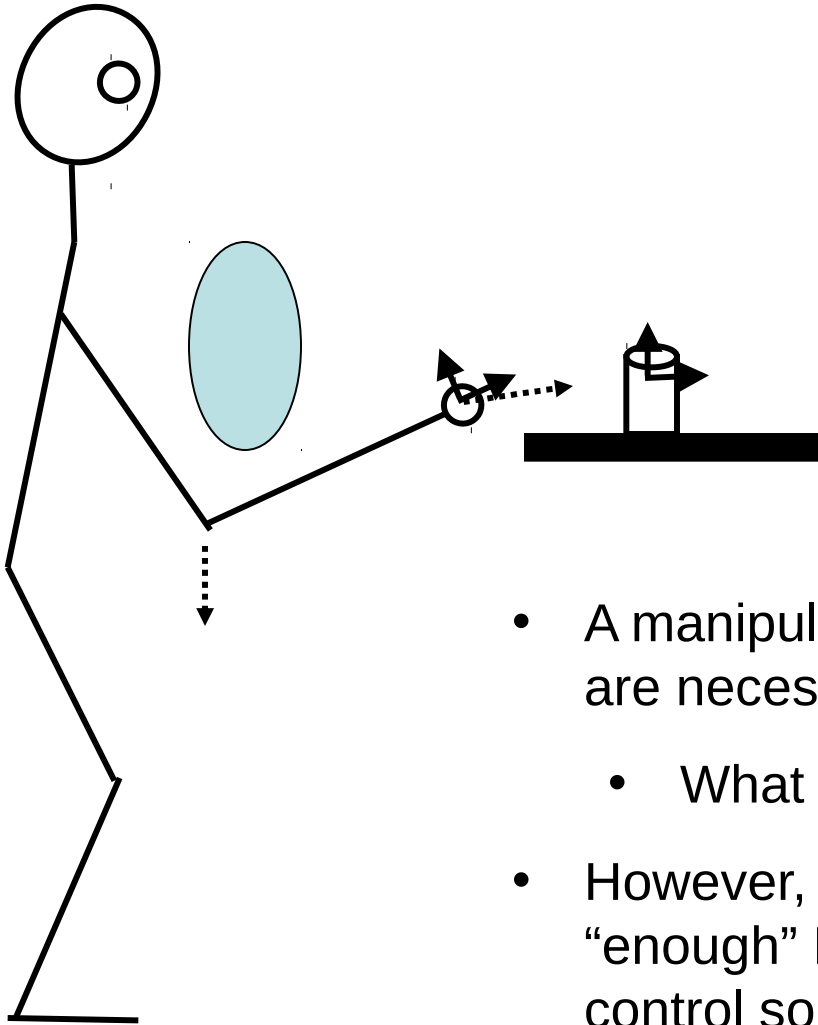


# Kinematic Redundancy



- A manipulator may have more DOFs than are necessary to control a desired variable
  - What do you do w/ the extra DOFs?
- However, even if the manipulator has “enough” DOFs, it may still be unable to control some variables in some configurations...

# Jacobian Range Space

Before we think about redundancy, let's look at the range space of the Jacobian transform:

The velocity Jacobian maps joint velocities onto end effector velocities:  $v = J_v(q)\dot{q}$

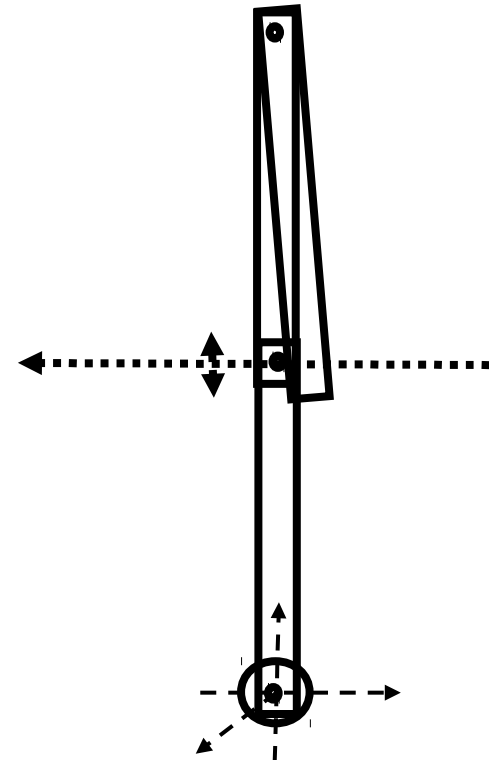
$$J_v(q) : Q \rightarrow V$$

Space of joint velocities

- This is the domain of  $J$ :  $D(J_v)$

Space of end effector velocities

- This is the range space of  $J$ :  $R(J_v)$



# Jacobian Range Space

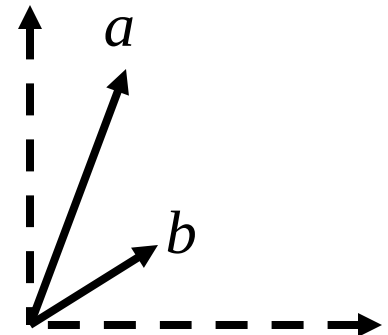
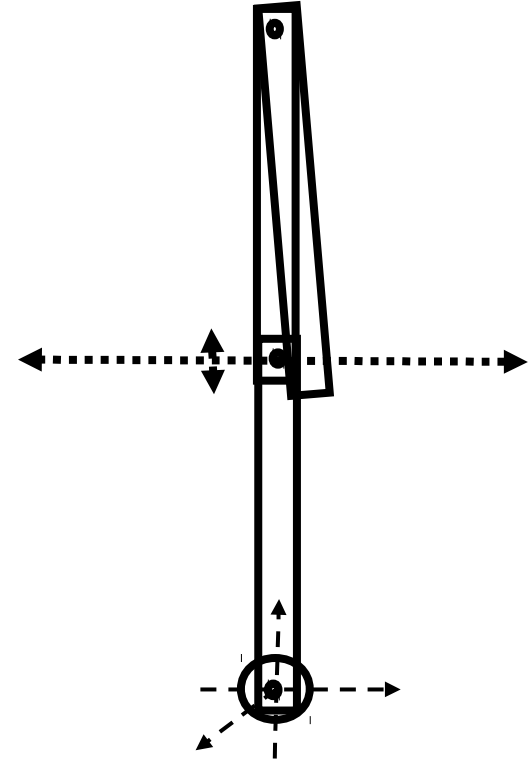
$$J_v(q) : Q \rightarrow V$$

In some configurations, the range space of the Jacobian may not span the entire space of the variable to be controlled:

$$\exists v \in V, v \notin R(J_v(q))$$

$$R(J_v(q)) \text{ spans } V \text{ if } \forall v \in V, v \in R(J_v(q))$$

Example:  $a$  and  $b$  span this two dimensional space:

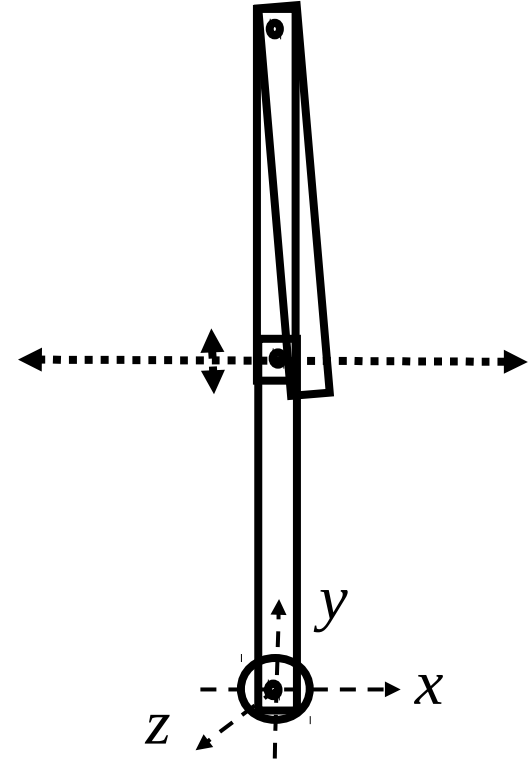


# Jacobian Range Space

This is the case in the manipulator to the right:

- In this configuration, the Jacobian does not span the  $y$  direction (or the  $z$  direction)

$$y \in V, y \notin R(J_v(q))$$



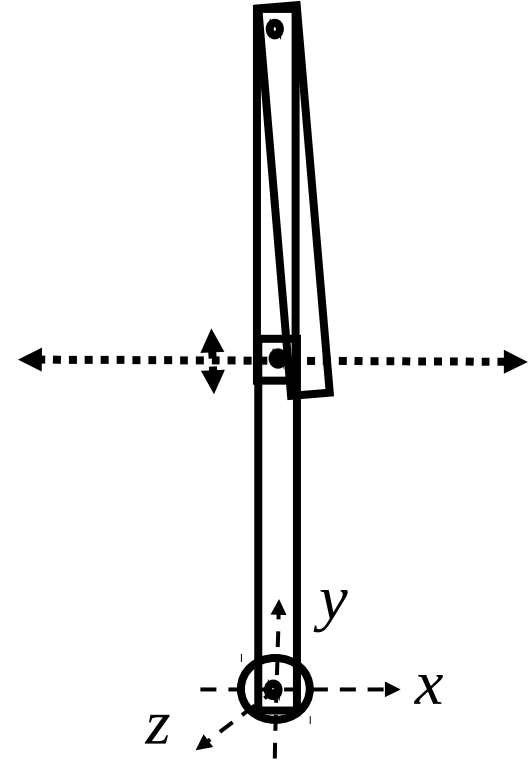
# Jacobian Range Space

Let's calculate the velocity Jacobian:

$$J_v(q) = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \\ 0 & 0 & 0 \end{bmatrix}$$

Joint configuration of manipulator:  $q = \begin{pmatrix} \frac{\pi}{2} \\ 0 \\ \pi \end{pmatrix}$

$$J_v(q) = \begin{bmatrix} -l_1 - l_2 + l_3 & -l_2 + l_3 & l_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



There is no joint velocity,  $\dot{q}$ , that will produce a  $y$  velocity,  $y = J_v(q)\dot{q}$

Therefore, you're in a singularity.

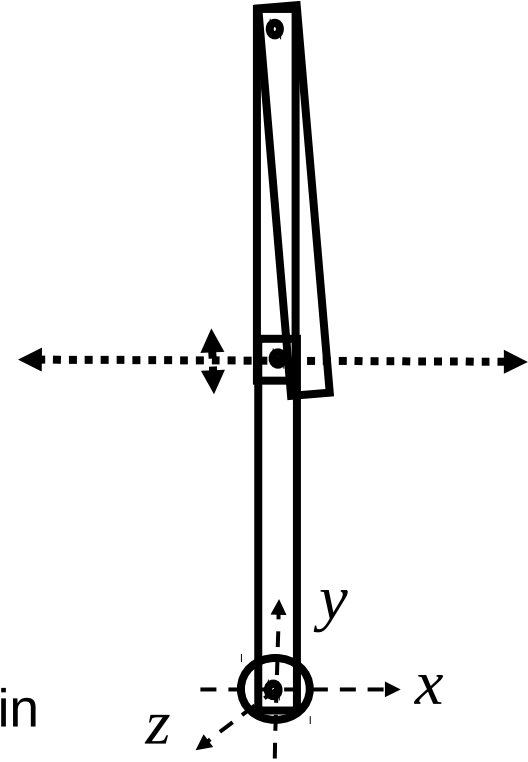
# Jacobian Singularities

In singular configurations:

- $J_v(q)$  does not span the space of Cartesian velocities
- $J_v(q)$  loses rank

Test for kinematic singularity:

- If  $\det[J(q)J(q)^T]$  is zero, then manipulator is in a singular configuration

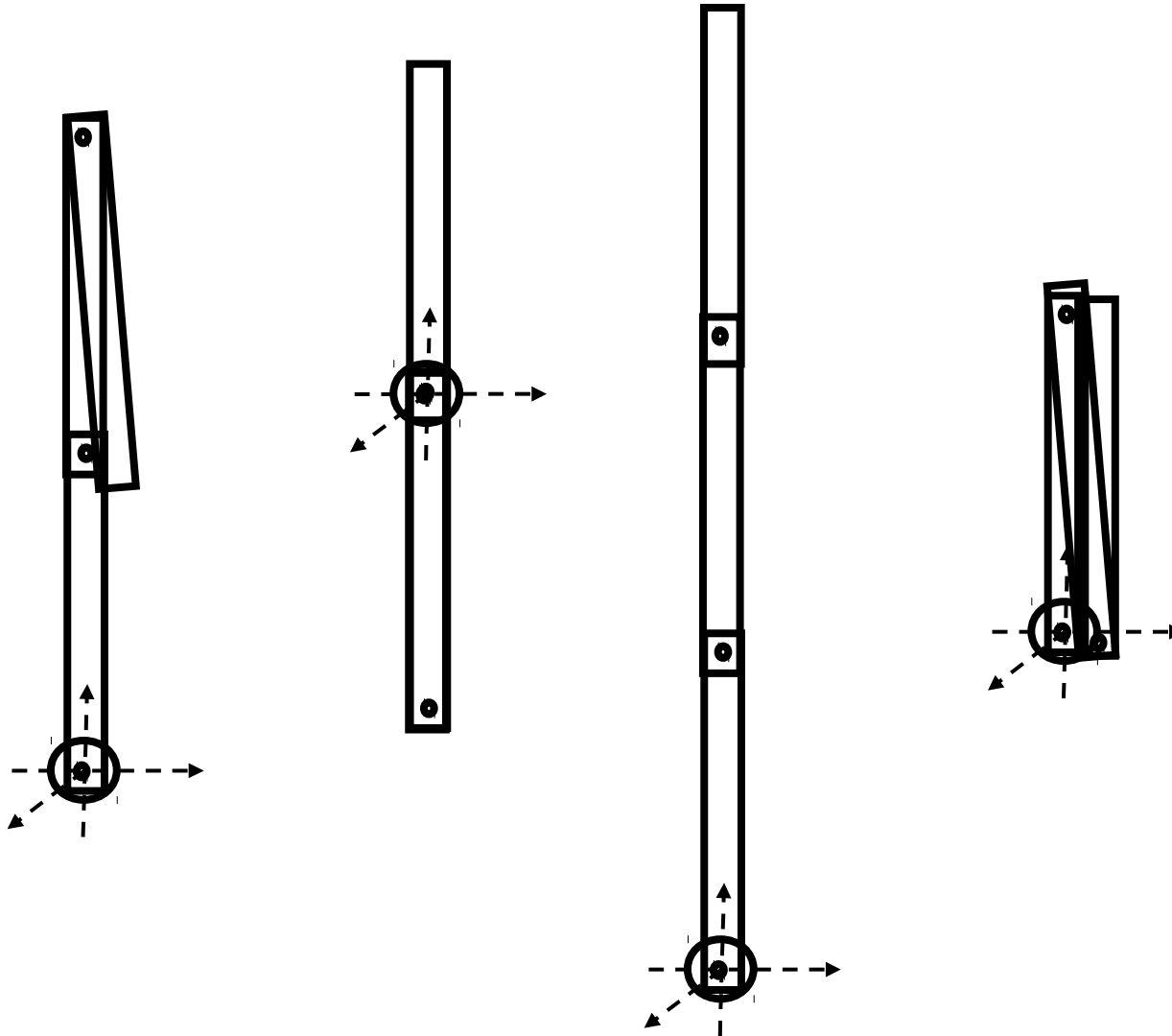


Example:

$$\det[J(q)J(q)^T] = \det \begin{bmatrix} -l_1 - l_2 + l_3 & -l_2 + l_3 & l_3 \\ 0 & 0 & 0 \end{bmatrix} = \det \begin{bmatrix} -l_1 - l_2 + l_3 & 0 \\ -l_2 + l_3 & 0 \\ l_3 & 0 \end{bmatrix} = \det \begin{bmatrix} \text{something} & 0 \\ 0 & 0 \end{bmatrix} = 0$$

# Jacobian Singularities: Example

The four singularities of the three-link planar arm:



# Jacobian Singularities and Cartesian Control

Cartesian control involves calculating the inverse or pseudoinverse:

$$J^\# = J^T (JJ^T)^{-1}$$

However, in singular configurations, the pseudoinverse (or inverse) does not exist because  $(JJ^T)^{-1}$  is undefined.

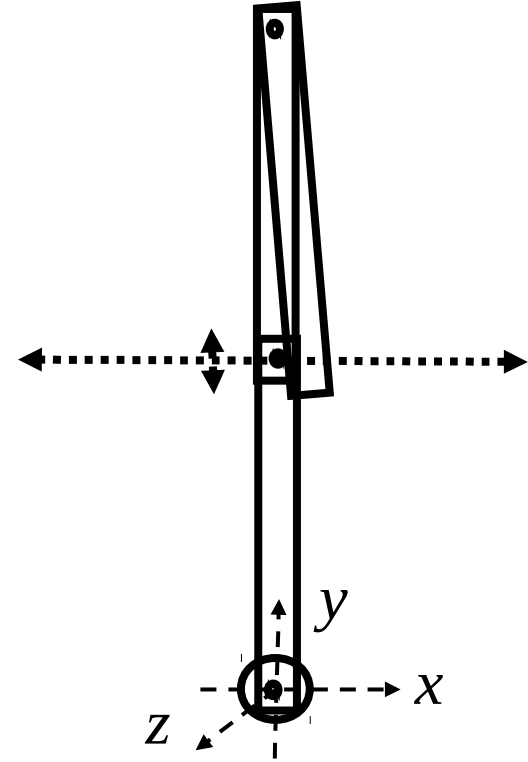
As you approach a singular configuration, joint velocities in the singular direction calculated by the pseudoinverse get very large:

$$\dot{q} = J^\# \dot{x}_s = J^T (JJ^T)^{-1} \dot{x}_s = \text{big}$$

In Jacobian transpose control, joint velocities in the singular direction (*i.e.* the gradient) go to zero:

$$\dot{q} = J^T \dot{x}_s = 0$$

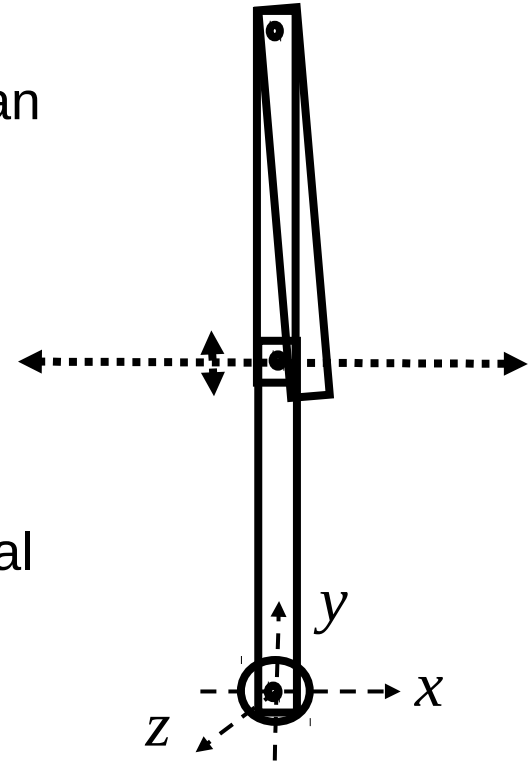
Where  $\dot{x}_s$  is a singular direction.





# Jacobian Singularities and Cartesian Control

- So, singularities are mostly a problem for Jacobian pseudoinverse control where the pseudoinverse “blows up”.
- Not much of a problem for transpose control
  - The worst that can happen is that the manipulator gets “stuck” in a singular configuration because the direction of the goal is in a singular direction.
  - This “stuck” configuration is unstable – any motion away from the singular configuration will allow the manipulator to continue on its way.

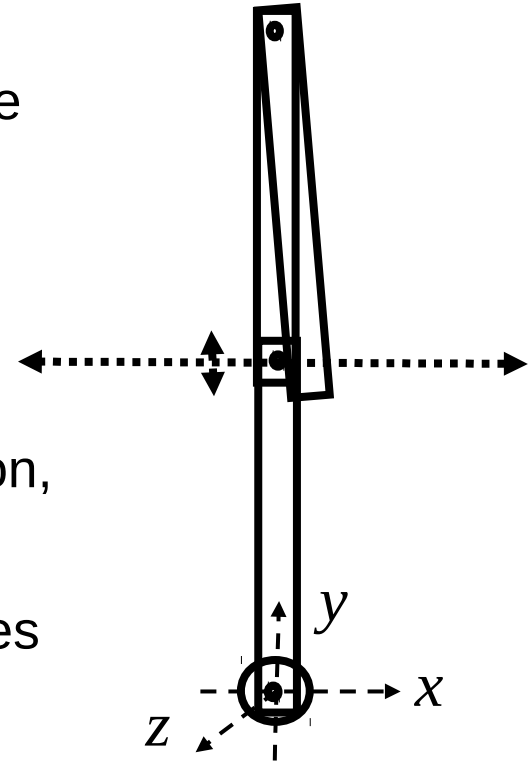


# Jacobian Singularities and Cartesian Control

One way to get the “best of both worlds” is to use the “damped least squares inverse” – aka the singularity robust (SR) inverse:

$$J^* = J^T (JJ^T + k^2 I)^{-1}$$

- Because of the additional term inside the inversion, the SR inverse does not blow up.
- In regions near a singularity, the SR inverse trades off exact trajectory following for minimal joint velocities.



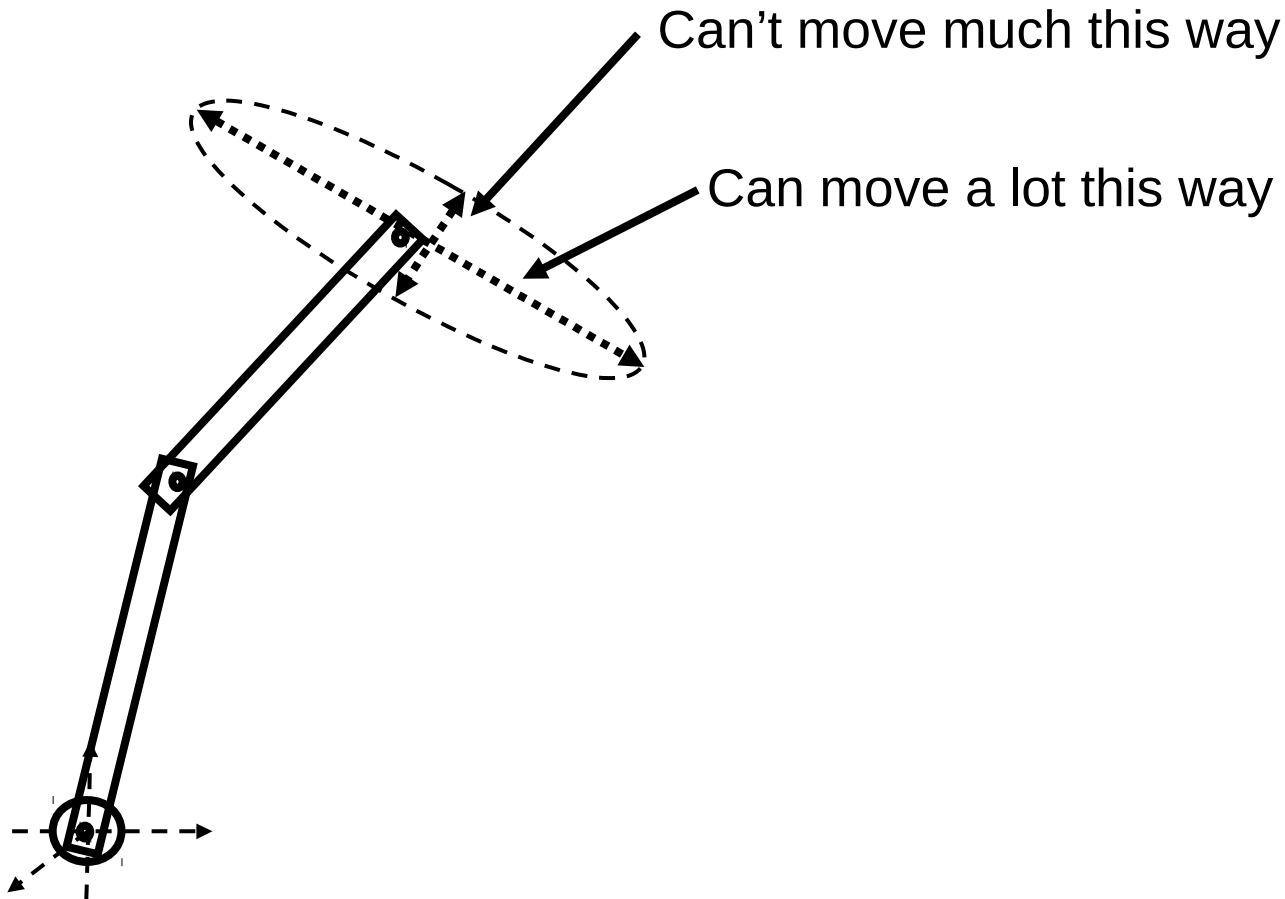
BTW, another way to handle singularities is simply to avoid them – this method is preferred by many

- More on this in a bit...

# Manipulability Ellipsoid

Can we characterize how close we are to a singularity?

Yes – imagine the possible instantaneous motions are described by an ellipsoid in Cartesian space.



# Manipulability Ellipsoid

The manipulability ellipsoid is an ellipse in Cartesian space corresponding to the twists that unit joint velocities can generate:

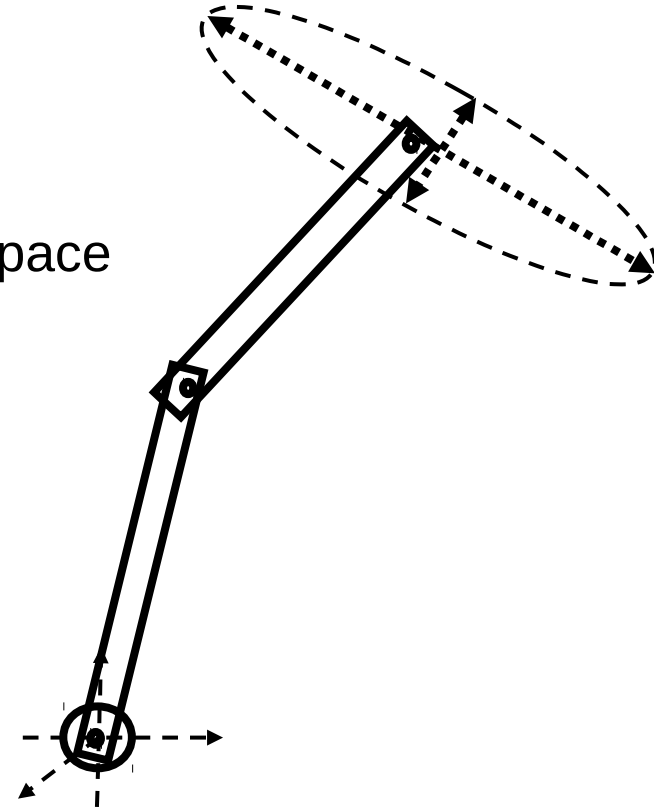
$\dot{q}^T \dot{q} = 1$  ← A unit sphere in joint velocity space

$(J^\# \dot{x})^T J^\# \dot{x} = 1$  ← Project the sphere into Cartesian space

$$\dot{x}^T \left( J^T (J J^T)^{-1} \right)^T J^T (J J^T)^{-1} \dot{x} = 1$$

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

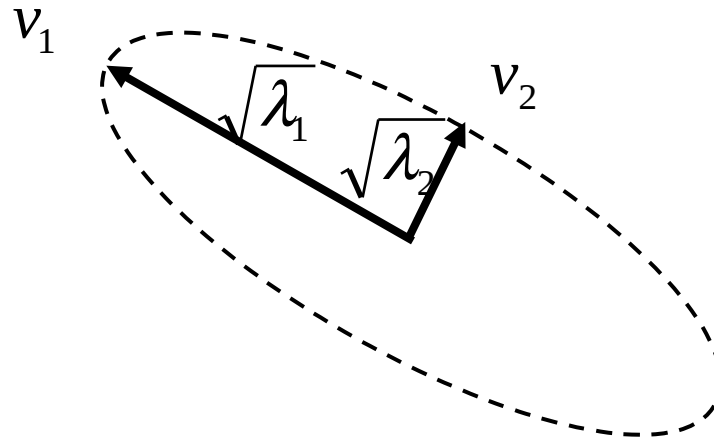
$\dot{x}^T (J J^T)^{-1} \dot{x} = 1$  ← The space of feasible Cartesian velocities



# Manipulability Ellipsoid

You can calculate the directions and magnitudes of the principle axes of the ellipsoid by taking the eigenvalues and eigenvectors of  $JJ^T$

- The lengths of the axes are the square roots of the eigenvalues



Yoshikawa's manipulability measure:  $\sqrt{\det(JJ^T)}$

- You try to maximize this measure
- Maximized in *isotropic* configurations
- This measures the volume of the ellipsoid

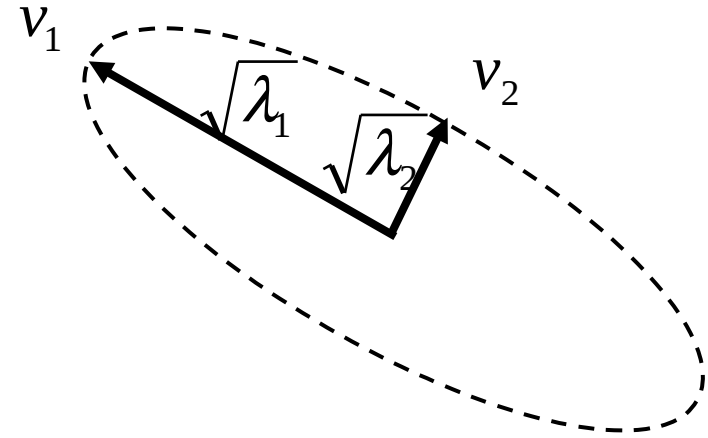
# Manipulability Ellipsoid

Another characterization of the manipulability ellipsoid: the ratio of the largest eigenvalue to the smallest eigenvalue:

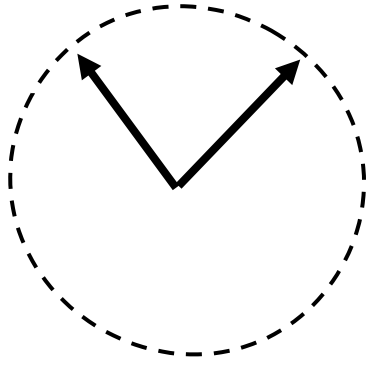
- Let  $\lambda_1$  be the largest eigenvalue and let  $\lambda_n$  be the smallest.
- Then the *condition number* of the ellipsoid is:

$$k = \frac{\sqrt{\lambda_1}}{\sqrt{\lambda_n}}$$

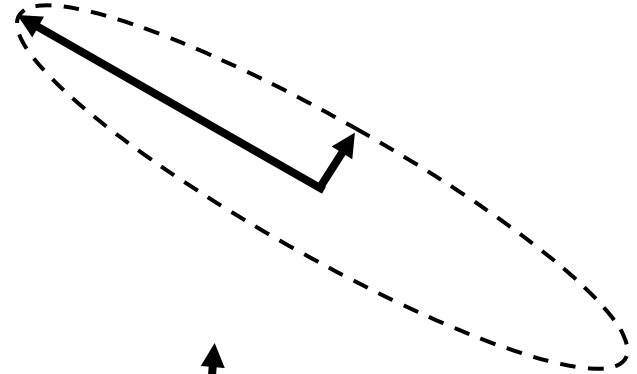
- The closer to one the condition number, the more *isotropic* the ellipsoid is.



# Manipulability Ellipsoid



Isotropic manipulability  
ellipsoid



NOT isotropic  
manipulability ellipsoid

# Force Manipulability Ellipsoid

You can also calculate a manipulability ellipsoid for force:

$\tau^T \tau = 1$    ← A unit sphere in the space of joint torques

$$\tau = J^T F$$

$$(J^T F)^T J^T F = 1$$

$F^T J J^T F = 1$    ← The space of feasible Cartesian wrenches

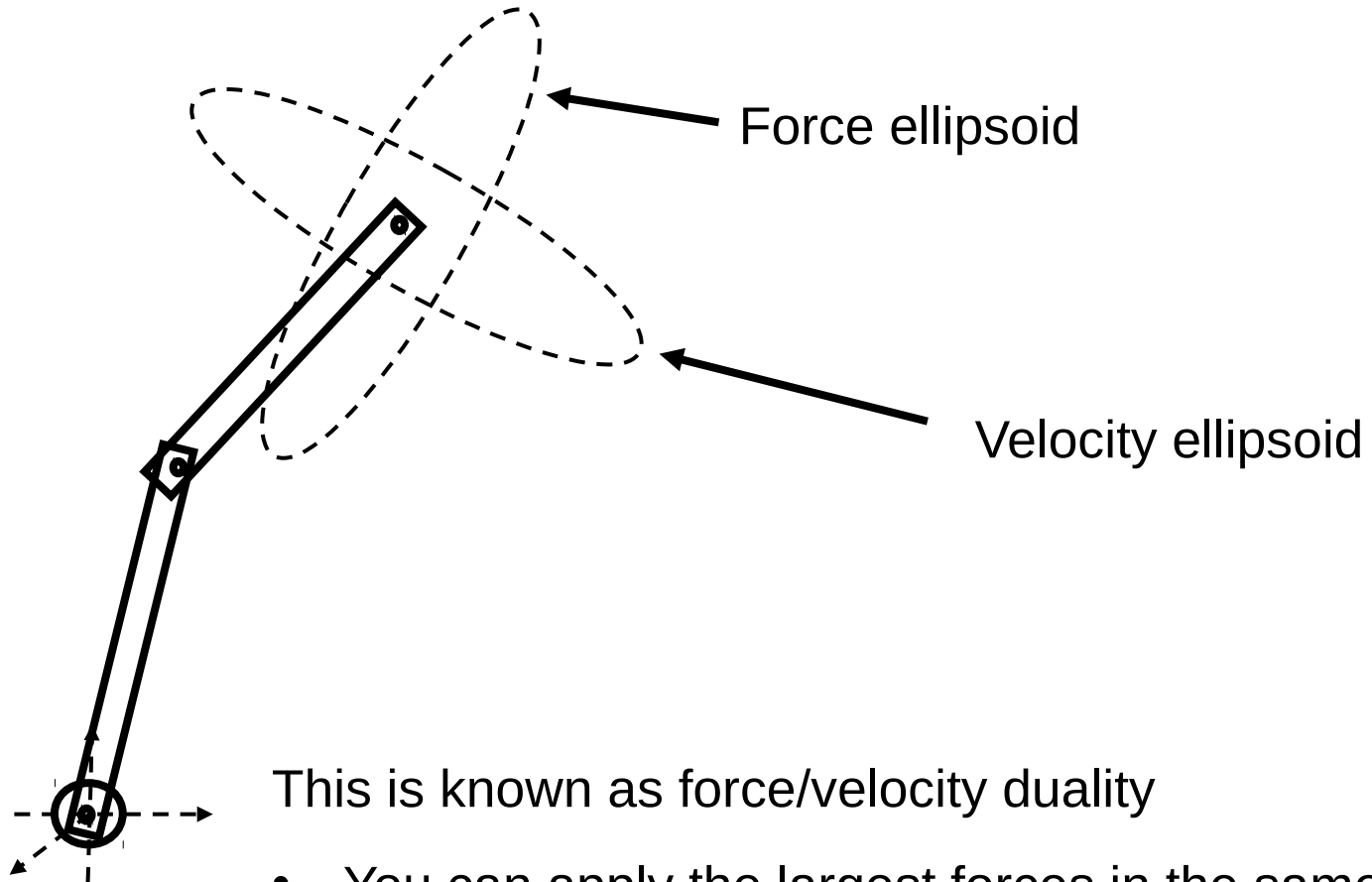


# Manipulability Ellipsoid

Principle axes of the force manipulability ellipsoid: the eigenvalues and eigenvectors of:  $(JJ^T)^{-1}$

- $(JJ^T)^{-1}$  has the same eigenvectors as  $JJ^T$ :  $v_i^v = v_i^f$
- But, the eigenvalues of the force and velocity ellipsoids are reciprocals:  
$$\lambda_i^f = \frac{1}{\lambda_i^v}$$
- Therefore, the shortest principle axes of the velocity ellipsoid are the longest principle axes of the force ellipsoid and vice versa...

# Velocity and force manipulability are orthogonal!



This is known as force/velocity duality

- You can apply the largest forces in the same directions that your max velocity is smallest
- Your max velocity is greatest in the directions where you can only apply the smallest forces

# Manipulability Ellipsoid: Example

Solve for the principle axes of the manipulability ellipsoid for the planar two link manipulator with unit length links at

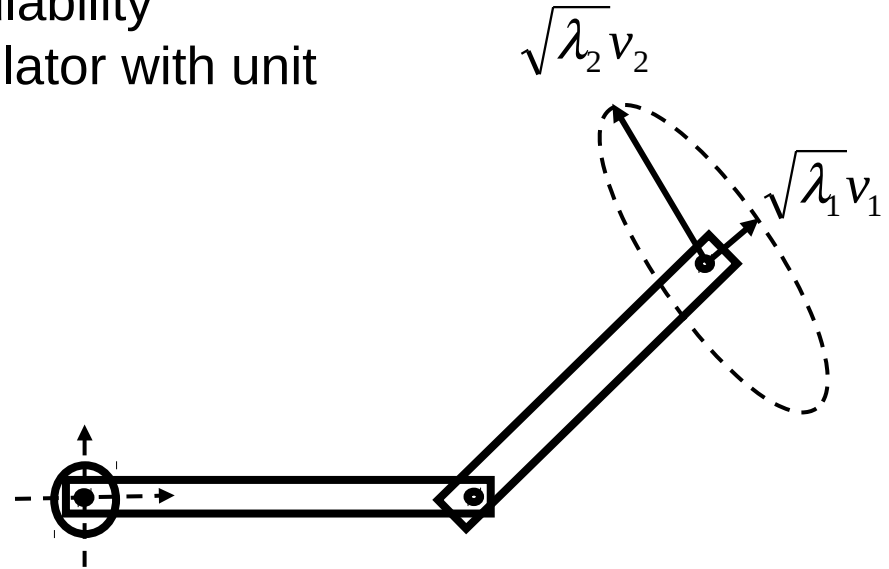
$$q = \begin{pmatrix} 0 \\ \frac{\pi}{4} \end{pmatrix}$$

$$J(q) = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}$$

$$J(q) = \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 1 + \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$J(q)J(q)^T = \begin{bmatrix} 1 - \lambda & -1 + \frac{1}{\sqrt{2}} \\ -1 + \frac{1}{\sqrt{2}} & 2 + \sqrt{2} - \lambda \end{bmatrix}$$

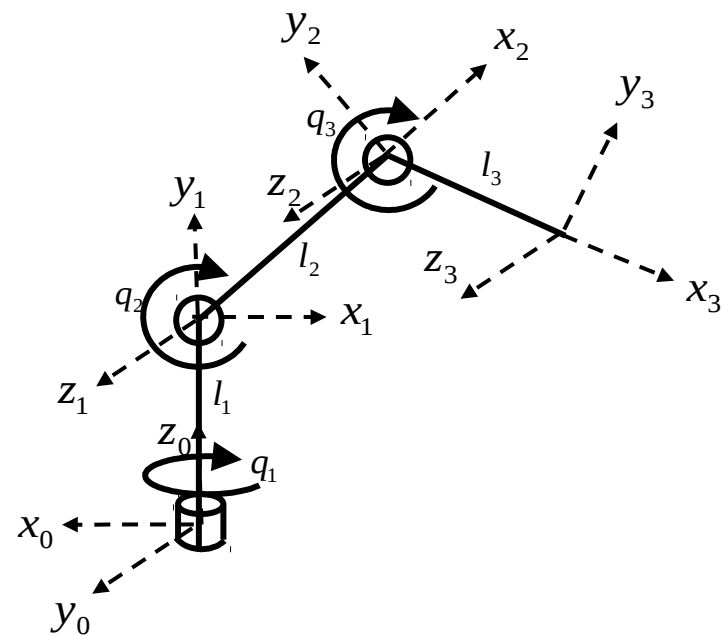
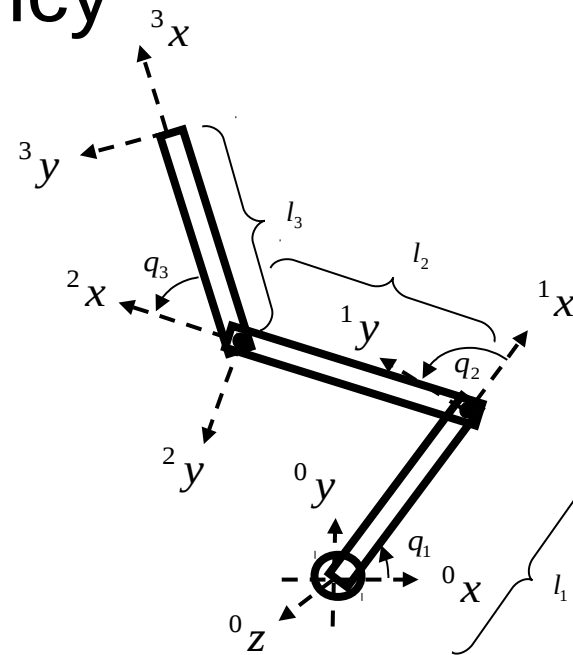
Principle axes:  $\sqrt{\lambda_1} v_1 = \begin{pmatrix} -0.3029 \\ -0.1568 \end{pmatrix}$   
 $\sqrt{\lambda_2} v_2 = \begin{pmatrix} -0.9530 \\ 1.8411 \end{pmatrix}$



# Kinematic redundancy

A general-purpose robot arm frequently has more DOFs than are strictly necessary to perform a given function

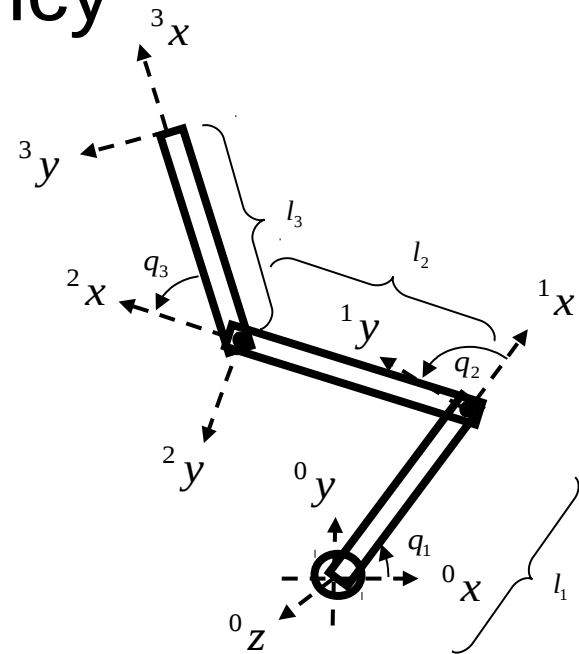
- in order to independently control the position of a planar manipulator end effector, only two DOFs are strictly necessary
  - If the manipulator has three DOFs, then it is *redundant w.r.t. the task* of controlling two dimensional position.
- In order to independently control end effector position in 3-space, you need at least 3 DOFs
- In order to independently control end effector position and orientation, at least 6 DOFs are needed (they have to be configured right, too...)



# Kinematic redundancy

The local redundancy of an arm can be understood in terms of the local Jacobian

- The manipulator controls a number of Cartesian DOFs equal to the number of independent rows in the Jacobian



$$J = \begin{bmatrix} \dot{J}_{11} & \dot{J}_{12} & \dot{J}_{13} \\ \dot{J}_{21} & \dot{J}_{22} & \dot{J}_{23} \end{bmatrix}$$

Since there are two *independent* rows, you can control two Cartesian DOFs independently ( $m=2$ )

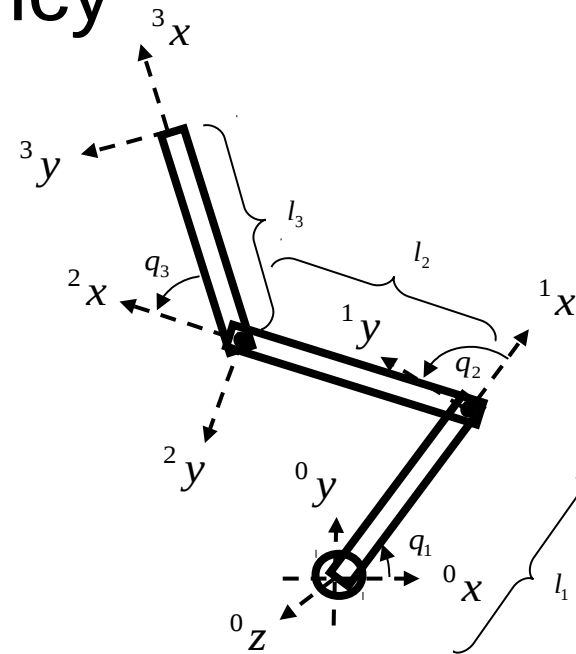
You use three joints to control two Cartesian DOFs ( $n=3$ )

Since the number of independent Cartesian directions is less than the number of joints, ( $m < n$ ), this manipulator is redundant w.r.t. the task of controlling those Cartesian directions.

# Kinematic redundancy

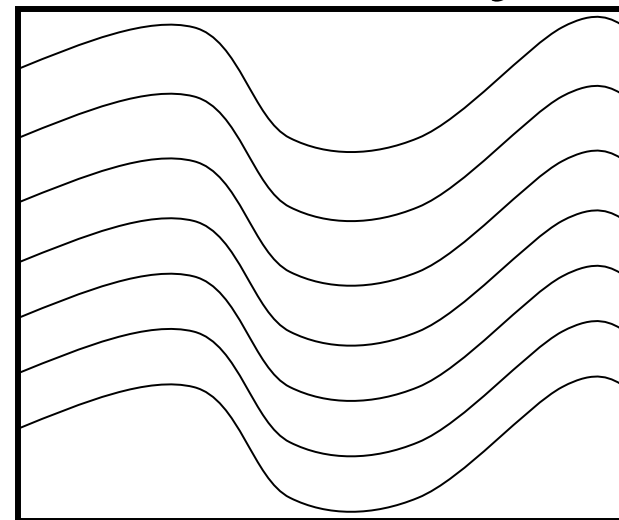
What does this redundant space look like?

- At first glance, you might think that it's linear because the Jacobian is linear
- But, the Jacobian is only *locally linear*



The dimension of the redundant space is the number of joints – the number of independent Cartesian DOFs:  $n-m$ .

- For the three link planar arm, the redundant space is a set of one dimensional curves traced through the three dimensional joint space.
- Each curve corresponds to the set of joint configurations that place the end effector in the same position.



Redundant manifolds in joint space

# Kinematic redundancy

Joint velocities in redundant directions causes no motion at the end effector

- These are *internal motions* of the manipulator.

Redundant joint velocities satisfy this

$$\text{equation: } 0 = J(q)\dot{q}$$



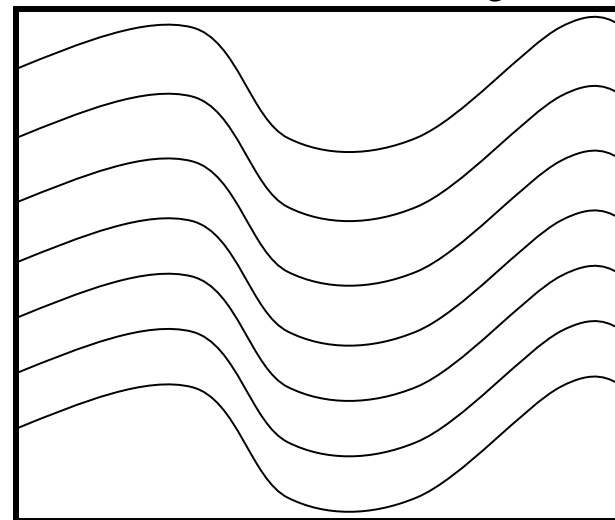
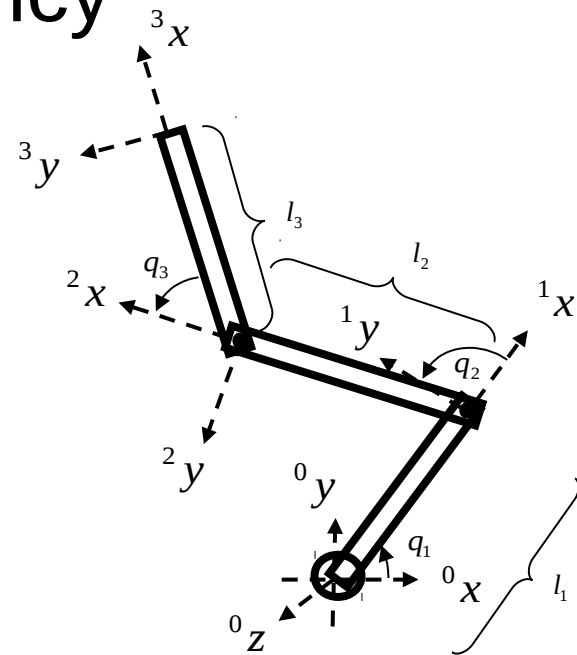
the *null space* of  $J(q)$



$$N(J(q)) = \{\dot{q} \in \dot{Q} : 0 = J(q)\dot{q}\}$$

Compare to the *range space* of  $J(q)$  :

$$R(J(q)) = \{\dot{x} \in \dot{X} : \exists \dot{q} \in \dot{Q}, \dot{x} = J(q)\dot{q}\}$$



Redundant manifolds in joint space

# Null space and Range space

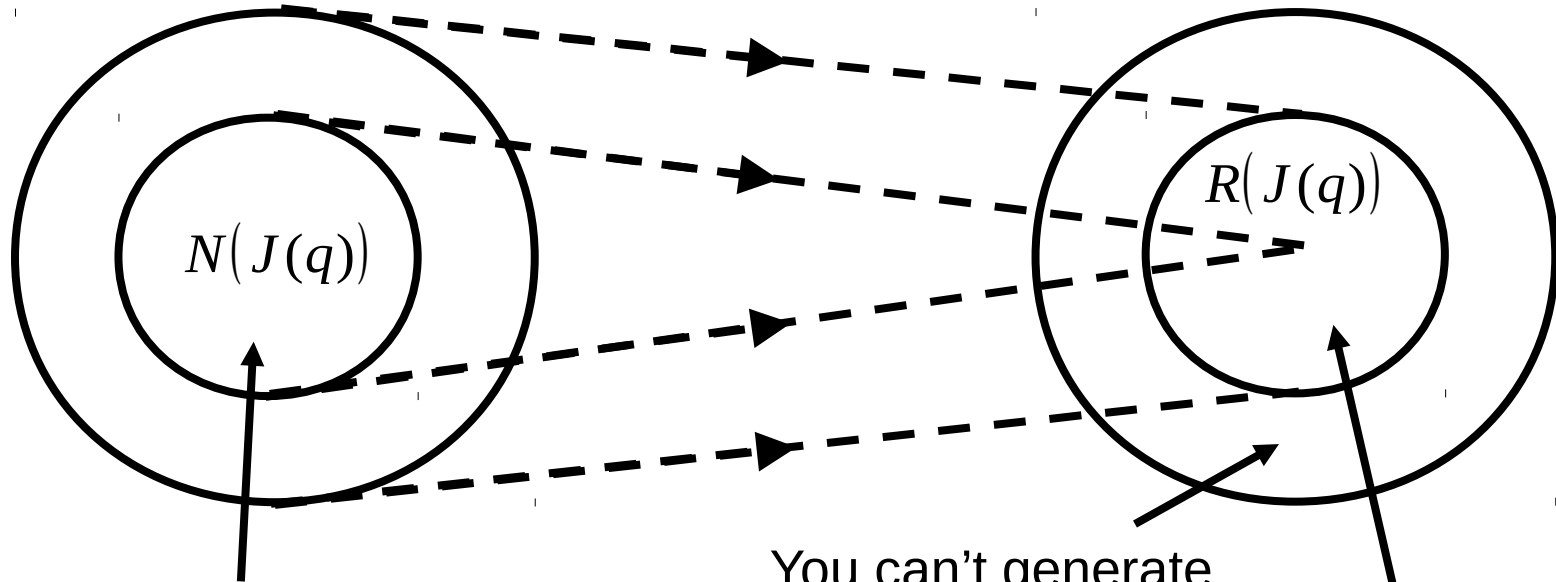
Joint space

$$Q \subseteq SO(n-1)$$

Cartesian space

$$X \subseteq R^m$$

$$\dot{x} = J(q)\dot{q}$$



$$N(J(q))$$

$$R(J(q))$$

$$N(J(q)) = \{\dot{q} \in \dot{Q} : 0 = J(q)\dot{q}\}$$

Null space

- Motions in the null space are internal motions

You can't generate these motions

$$R(J(q)) = \{\dot{x} \in \dot{X} : \exists \dot{q} \in \dot{Q}, \dot{x} = J(q)\dot{q}\}$$

Range space

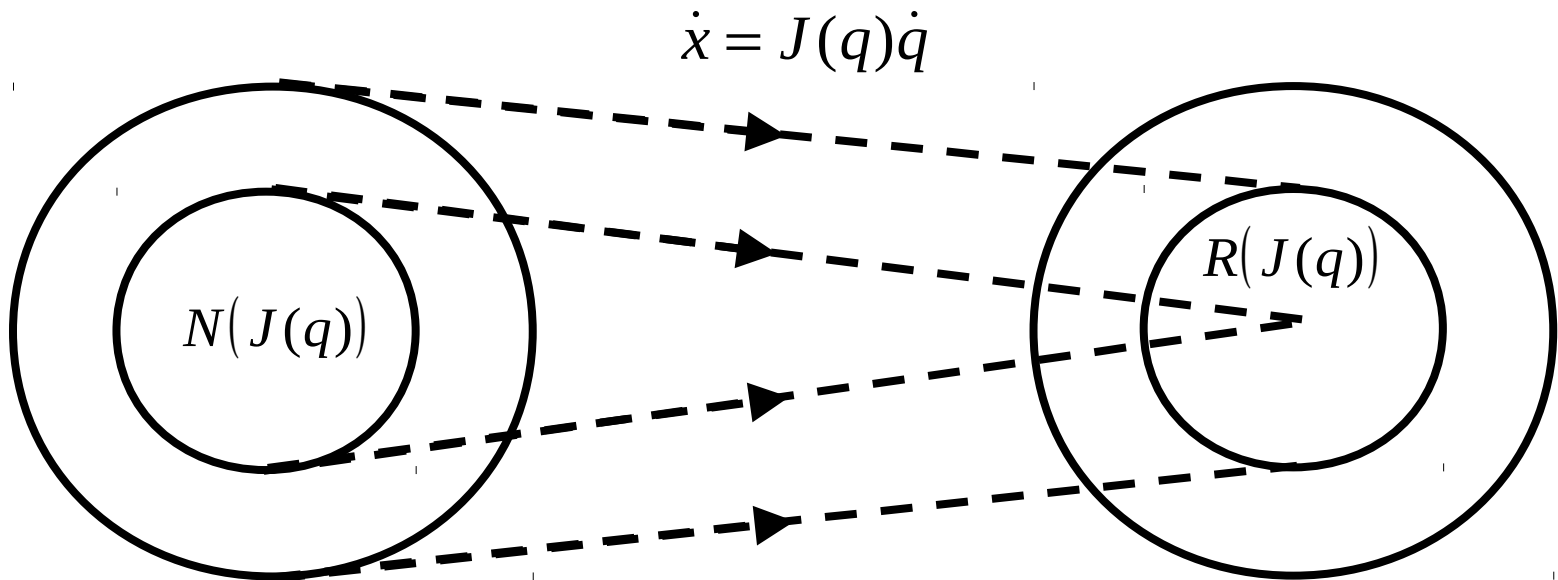


# Null space and Range space

Degree of manipulability:  $\dim(R(J(q)))$

Degree of redundancy:  $\dim(N(J(q)))$

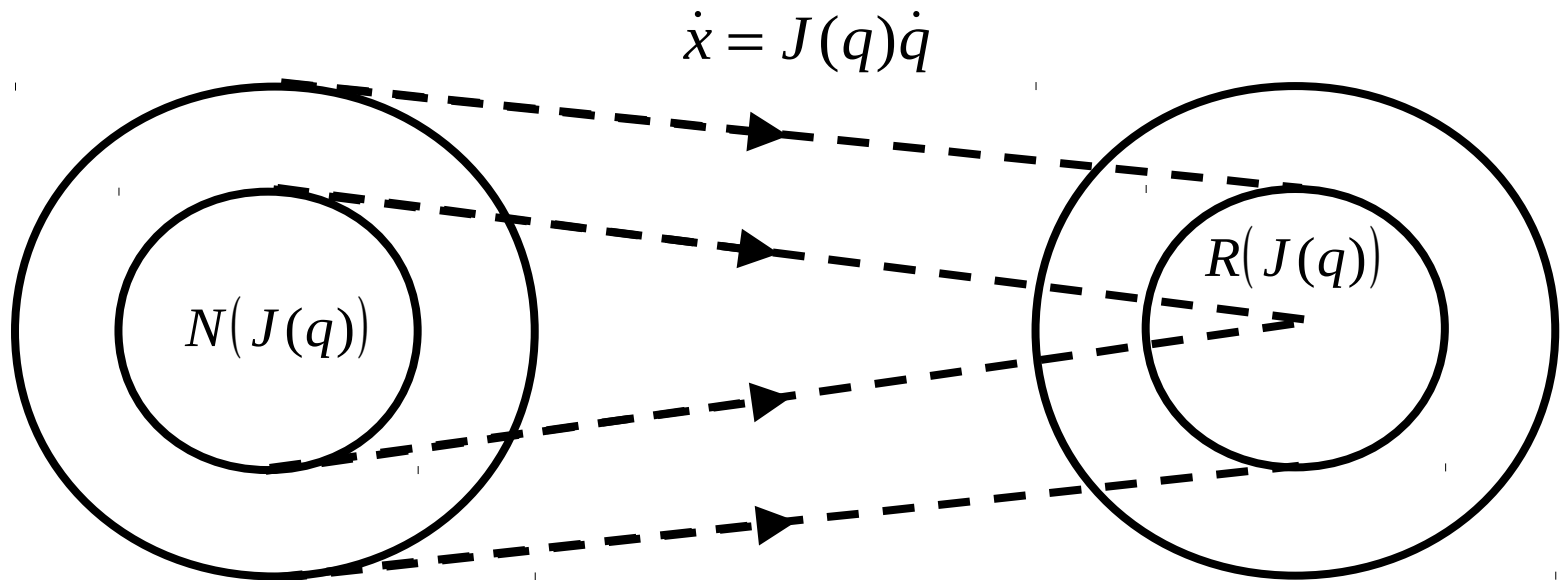
$$\dim(N(J(q))) + \dim(R(J(q))) = \text{total DOF of manipulator}$$



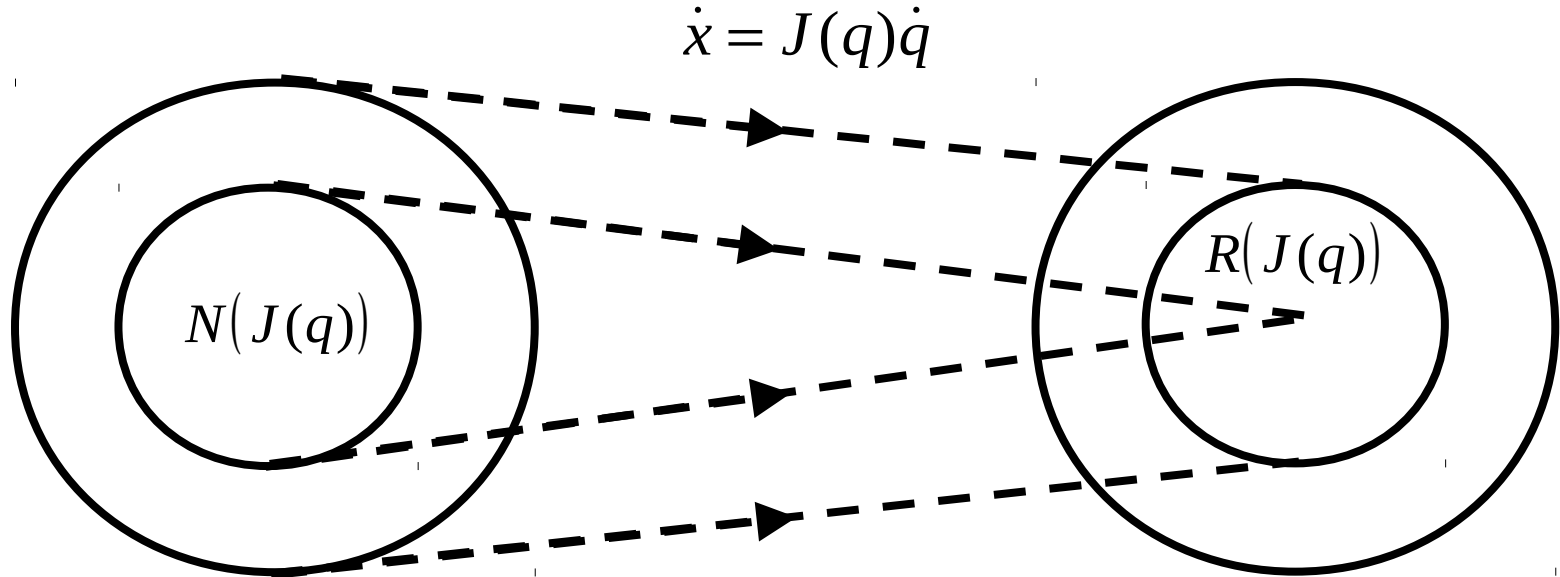
# Null space and Range space

As the manipulator moves to new configurations, the degree of manipulability may temporarily decrease – these are the singular configurations.

- There is a corresponding increase in degree of redundancy.



# Null space and Range space

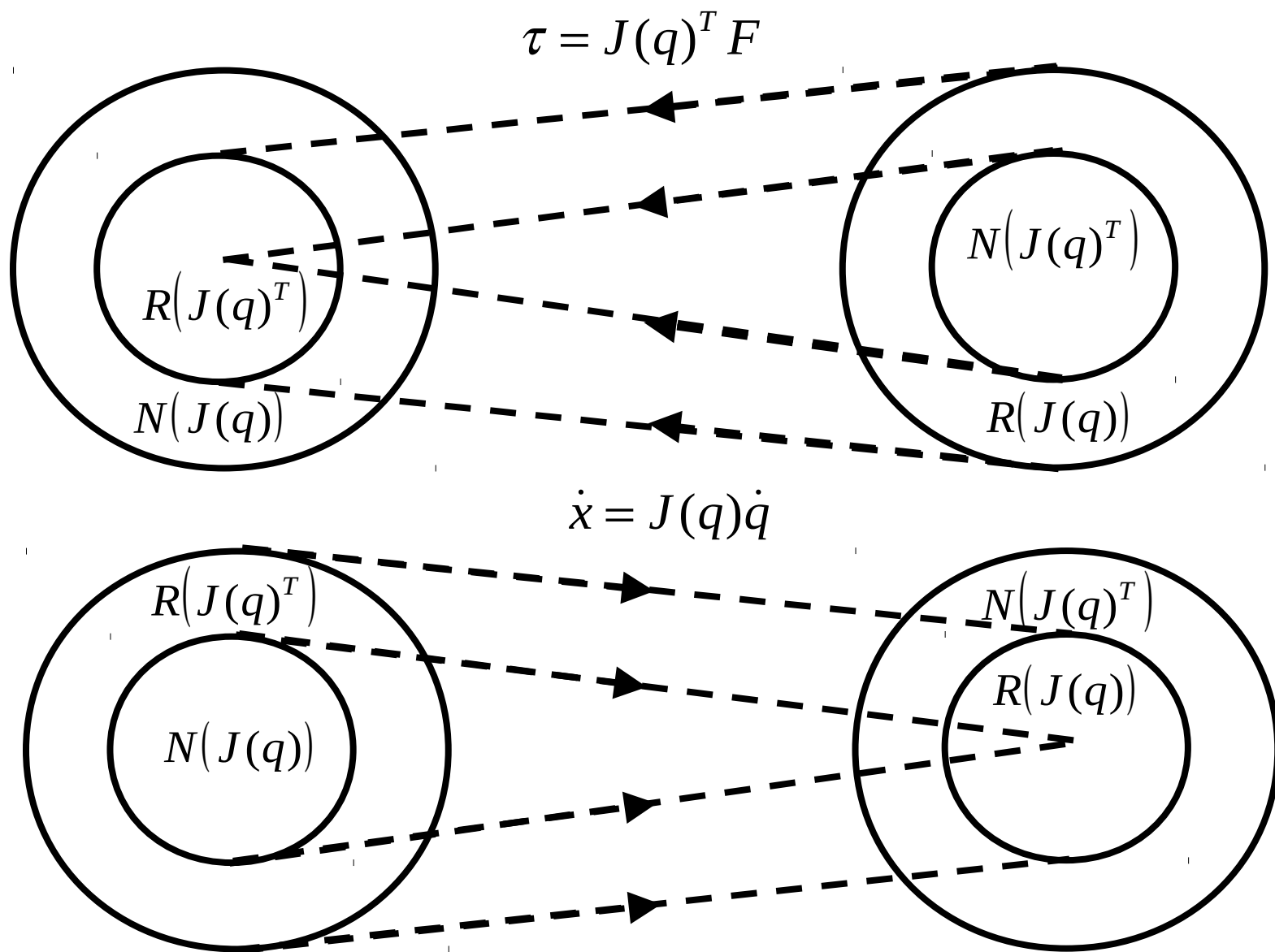


Remember the Jacobian's application to statics:  $\tau = J(q)^T F$

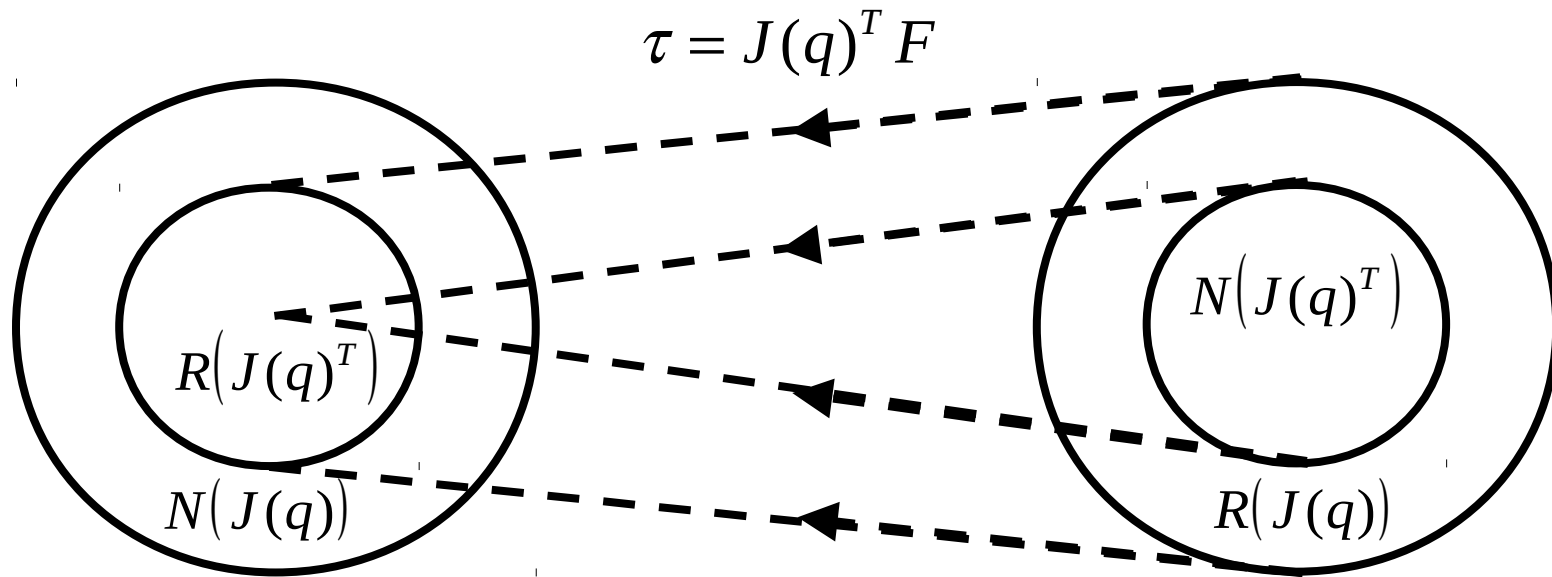
$$R(J(q)) = N^\perp(J(q)^T)$$

$$N(J(q)) = R^\perp(J(q)^T)$$

# Null space and Range space in the Force Domain



# Null space and Range space in the Force Domain



$$R(J(q)) = N^\perp(J(q)^T)$$

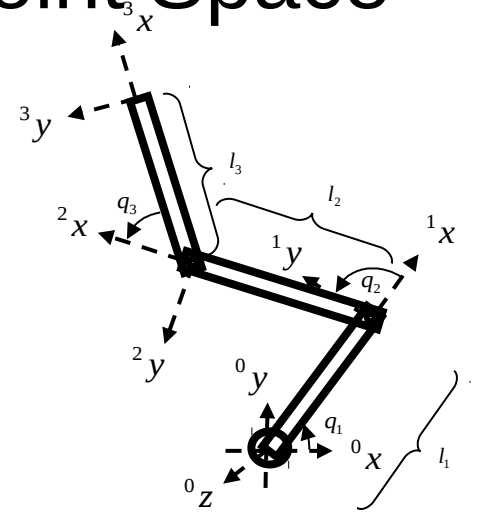
$$N(J(q)) = R^\perp(J(q)^T)$$

- A Cartesian force cannot generate joint torques in the joint velocity null space.
- ...

# Doing Things in the Redundant Joint Space

Motions in the redundant space do not affect the position of the end effector.

- Since they don't change end effector position, is there something we would like to do in this space?
  - Optimize kinematic manipulability?
  - Stay away from obstacles?
  - Something else?



# Doing Things in the Redundant Joint Space

Assume that you are given a joint velocity,  $\dot{q}_0$ , you would like to achieve while also achieving a desired end effector twist,  $\dot{x}_d$

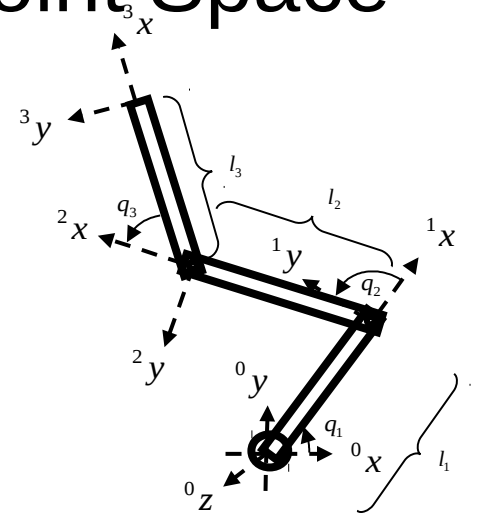
- Required objective:  $\dot{x}_d$
- Desired objective:  $\dot{q}_0$

$$f(\dot{q}) = (\dot{q} - \dot{q}_0)^T (\dot{q} - \dot{q}_0)$$

$$g(\dot{q}) = J\dot{q} - \dot{x}$$

Minimize  $f(z)$  subject to  $g(z) = 0$  :

Use lagrange multiplier method:  $\nabla_z f(z) = \lambda \nabla_z g(z)$



# Doing Things in the Redundant Joint Space

$$\nabla f = (\dot{q} - \dot{q}_0)^T$$

$$\nabla g = J$$

$$\nabla_z f(z) = \lambda \nabla_z g(z)$$

$$(\dot{q} - \dot{q}_0)^T = \lambda^T J$$

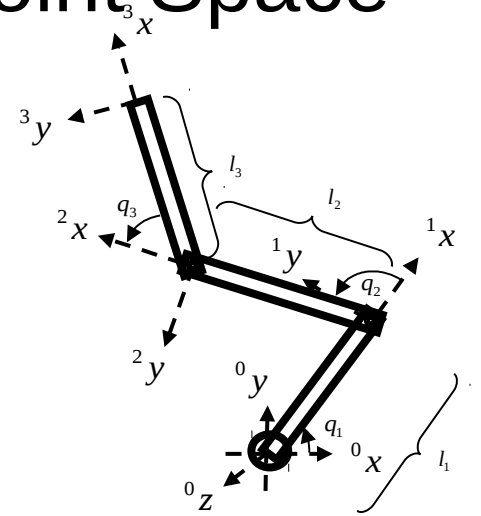
$$\dot{q} = J^T \lambda - \dot{q}_0$$

$$J(J^T \lambda - \dot{q}_0) = \dot{x}$$

$$\lambda = (JJ^T)^{-1}(\dot{x} - J\dot{q}_0)$$

$$\dot{q} = J^T (JJ^T)^{-1}(\dot{x} - J\dot{q}_0) + \dot{q}_0$$

$$\dot{q} = J^\# \dot{x} + (I - J^\# J) \dot{q}_0$$

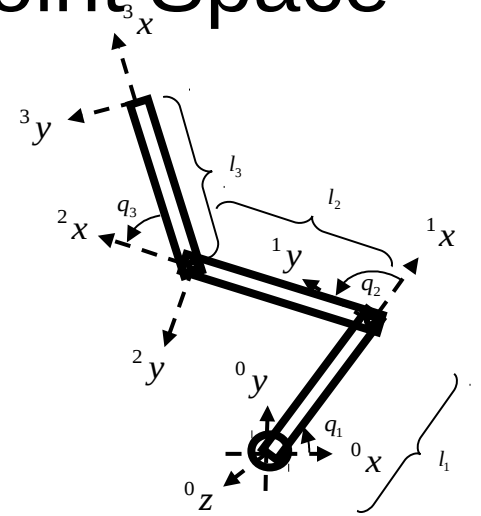




# Doing Things in the Redundant Joint Space

$$\dot{q} = J^\# \dot{x} + \underbrace{(I - J^\# J)}_{\uparrow} \dot{q}_0$$

Homogeneous part of the solution



Null space projection matrix:  $I - J^\# J$

- This matrix projects an *arbitrary* vector into the null space of  $J$
- This makes it easy to do things in the redundant space – just calculate what you would like to do and project it into the null space.

# Things You Might do in the Null Space

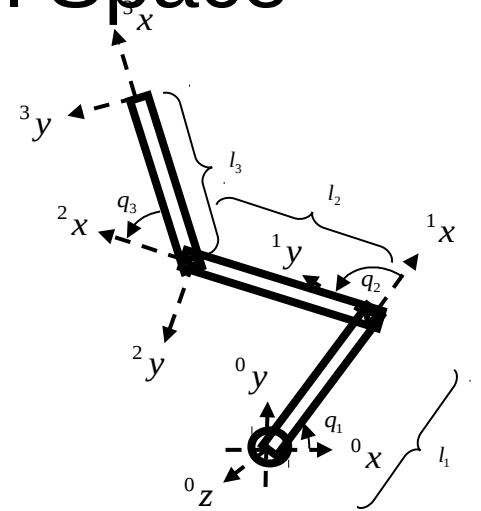
Avoid kinematic singularities:

1. Calculate the gradient of the manipulability measure:

$$\dot{q}_0 = \nabla \sqrt{\det(JJ^T)}$$

2. Project into null space:

$$\dot{q} = J^\# \dot{x} + (I - J^\# J) \dot{q}_0$$



Avoid joint limits:

1. Calculate a gradient of the squared distance from a joint limit:

$$\dot{q}_0 = \alpha(q_m - q)$$

2. Project into null space:

$$\dot{q} = J^\# \dot{x} + (I - J^\# J) \dot{q}_0$$

- where  $q_m$  is the joint configuration at the center of the joints
- and  $q$  is the current joint position

# Things You Might do in the Null Space

Avoid kinematic obstacles:

1. Consider a set of control points (nodes) on the manipulator:
2. Move all nodes away from the object:
3. Project desired motion into joint space:
4. Project into null space:

$$\{x_1, x_2, x_3\}$$

$$\nabla x_i = x_i - x_{obstacle}$$

$$\dot{q}_0 = \sum_{i \in nodes} J_i^T \nabla x_i$$

$$\dot{q} = J^\# \dot{x} + (I - J^\# J) \dot{q}_0$$

