Kinematic Design of Redundant Robotic Manipulators that are Optimally Fault Tolerant

Presented by:
Khaled M. Ben-Gharbia

January 15th, 2013
Why Fault Tolerance?

- Applications
  - hazardous waste cleanup
  - space/underwater exploration
  - anywhere failures are likely or intervention is costly

- Common failure mode
  - locked actuators
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:

  \[
  x = f(\theta)
  \]

  \[
  x_1 = a_1 \cos(\theta_1) + a_2 \cos(\theta_1 + \theta_2) + a_3 \cos(\theta_1 + \theta_2 + \theta_3)
  \]

  \[
  x_2 = a_1 \sin(\theta_1) + a_2 \sin(\theta_1 + \theta_2) + a_3 \sin(\theta_1 + \theta_2 + \theta_3)
  \]

  \[\Rightarrow\] taking the derivative w.r.t time

  \[\Rightarrow \dot{x} = J(\theta)\dot{\theta}\]

  - Geometrically

  \[J = \begin{bmatrix} j_1 & j_2 & j_3 \end{bmatrix} = \begin{bmatrix} z_1 \times p_1 & z_2 \times p_2 & z_3 \times p_3 \end{bmatrix}\]

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

- **Spatial manipulator**

  \[j_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}\]
Introduction  
Kinematics  
Fault Tolerance  
Goal  
3R Planar  
4R Planar  
4R Spatial  
Conclusions

Jacobian (J) Matrix

• 3R planar manipulator
  • Forward Kinematics:

  \[ x = f(\theta) \]

  \[ x_1 = a_1 \cos(\theta_1) + a_2 \cos(\theta_1 + \theta_2) + a_3 \cos(\theta_1 + \theta_2 + \theta_3) \]

  \[ x_2 = a_1 \sin(\theta_1) + a_2 \sin(\theta_1 + \theta_2) + a_3 \sin(\theta_1 + \theta_2 + \theta_3) \]

  \[ \implies \text{taking the derivative w.r.t time} \]

  \[ \dot{x} = J(\theta)\dot{\theta} \]

  • 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:
    \[
    x = f(\theta)
    \]
    \[
    x_1 = a_1 \cos(\theta_1) + a_2 \cos(\theta_1 + \theta_2) + a_3 \cos(\theta_1 + \theta_2 + \theta_3)
    \]
    \[
    x_2 = a_1 \sin(\theta_1) + a_2 \sin(\theta_1 + \theta_2) + a_3 \sin(\theta_1 + \theta_2 + \theta_3)
    \]
    \[
    \Rightarrow \text{ taking the derivative w.r.t time}
    \]
    \[
    \dot{x} = J(\theta)\dot{\theta}
    \]
  - 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 \( \sigma_m \) 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: \( K = \min_{f=1}^{n} f \sigma_m \)

Isotropic & Optimal \( J \): \( K = \sigma \sqrt{\frac{n-m}{n}} \) for all \( f \)
Isotropic and Optimally Fault Tolerant Jacobian

- equal $\sigma_i$’s
- equal $f \sigma_m$ for all $f$
- equal $\|j_i\|$ for all $i$

Example: $2 \times 3$ isotropic and optimally fault tolerant $J$:

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

Null space equally distributed: $n_J = \begin{bmatrix} \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} \end{bmatrix}$
Example: $2 \times 3$ Optimally Fault Tolerant Jacobian

Remaining dexterity: $\mathcal{K} = f \sigma_2 = \sqrt{\frac{1}{3}}$
Example: $2 \times 3$ Optimally Fault Tolerant Jacobian

Remaining dexterity: $\mathcal{K} = f \sigma_2 = \sqrt{\frac{1}{3}}$
Example: $2 \times 3$ 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
The total possible Jacobians are 48 ($2^3 \times 3!$)

All of these Jacobians correspond to only 4 different manipulators

Link lengths:

<table>
<thead>
<tr>
<th>Robot</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$L_s$</td>
<td>$L_s$</td>
<td>$L_s$</td>
</tr>
<tr>
<td>2</td>
<td>$L_s$</td>
<td>$L_l$</td>
<td>$L_s$</td>
</tr>
<tr>
<td>3</td>
<td>$L_l$</td>
<td>$L_s$</td>
<td>$L_s$</td>
</tr>
<tr>
<td>4</td>
<td>$L_l$</td>
<td>$L_l$</td>
<td>$L_s$</td>
</tr>
</tbody>
</table>

$L_s = \sqrt{2/3}$ and $L_l = \sqrt{2}$
The total possible Jacobians are 48 \((2^3 \times 3!\)\)

All of these Jacobians correspond to only 4 different manipulators

Link lengths:

<table>
<thead>
<tr>
<th>Robot</th>
<th>(a_1)</th>
<th>(a_2)</th>
<th>(a_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(L_s)</td>
<td>(L_s)</td>
<td>(L_s)</td>
</tr>
<tr>
<td>2</td>
<td>(L_s)</td>
<td>(L_l)</td>
<td>(L_s)</td>
</tr>
<tr>
<td>3</td>
<td>(L_l)</td>
<td>(L_s)</td>
<td>(L_s)</td>
</tr>
<tr>
<td>4</td>
<td>(L_l)</td>
<td>(L_l)</td>
<td>(L_s)</td>
</tr>
</tbody>
</table>

\(L_s = \sqrt{2/3}\) and \(L_l = \sqrt{2}\)
3DOF Planar Manipulators

- The total possible Jacobians are 48 ($2^3 \times 3!$)

- All of these Jacobians correspond to only 4 different manipulators

- Link lengths:

<table>
<thead>
<tr>
<th>Robot</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$L_S$</td>
<td>$L_S$</td>
<td>$L_S$</td>
</tr>
<tr>
<td>2</td>
<td>$L_S$</td>
<td>$L_l$</td>
<td>$L_S$</td>
</tr>
<tr>
<td>3</td>
<td>$L_l$</td>
<td>$L_S$</td>
<td>$L_S$</td>
</tr>
<tr>
<td>4</td>
<td>$L_l$</td>
<td>$L_l$</td>
<td>$L_S$</td>
</tr>
</tbody>
</table>

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

Why 4?
The total possible Jacobians are 48 \((2^3 \times 3!)\)

All of these Jacobians correspond to only 4 different manipulators

Link lengths:

<table>
<thead>
<tr>
<th>Robot</th>
<th>(a_1)</th>
<th>(a_2)</th>
<th>(a_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(L_s)</td>
<td>(L_s)</td>
<td>(L_s)</td>
</tr>
<tr>
<td>2</td>
<td>(L_s)</td>
<td>(L_l)</td>
<td>(L_s)</td>
</tr>
<tr>
<td>3</td>
<td>(L_l)</td>
<td>(L_s)</td>
<td>(L_s)</td>
</tr>
<tr>
<td>4</td>
<td>(L_l)</td>
<td>(L_l)</td>
<td>(L_s)</td>
</tr>
</tbody>
</table>

\(L_s = \sqrt{2/3}\) and \(L_l = \sqrt{2}\)

Why 4?
The total possible Jacobians are $48 \left(2^3 \times 3\right)!$)

All of these Jacobians corresponds to only 4 different manipulators

Link lengths:

<table>
<thead>
<tr>
<th>Robot</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$L_s$</td>
<td>$L_s$</td>
<td>$L_s$</td>
</tr>
<tr>
<td>2</td>
<td>$L_s$</td>
<td>$L_l$</td>
<td>$L_s$</td>
</tr>
<tr>
<td>3</td>
<td>$L_l$</td>
<td>$L_s$</td>
<td>$L_s$</td>
</tr>
<tr>
<td>4</td>
<td>$L_l$</td>
<td>$L_l$</td>
<td>$L_s$</td>
</tr>
</tbody>
</table>

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

Why 4?
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 \(K\) as a function of distance from the base
  - \(K\) is not a function of \(\theta_1\)
  - There is a symmetry while rotating around the base
  - \(x\)-axis trajectory was selected
Finding Maximum $\kappa$

Use homogenous solution (Null Motion)

$$\dot{\theta} = J^+ \dot{x} + \alpha n_J$$
Finding Maximum $\mathcal{K}$

Use homogenous solution (Null Motion)

\[ \dot{\theta} = J^+ \dot{x} + \alpha n_j \]
• **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.


Optimally $2 \times 4$ Fault Tolerant Jacobian (Two Failures)

- A $2 \times 4$ 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: \((\pm \frac{1}{2\sqrt{2}}, \pm \frac{1}{2\sqrt{2}}, \pm \frac{1}{2\sqrt{2}}), (\pm \frac{1}{2\sqrt{2}}, 0, \pm \frac{1}{2\sqrt{2}}), \text{ or } (0, \pm \frac{1}{2\sqrt{2}}, 0)\)
- Thus the total number of different manipulators is $2^3 + 2^2 + 2 = 14$
14 Optimal Fault Tolerant Planar 4R Manipulators

<table>
<thead>
<tr>
<th>Robot</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/9</td>
<td>$L_a/L_d$</td>
<td>$L_a$</td>
<td>$L_a$</td>
<td>$L_b$</td>
</tr>
<tr>
<td>2/10</td>
<td>$L_a/L_d$</td>
<td>$L_a$</td>
<td>$L_d$</td>
<td>$L_b$</td>
</tr>
<tr>
<td>3/11</td>
<td>$L_a/L_d$</td>
<td>$L_c$</td>
<td>$L_a$</td>
<td>$L_b$</td>
</tr>
<tr>
<td>4/12</td>
<td>$L_a/L_d$</td>
<td>$L_c$</td>
<td>$L_d$</td>
<td>$L_b$</td>
</tr>
<tr>
<td>5/13</td>
<td>$L_a/L_d$</td>
<td>$L_d$</td>
<td>$L_a$</td>
<td>$L_b$</td>
</tr>
<tr>
<td>6/14</td>
<td>$L_a/L_d$</td>
<td>$L_d$</td>
<td>$L_d$</td>
<td>$L_b$</td>
</tr>
<tr>
<td>7</td>
<td>$L_c$</td>
<td>$L_a$</td>
<td>$L_c$</td>
<td>$L_b$</td>
</tr>
<tr>
<td>8</td>
<td>$L_c$</td>
<td>$L_d$</td>
<td>$L_c$</td>
<td>$L_b$</td>
</tr>
</tbody>
</table>

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

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

- **Robot 1** has only a peak at the optimal design point.
- **Robot 14** has a wide range of $\mathcal{K}$ larger than the optimal value for 80% of its total workspace.
## Comparison

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


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
  - $\alpha_i$: Link twist
  - $d_i$: Joint offset
  - $\theta_i$: 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

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

\[ \hat{x}_i = \pm \frac{\omega_i \times \omega_{i+1}}{||\omega_i \times \omega_{i+1}||} \]

\[ \hat{x}_i \] 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 \( d_i \).
Calculating DH Parameters from a Given Desired Jacobian

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

- \( \hat{x}_i = \pm \frac{\omega_i \times \omega_{i+1}}{||\omega_i \times \omega_{i+1}||} \)
- \( \hat{x}_i \) 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 \( d_i \)
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 \( d_i \)

\[ j_i = \begin{bmatrix} \mathbf{v}_i \\ \mathbf{\omega}_i \end{bmatrix} = \begin{bmatrix} \mathbf{z}_{i-1} \times p_{i-1} \\ \mathbf{z}_{i-1} \end{bmatrix} \]
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

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

- \( \hat{x}_i = \pm \frac{\omega_i \times \omega_{i+1}}{||\omega_i \times \omega_{i+1}||} \)
- \( \hat{x}_i \) 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 \( d_i \)
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 \(\hat{z}_{i-1}\)

\(\rho_{i-1}\) 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 \(d_i\)

\[
j_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} z_{i-1} \times p_{i-1} \\ z_{i-1} \end{bmatrix}
\]
3 × 4 Optimally Fault Tolerant Jacobian

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

Null space equally distributed: \( n_J = \frac{1}{2} [1 \ 1 \ 1 \ 1] \)

Remaining dexterity: \( \mathcal{K} = f \sigma_3 = \frac{1}{2} \)
Rotation axes are not defined by only the $3 \times 4 J$

By presenting $J$ as

$$J_{6\times4} = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix}$$

There is some freedom in selecting $J_\omega \implies$ thus, having different robots
Characterizing All $6 \times 4$ Fault Tolerant Jacobians

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

$$
\begin{align*}
\omega_1 &= \begin{bmatrix} 0 \\ \cos(\beta_1) \\ \sin(\beta_1) \end{bmatrix}, \quad \\
\omega_2 &= \begin{bmatrix} \frac{2\sqrt{2}}{3} \cos(\beta_2) \\ \frac{1}{3} \cos(\beta_2) \\ \sin(\beta_2) \end{bmatrix}, \quad \\
\omega_3 &= \begin{bmatrix} \frac{2\sqrt{2}}{3} \sin(\beta_3) \\ -\left(\frac{\sqrt{3}}{2} \cos(\beta_3) + \frac{1}{6} \sin(\beta_3)\right) \\ -\frac{1}{2} \cos(\beta_3) + \frac{\sqrt{3}}{6} \sin(\beta_3) \end{bmatrix}, \\
\omega_4 &= \begin{bmatrix} \frac{2\sqrt{2}}{3} \sin(\beta_4) \\ -\left(\frac{\sqrt{3}}{2} \cos(\beta_4) + \frac{1}{6} \sin(\beta_4)\right) \\ -\left(-\frac{1}{2} \cos(\beta_4) + \frac{\sqrt{3}}{6} \sin(\beta_4)\right) \end{bmatrix}
\end{align*}
$$

$0^\circ \leq \beta_i \leq 360^\circ$
- All rotate within a circle centered at origin
The Relationship Between DH Parameters and $\beta_i$'s

- DH parameters are parameterized as:
  \[
  \begin{align*}
  \alpha_i &= f_{\alpha_i}(\beta_i, \beta_{i+1}) \\
  a_i &= f_{a_i}(\beta_i, \beta_{i+1}) \\
  \theta_i &= f_{\theta_i}(\beta_{i-1}, \beta_i, \beta_{i+1}) \\
  d_i &= f_{d_i}(\beta_{i-1}, \beta_i, \beta_{i+1})
  \end{align*}
  \]

- Because the 5th coordinate frame (tool) is arbitrary,
  \[
  \begin{align*}
  \alpha_4 &= 0 \\
  a_4 &= \sqrt{3}/2 \\
  \theta_4 &= f_{\theta_4}(\beta_3, \beta_4) \\
  d_4 &= f_{d_4}(\beta_3, \beta_4)
  \end{align*}
  \]

- $\theta_1 = d_1 = 0$ arbitrarily
Global Measurements

- Maximum reach

- The fraction of the total workspace that is fault tolerant, denoted $W_K$ ($K \geq \gamma K_{opt}$)
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.

$$W_K = \frac{n_f}{n_r}$$
Multiple Self-Motion Manifolds

- Some of the points have multiple self-motion manifolds
- On one manifold, $K \geq \gamma K_{opt}$
- 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 $K \geq \gamma K_{opt}$

Computationally expensive, i.e., 10-30 minutes/robot
Some of the points have multiple self-motion manifolds.

On one manifold, $K \geq \gamma K_{opt}$.

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 $K \geq \gamma K_{opt}$.

Computationally expensive, i.e., 10-30 minutes/robot.
Examples of Manipulators with Common Link Twist Parameters

- Setting $\alpha_i$’s to $\pm 90^\circ$, $0^\circ$, or $180^\circ$ is common in many commercial manipulators.

- Recall that the parameter $\alpha_i$ is defined as the angle between the rotation axes of joints $i$ and $i+1$, which is the same as $\omega_i$ and $\omega_{i+1}$. 
Link Twist $\alpha_i = 0^\circ$ or $180^\circ$

- $\omega_i \cdot \omega_{i+1} = 1$ when $\alpha_i = 0^\circ$
- $\omega_i \cdot \omega_{i+1} = -1$ when $\alpha_i = 180^\circ$
- This yields to discrete value of $\beta_i$ and $\beta_{i+1}$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\alpha_i$ [degrees]</th>
<th>$(\beta_i, \beta_{i+1})$ [degrees]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>(90, 90)</td>
</tr>
<tr>
<td></td>
<td>180</td>
<td>(90, 270)</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>(30, 120), (210, 300)</td>
</tr>
<tr>
<td></td>
<td>180</td>
<td>(30, 300), (210, 120)</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>(60, 60), (240, 240)</td>
</tr>
<tr>
<td></td>
<td>180</td>
<td>(60, 240), (240, 60)</td>
</tr>
</tbody>
</table>

$\alpha_i$ and $\alpha_{i+1}$ can’t be both $0^\circ$ (or $180^\circ$)
Link Twist $\alpha_i = \pm 90^\circ$

- $\omega_i \cdot \omega_{i+1} = 0$
- $0 \leq \beta_1 < 180^\circ$ (when $180 \leq \beta_1 < 360^\circ$ the resulting robots are mirrors), but $\beta_2 = f_{\beta_2}(\beta_1)$, $\beta_3 = f_{\beta_3}(\beta_2)$, and $\beta_4 = f_{\beta_4}(\beta_3)$
**Link Twist** $\alpha_i = \pm 90^\circ$

- $\omega_i \cdot \omega_{i+1} = 0$
- $0 \leq \beta_1 < 180^\circ$ (when $180 \leq \beta_1 < 360^\circ$ the resulting robots are mirrors), but $\beta_2 = f_{\beta_2}(\beta_1)$, $\beta_3 = f_{\beta_3}(\beta_2)$, and $\beta_4 = f_{\beta_4}(\beta_3)$
Link Twist $\alpha_i = \pm 90^\circ$

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

\[ \beta_1 = 10^\circ \]
Link Twist $\alpha_i = \pm 90^\circ$ 

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

<table>
<thead>
<tr>
<th>Robot Group</th>
<th>Relationship between joint axes $i - 1$ and $i$ $i = (1, 2, 3, 4)$</th>
<th>Size of Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(\parallel, \parallel, \parallel, \parallel)$</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>$(\parallel, \parallel, \perp, \parallel)$</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>$(\perp, \parallel, \parallel, \parallel)$</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>$(\parallel, \perp, \parallel, \parallel)$</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>$(\parallel, \perp, \perp, \parallel)$</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>$(\perp, \parallel, \perp, \parallel)$</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>$(\perp, \perp, \parallel, \parallel)$</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>$(\perp, \perp, \perp, \parallel)$</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

Best out of each group:

<table>
<thead>
<tr>
<th>$W_K$ [%]</th>
<th>max reach [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>59</td>
<td>5.19</td>
</tr>
<tr>
<td>30</td>
<td>4.96</td>
</tr>
<tr>
<td>67</td>
<td>3.92</td>
</tr>
<tr>
<td>48</td>
<td>3.93</td>
</tr>
<tr>
<td>75</td>
<td>5.50</td>
</tr>
</tbody>
</table>
## Manipulator Categories

<table>
<thead>
<tr>
<th>Robot Group</th>
<th>Relationship between joint axes $i-1$ and $i$ $i = (1, 2, 3, 4)$</th>
<th>Size of Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>($\parallel, \parallel, \parallel, \parallel$)</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>($\parallel, \parallel, \perp, \parallel$)</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>($\perp, \parallel, \parallel, \parallel$)</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>($\parallel, \perp, \parallel, \parallel$)</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>($\parallel, \perp, \perp, \parallel$)</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>($\perp, \parallel, \perp, \parallel$)</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>($\perp, \perp, \parallel, \parallel$)</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>($\perp, \perp, \perp, \parallel$)</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

### Best out of each group:

<table>
<thead>
<tr>
<th></th>
<th>$W_K$ [%]</th>
<th>max reach [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>59</td>
<td>5.19</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>4.96</td>
</tr>
<tr>
<td>3</td>
<td>67</td>
<td>3.92</td>
</tr>
<tr>
<td>4</td>
<td>48</td>
<td>3.93</td>
</tr>
<tr>
<td>5</td>
<td>75</td>
<td>5.50</td>
</tr>
</tbody>
</table>
## Manipulator Categories

<table>
<thead>
<tr>
<th>Robot Group</th>
<th>Relationship between joint axes $i - 1$ and $i$ $i = (1, 2, 3, 4)$</th>
<th>Size of Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(|, |, |, |)$</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>$(|, |, \perp, |)$</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>$(\perp, |, |, |)$</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>$(|, \perp, |, |)$</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>$(|, \perp, \perp, |)$</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>$(\perp, |, \perp, |)$</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>$(\perp, \perp, |, |)$</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>$(\perp, \perp, \perp, |)$</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

**Best out of each group:**

<table>
<thead>
<tr>
<th>$W_K^C$ [%]</th>
<th>$\text{max reach}$ [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>59</td>
<td>5.19</td>
</tr>
<tr>
<td>30</td>
<td>4.96</td>
</tr>
<tr>
<td>67</td>
<td>3.92</td>
</tr>
<tr>
<td>48</td>
<td>3.93</td>
</tr>
<tr>
<td>75</td>
<td>5.50</td>
</tr>
</tbody>
</table>
The Best Manipulator

DH parameters:

<table>
<thead>
<tr>
<th>i</th>
<th>$\alpha_i$ [degrees]</th>
<th>$a_i$ [m]</th>
<th>$d_i$ [m]</th>
<th>$\theta_i$ [degrees]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90</td>
<td>$\sqrt{2}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>-90</td>
<td>$\sqrt{2}$</td>
<td>1</td>
<td>180</td>
</tr>
<tr>
<td>3</td>
<td>90</td>
<td>$\sqrt{2}$</td>
<td>-1</td>
<td>180</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>$\sqrt{3}/2$</td>
<td>1/2</td>
<td>145</td>
</tr>
</tbody>
</table>
Columns Permutation and/or Sign Change Effect

- Sign change for every 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).


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!