta_4) \end{aligned}$$

- $\theta_1 = d_1 = 0$ arbitrarily

Global Measurements

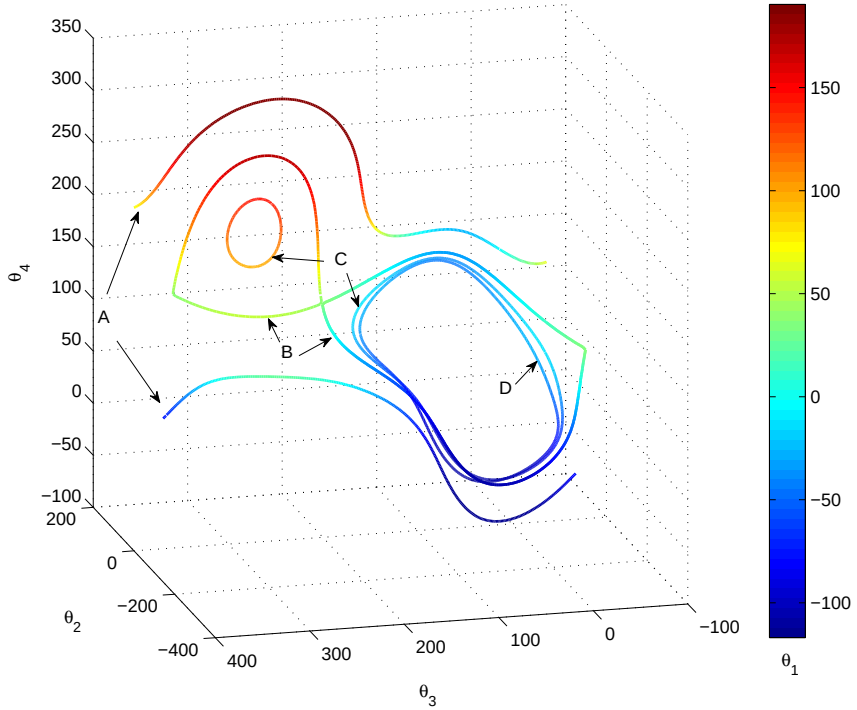
- Maximum reach
- The fraction of the total workspace that is fault tolerant, denoted $W_{\mathcal{K}}$ ($\mathcal{K} \geq \gamma \mathcal{K}_{opt}$)

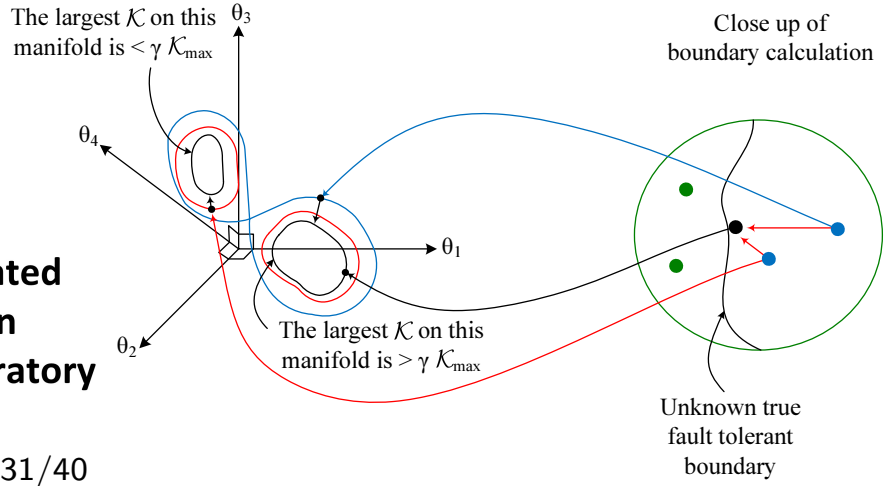
Monte Carlo Integration Algorithm



- One million uniformly distributed random samples are generated in the joint space
- The maximum reach is computed from the largest norm
- 10,000 uniformly distributed random samples within a sphere of radius of 110% of the maximum reach are used
- $W_{\mathcal{K}} = n_f / n_r$

Multiple Self-Motion Manifolds

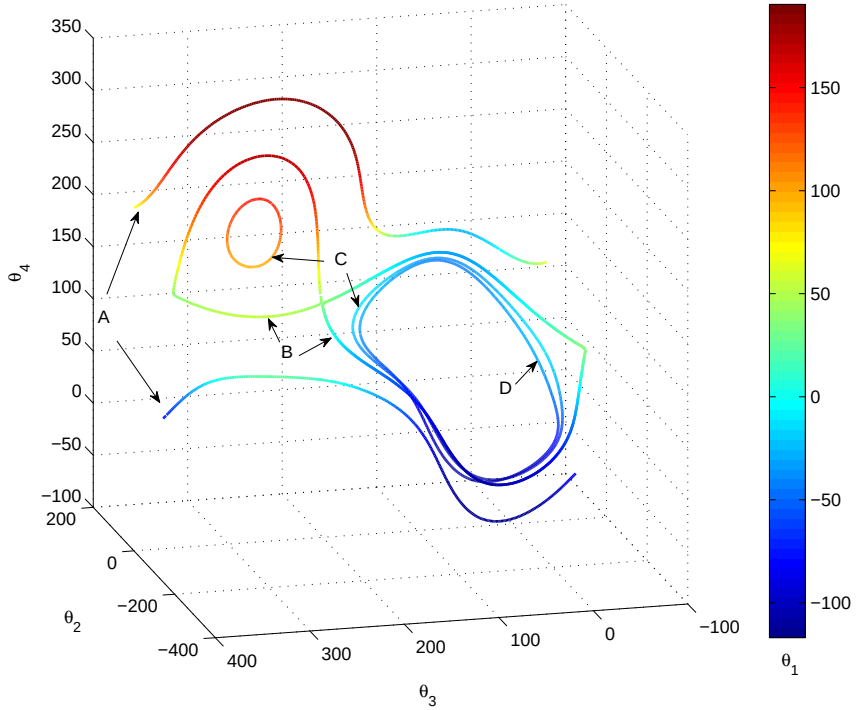


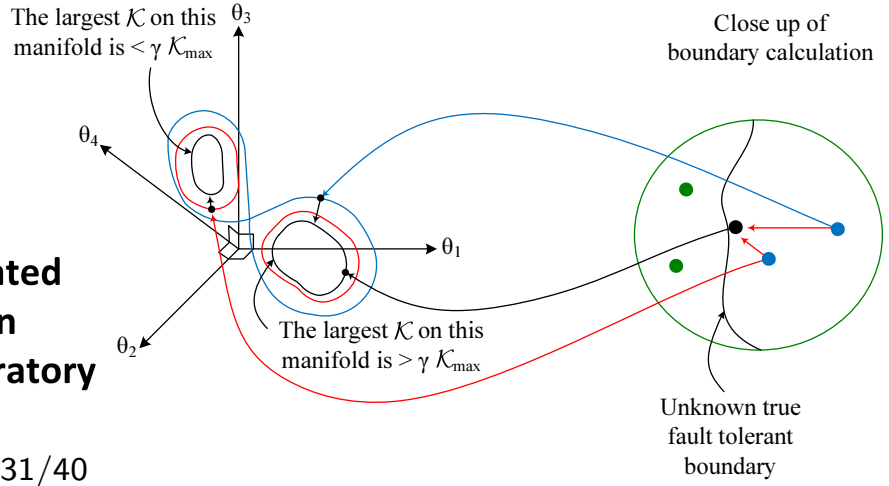


- Some of the points have multiple self-motion manifolds
- On one manifold, $\mathcal{K} \geq \gamma\mathcal{K}_{opt}$
- Missing being on that manifold would fail the test of having this point inside the fault tolerant workspace
- 5 joint configurations (whose locations are close to the point) are selected to increase the probability of $\mathcal{K} \geq \gamma\mathcal{K}_{opt}$
- Computationally expensive, i.e., 10-30 minutes/robot

ARticulated Motion LABoratory

Multiple Self-Motion Manifolds





- Some of the points have multiple self-motion manifolds
- On one manifold, $\mathcal{K} \geq \gamma\mathcal{K}_{opt}$
- Missing being on that manifold would fail the test of having this point inside the fault tolerant workspace
- 5 joint configurations (whose locations are close to the point) are selected to increase the probability of $\mathcal{K} \geq \gamma\mathcal{K}_{opt}$
- Computationally expensive, i.e., 10-30 minutes/robot

ARticulated Motion LABoratory

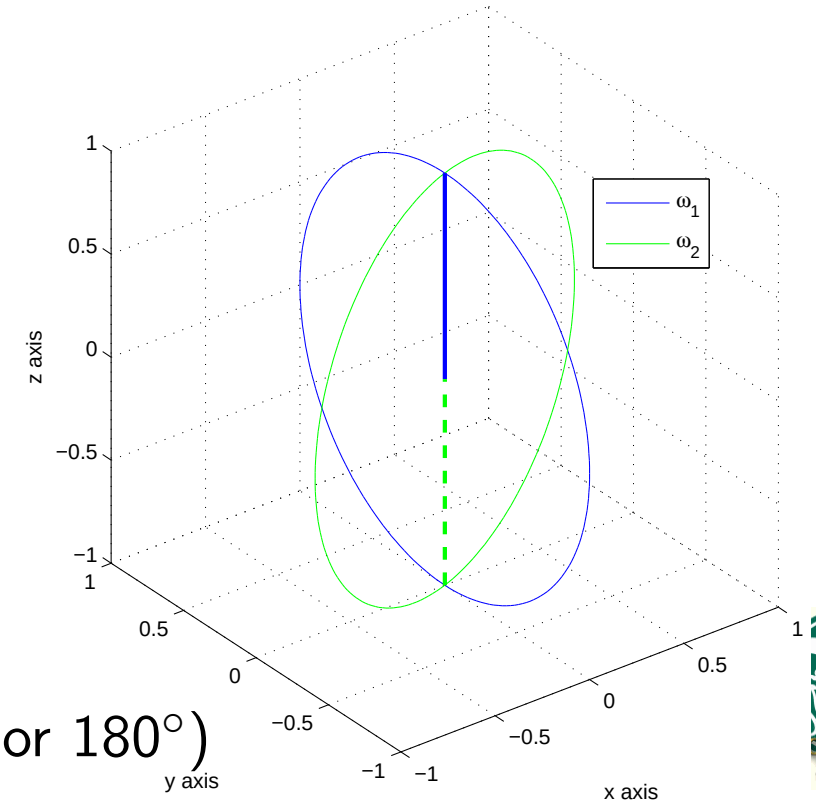
Examples of Manipulators with Common Link Twist Parameters

- Setting α_i's to $\pm 90°$, $0°$, or $180°$ is common in many commercial manipulators
- Recall that the parameter α_i is defined as the angle between the rotation axes of joints i and $i+1$, which is the same as ω_i and ω_{i+1}

Link Twist $\alpha_i = 0°$ or $180°$

- $\omega_i \cdot \omega_{i+1} = 1$ when $\alpha_i = 0°$
- $\omega_i \cdot \omega_{i+1} = -1$ when $\alpha_i = 180°$
- This yields to discrete value of β_i and β_{i+1}

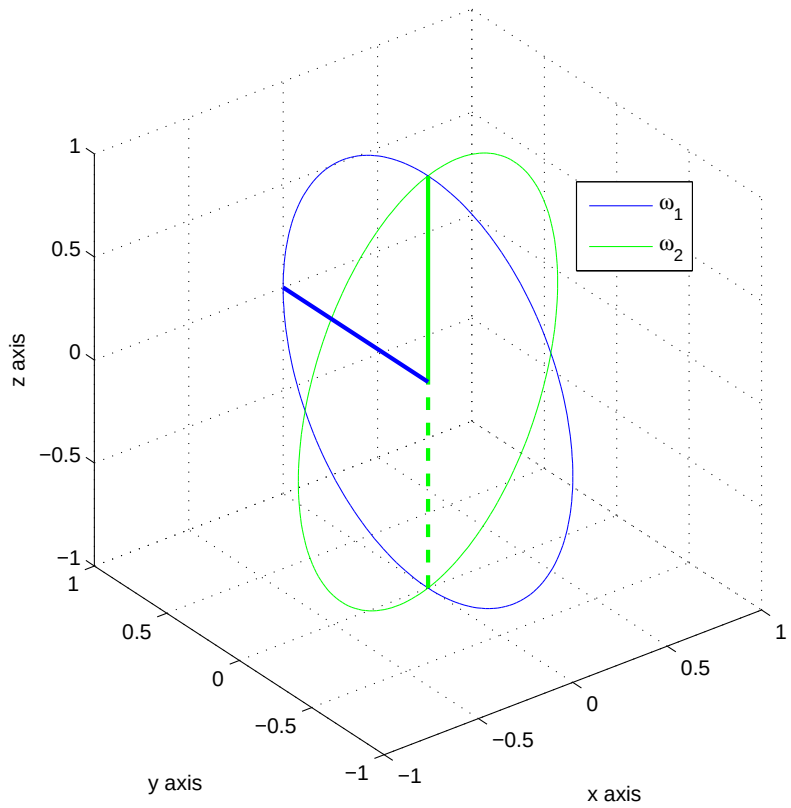
i	α_i [degrees]	$(\beta_i,\ \beta_{i+1})$ [degrees]
1	0	(90, 90)
	180	(90, 270)
2	0	(30, 120), (210, 300)
	180	(30, 300), (210, 120)
3	0	(60, 60), (240, 240)
	180	(60, 240), (240, 60)



- α_i and α_{i+1} can't be both $0°$ (or $180°$)

Link Twist $\alpha_i = \pm 90°$

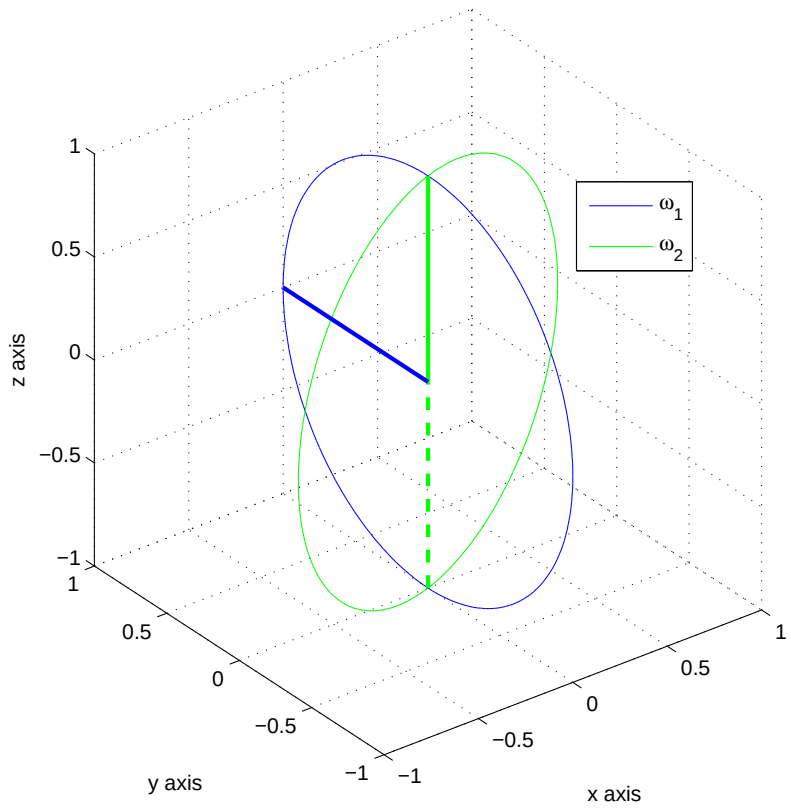
- $\omega_i \cdot \omega_{i+1} = 0$
- $0 \leq \beta_1 < 180°$ (when $180 \leq \beta_1 < 360°$ the resulting robots are mirrors), but $\beta_2 = f_{\beta_2}(\beta_1)$, $\beta_3 = f_{\beta_3}(\beta_2)$, and $\beta_4 = f_{\beta_4}(\beta_3)$



$\beta_1 = 0°$

Link Twist $\alpha_i = \pm 90°$

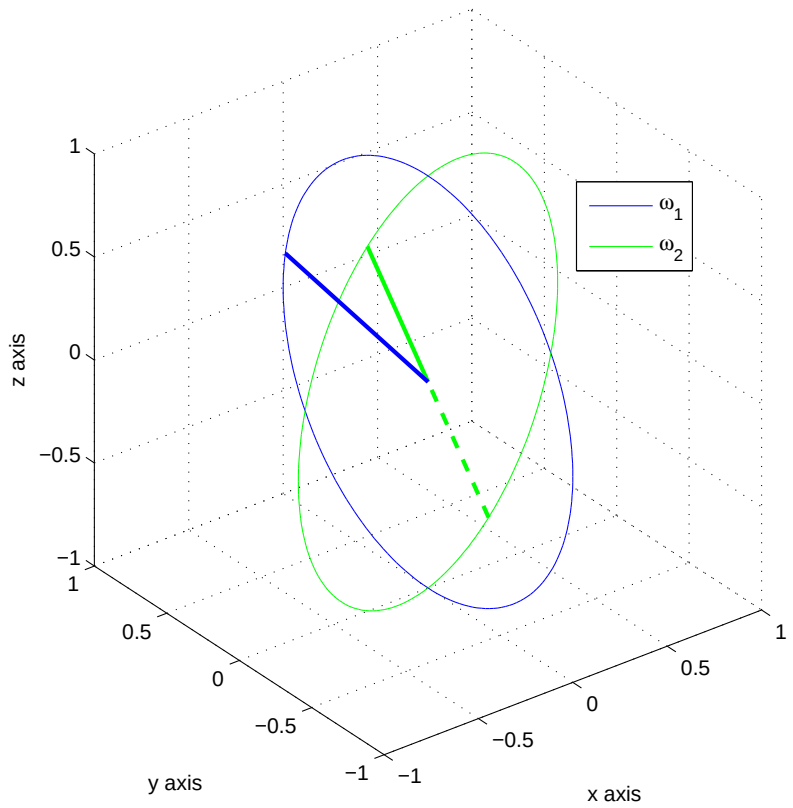
- $\omega_i \cdot \omega_{i+1} = 0$
- $0 \leq \beta_1 < 180°$ (when $180 \leq \beta_1 < 360°$ the resulting robots are mirrors), but $\beta_2 = f_{\beta_2}(\beta_1)$, $\beta_3 = f_{\beta_3}(\beta_2)$, and $\beta_4 = f_{\beta_4}(\beta_3)$



$\beta_1 = 0°$

Link Twist $\alpha_i = \pm 90°$

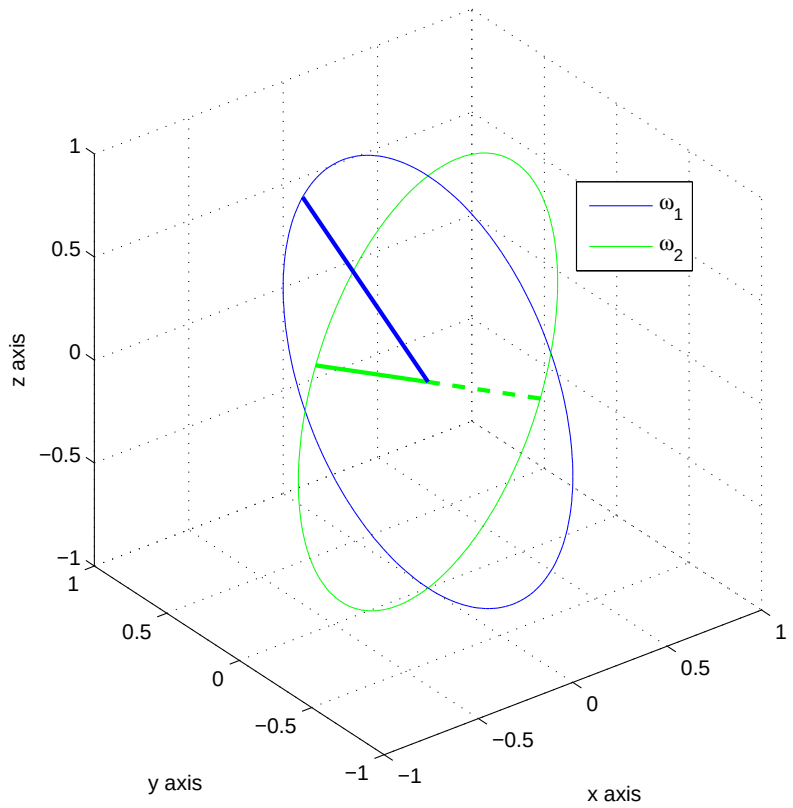
- $\omega_i \cdot \omega_{i+1} = 0$
- $0 \leq \beta_1 < 180°$ (when $180 \leq \beta_1 < 360°$ the resulting robots are mirrors), but $\beta_2 = f_{\beta_2}(\beta_1)$, $\beta_3 = f_{\beta_3}(\beta_2)$, and $\beta_4 = f_{\beta_4}(\beta_3)$



$\beta_1 = 10°$

Link Twist $\alpha_i = \pm 90°$

- $\omega_i \cdot \omega_{i+1} = 0$
- $0 \leq \beta_1 < 180°$ (when $180 \leq \beta_1 < 360°$ the resulting robots are mirrors), but $\beta_2 = f_{\beta_2}(\beta_1)$, $\beta_3 = f_{\beta_3}(\beta_2)$, and $\beta_4 = f_{\beta_4}(\beta_3)$



$\beta_1 = 30°$

Manipulator Categories

Robot Group	Relationship between joint axes $i-1$ and i $i=(1,2,3,4)$	Size of Group
1	$(\parallel, \parallel, \parallel, \parallel)$	0
2	$(\parallel, \parallel, \perp, \parallel)$	0
3	$(\perp, \parallel, \parallel, \parallel)$	0
4	$(\parallel, \perp, \parallel, \parallel)$	8
5	$(\parallel, \perp, \perp, \parallel)$	8
6	$(\perp, \parallel, \perp, \parallel)$	8
7	$(\perp, \perp, \parallel, \parallel)$	8
8	$(\perp, \perp, \perp, \parallel)$	∞

Best out of each group:

$W_{\mathcal{K}}$ [%]	max reach [m]
59	5.19
30	4.96
67	3.92
48	3.93
75	5.50

Manipulator Categories

Robot Group	Relationship between joint axes $i-1$ and i $i=(1,2,3,4)$	Size of Group
1	$(\parallel, \parallel, \parallel, \parallel)$	0
2	$(\parallel, \parallel, \perp, \parallel)$	0
3	$(\perp, \parallel, \parallel, \parallel)$	0
4	$(\parallel, \perp, \parallel, \parallel)$	8
5	$(\parallel, \perp, \perp, \parallel)$	8
6	$(\perp, \parallel, \perp, \parallel)$	8
7	$(\perp, \perp, \parallel, \parallel)$	8
8	$(\perp, \perp, \perp, \parallel)$	∞

Best out of each group:

$W_{\mathcal{K}}$ [%]	max reach [m]
59	5.19
30	4.96
67	3.92
48	3.93
75	5.50

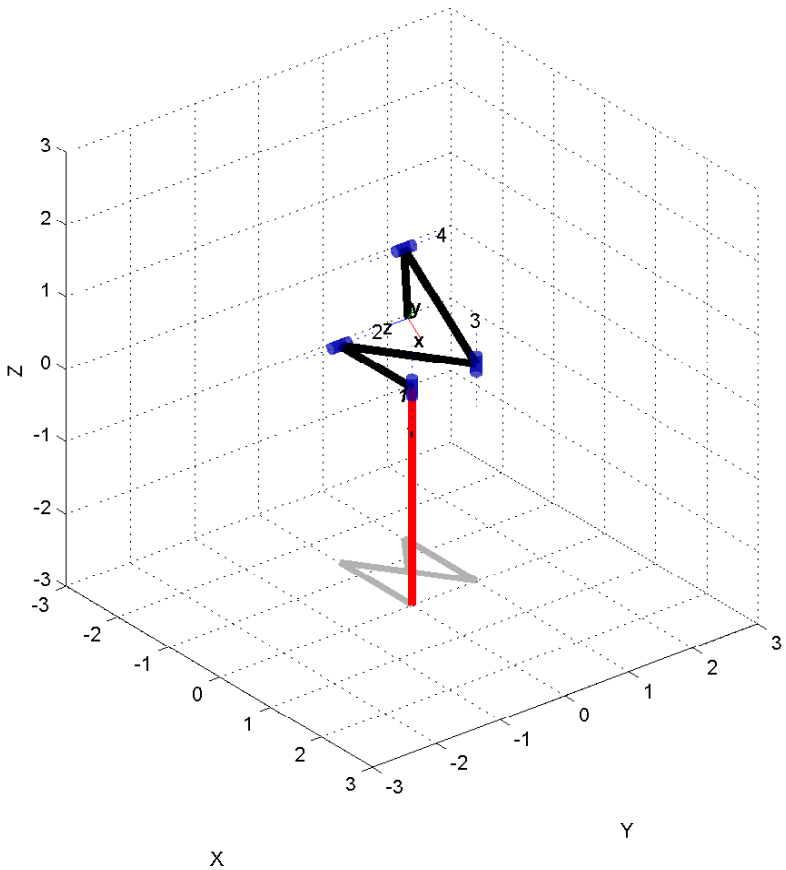
Manipulator Categories

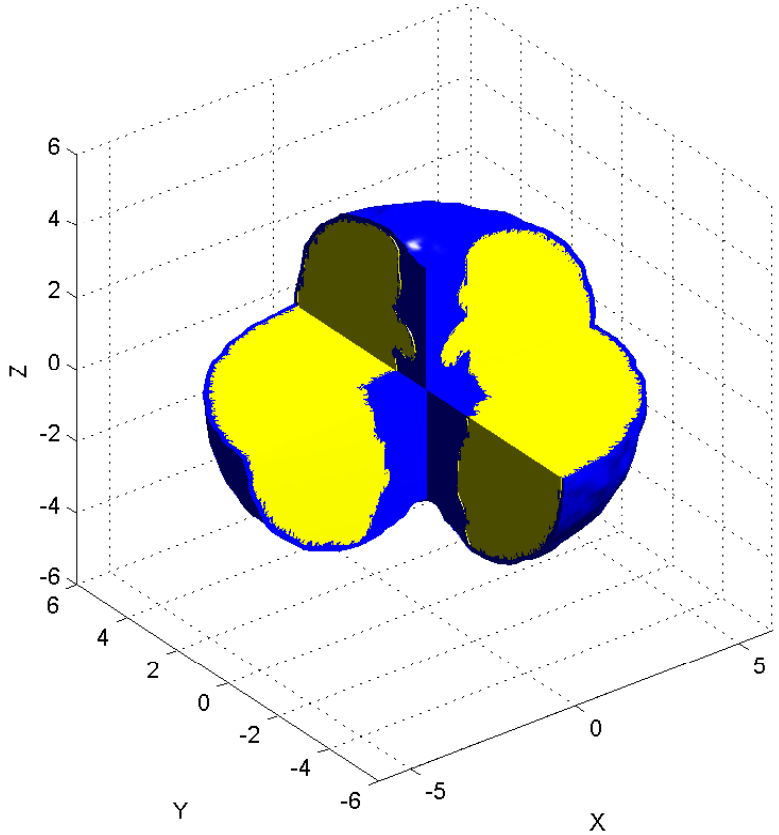
Robot Group	Relationship between joint axes $i-1$ and i $i=(1,2,3,4)$	Size of Group
1	$(\parallel, \parallel, \parallel, \parallel)$	0
2	$(\parallel, \parallel, \perp, \parallel)$	0
3	$(\perp, \parallel, \parallel, \parallel)$	0
4	$(\parallel, \perp, \parallel, \parallel)$	8
5	$(\parallel, \perp, \perp, \parallel)$	8
6	$(\perp, \parallel, \perp, \parallel)$	8
7	$(\perp, \perp, \parallel, \parallel)$	8
8	$(\perp, \perp, \perp, \parallel)$	∞

Best out of each group:

$W_{\mathcal{K}}$ [%]	max reach [m]
59	5.19
30	4.96
67	3.92
48	3.93
75	5.50

The Best Manipulator



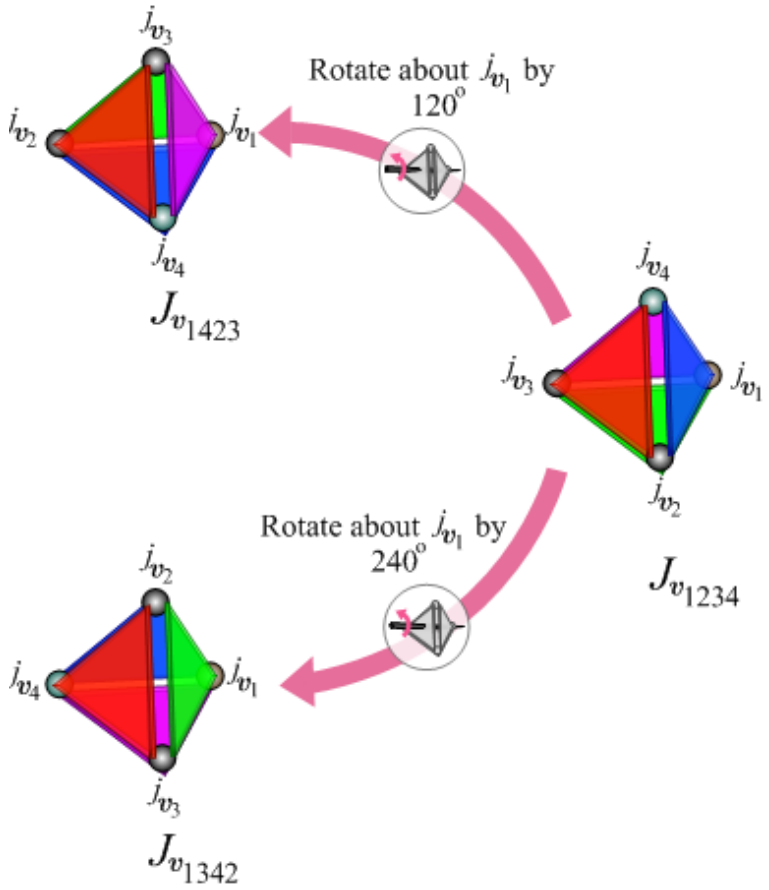


DH parameters:

i	α_i [degrees]	a_i [m]	d_i [m]	θ_i [degrees]
1	90	$\sqrt{2}$	0	0
2	-90	$\sqrt{2}$	1	180
3	90	$\sqrt{2}$	-1	180
4	0	$\sqrt{3}/2$	$1/2$	145

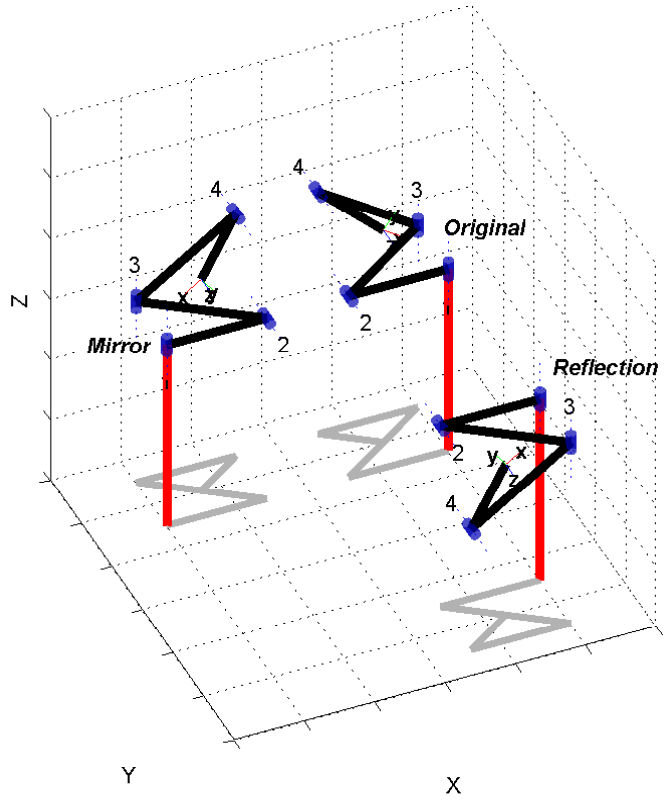
Columns Permutation and/or Sign Change Effect

- Sign change for very column is equivalent to reversing the direction of the corresponding joint axis
- A permutation is only equivalent to either a rotation or a reflection of the original Jacobian (A regular tetrahedron was useful to describe all permutations)



Columns Permutation and/or Sign Change Effect

- Sign change for very column is equivalent to reversing the direction of the corresponding joint axis
- A permutation is only equivalent to either a rotation or a reflection of the original Jacobian (A regular tetrahedron was useful to describe all permutations)



Cited Papers

K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, "An illustration of generating robots from optimal fault-tolerant Jacobians," 15th IASTED International Conference on Robotics and Applications, pp. 453–460, Cambridge, MA, Nov. 1–3, 2010.

K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, "Examples of spatial positioning redundant robotic manipulators that are optimally fault tolerant," IEEE International Conference on Systems, Man, and Cybernetics, pp. 1526–1531, Anchorage, Alaska, Oct. 9–12, 2011.

K. M. Ben-Gharbia, A. A. Maciejewski, and R. G. Roberts, "Kinematic Design of Redundant Robotic Manipulators for Spatial Positioning that are Optimally Fault Tolerant," IEEE Trans. Robot., (under review).

Conclusions

- Summary
  - There are multiple different robot designs that possess the same desired optimal Jacobian
  - Global properties are different
  - More optimal robot choices for designers
- Future directions
  - Study the case of optimally fault tolerant Jacobians for a six dimensional task space
  - Extend the Jacobian to DH parameters algorithm with any number of prismatic joints

Thanks!

?