MEAM 520
Lecture 15: Velocity Kinematics

Cynthia Sung, Ph.D.

Mechanical Engineering & Applied Mechanics

University of Pennsylvania

Last Class: What is the time derivative of a rotation matrix?

$$R = R(\theta) \in SO(3)$$

$$\dot{R} = \frac{dR}{dt} = ? = \frac{dR}{d\theta}\frac{d\theta}{dt}$$

? ✓

$$\frac{dR}{d\theta} = ?$$

$$R\,R^T = I$$

$$\frac{d}{d\theta}\left(R\,R^T\right) = \frac{d}{d\theta}(I)$$

$$\frac{dR}{d\theta}R^T + R\frac{dR^T}{d\theta} = 0$$

This equation has a special form.

$$\text{define } S = \frac{dR}{d\theta}R^T \quad S^T = R\frac{dR^T}{d\theta}$$

$$S + S^T = 0$$

Sidebar on Skew-Symmetric Matrices

$$S + S^T = 0$$

$$S = \begin{bmatrix} 0 & -s_3 & s_2 \\ s_3 & 0 & -s_1 \\ -s_2 & s_1 & 0 \end{bmatrix}$$

$$S(\vec{a}) = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

$$S(\vec{a})\,\vec{p} = ? = \vec{a} \times \vec{p}$$

Skew-symmetric matrices are a matrix-based way to represent a cross-product between vectors.

Skew-Symmetric Matrices

a skew-symmetric matrix formed from omega

The time derivative of a rotation matrix is...

$$\dot{R}(t) = S(\vec{\omega}(t))R(t)$$

times the rotation matrix itself

angular velocity of rotating frame w.r.t. the fixed frame at time t

$$\frac{dR}{d\theta} = S(\hat{\omega})\,R$$

The skew-symmetric matrix S defines the axis about which rotation is occurring.

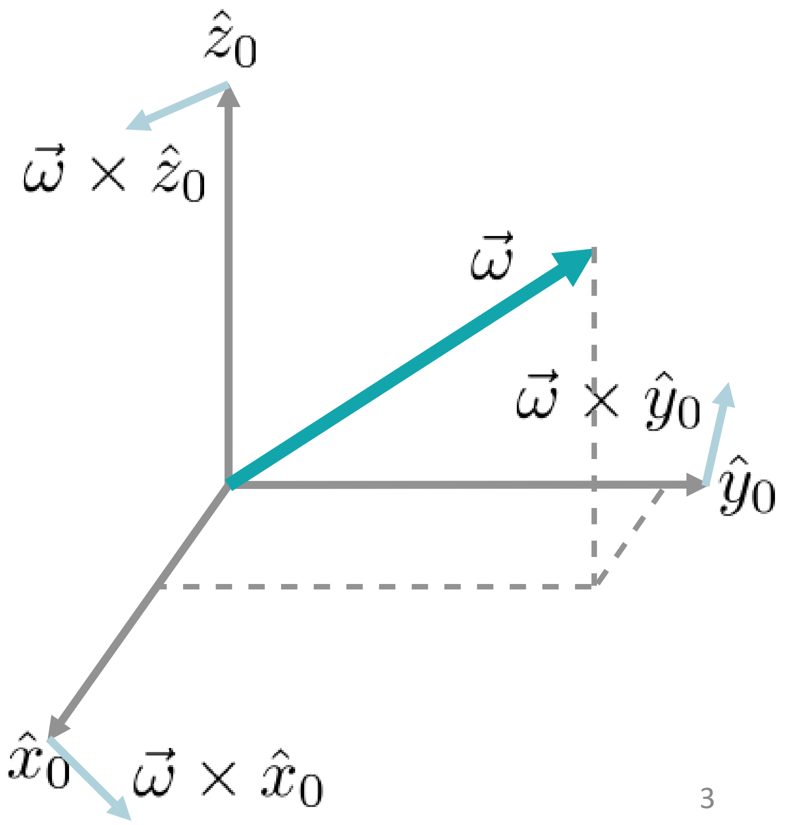
unit vector showing rotational axis

$$\frac{dR}{dt} = S(\vec{\omega})\,R$$

In general, you simply form S from the angular velocity vector and don't need to differentiate the matrix.

angular velocity vector

$$S(\vec{a})\,\vec{p} = ? = \vec{a} \times \vec{p}$$



Uses for Skew-Symmetric Matrices

$$\dot{R}(t) = S(\vec{\omega}(t))R(t)$$

You can derive the fact that you can add angular velocity vectors by expressing them in the same frame.

$$\omega^0_{0,2} = \omega^0_{0,1} + R^0_1 \omega^1_{1,2}$$

The angular velocity of frame 2 relative to frame 0 is equal to the angular velocity of frame 1 relative to frame 0, expressed in frame 0, plus the angular velocity of frame 2 relative to frame 1, expressed in frame 0

You can calculate the velocity of a point that is fixed to a rotating (but not translating) frame.

$$\begin{aligned} p^0 &= R^0_1 p^1 \\ \frac{d}{dt}p^0 &= ? = \dot{R}^0_1 p^1 \\ &= S(\vec{\omega}) R^0_1 p^1 \\ &= \vec{\omega} \times R^0_1 p^1 \\ &= \vec{\omega} \times p^0 \end{aligned}$$

You can calculate the velocity of a point that is fixed to a rotating and translating frame.

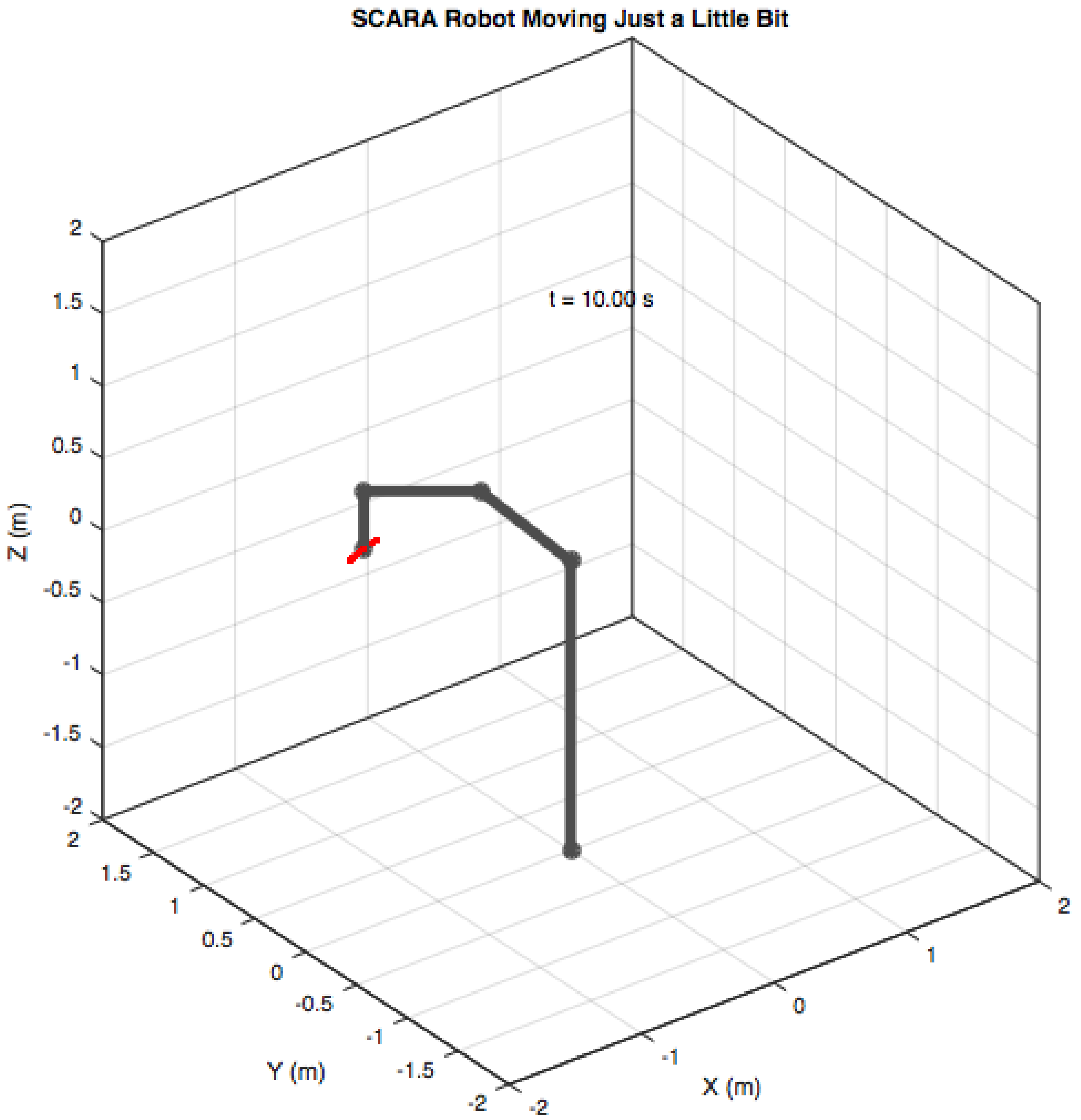
$$\begin{aligned} p^0 &= R^0_1(t) p^1 + o^0_1(t) \\ \dot{p}^0 &= \dot{R}^0_1 p^1 + \dot{o}^0_1 \\ \dot{p}^0 &= S(\omega^0) R^0_1 p^1 + \dot{o}^0_1 \\ \dot{p}^0 &= \omega^0 \times p^0 + \dot{o}^0_1 \end{aligned}$$

How do the velocities of the joints affect
the linear and angular velocity of the end-effector?

These quantities are related by the Jacobian,
a matrix that generalizes the notion of an ordinary
derivative of a scalar function.

Jacobians are useful for:

- planning and executing smooth trajectories
- determining singular configurations
- executing coordinated anthropomorphic motion
- deriving dynamic equations of motion
- transforming forces and torques from the end-effector to the manipulator joints.



What do you notice about how the tip moves when we actuate each joint individually?

The tip motion is approximately linear; ignore the curve and focus on the tangent.

Motion of a revolute joint makes the tip move in a circle around the joint axis; a larger radius creates faster motion for the same joint velocity.

Motion of a prismatic joint makes the tip move linearly along the joint axis at the joint speed.

Manipulator Jacobian

explore how changes in joint values affect the end-effector movement

could have N joints, but only six end-effector velocity terms

$$(v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$$

The Jacobian matrix lets us calculate how joint velocities turn into end-effector velocities; this mapping strongly depends on the robot's current configuration!

look at it in two parts: linear velocity and angular velocity

$$\boxed{v_n^0 = J_v \dot{q}} \qquad \omega_n^0 = J_\omega \dot{q}$$

How do we calculate the linear velocity Jacobian?

Differential Motion

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \vec{p} = f(\vec{q})$$

endpoint position

forward kinematics

joint values

nonlinear

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \dot{\vec{p}} = J_v(\vec{q})\,\dot{\vec{q}}$$

endpoint velocity

linear velocity Jacobian

joint values

joint velocity

linear approximation of fk derivative near a given point

For an n-dimensional joint space and a Cartesian workspace, the position Jacobian is a 3 x n matrix composed of the partial derivatives of the end-effector position with respect to each joint variable.

$$J_v(\vec{q}) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \cdots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \cdots & \frac{\partial y}{\partial q_n} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \cdots & \frac{\partial z}{\partial q_n} \end{bmatrix}$$

Another perspective:

$$x(t) = f(q_1(t), q_2(t), \ldots, q_n(t))$$

the time derivative can be found using

$$\frac{dx}{dt} = \sum_{i=1}^{n} \frac{\delta x}{\delta q_i} \frac{dq_i}{dt}$$

Using the Linear Velocity Jacobian

$$\dot{\vec{p}} = J_v(\vec{q})\,\dot{\vec{q}}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \cdots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \cdots & \frac{\partial y}{\partial q_n} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \cdots & \frac{\partial z}{\partial q_n} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

What units do the entries of the linear velocity Jacobian have?

Using the Linear Velocity Jacobian

When will the Jacobian be constant across poses?
a Cartesian robot

$$\dot{\vec{p}} = J_v(\vec{q})\,\dot{\vec{q}}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \cdots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \cdots & \frac{\partial y}{\partial q_n} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \cdots & \frac{\partial z}{\partial q_n} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

distance/time
such as m/s

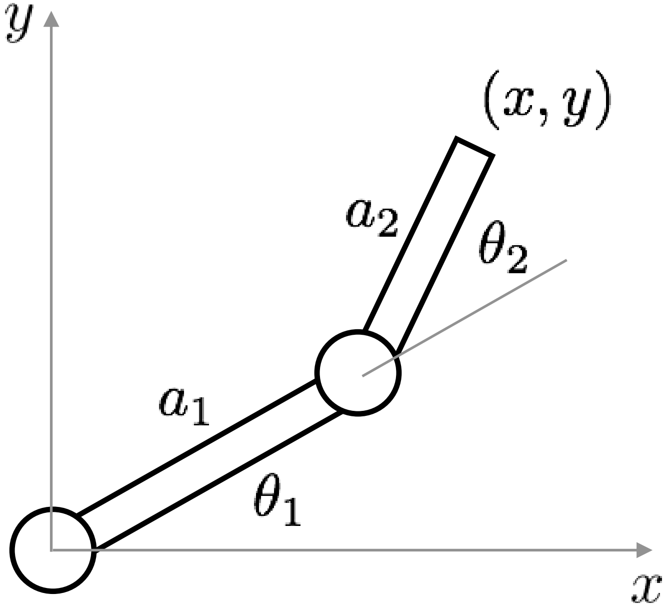
Evaluate the Jacobian at the
robot's current pose

revolute: angle/time such as rad/s
prismatic: distance/time such as
m/s

What units do the entries of the linear velocity Jacobian have?

if joint *i* is revolute, column *i* is in distance/angle such as m/rad or m
if joint *i* is prismatic and units match, column *i* is unitless

Example: Planar RR



$$J_v(\vec{q}) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \cdots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \cdots & \frac{\partial y}{\partial q_n} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \cdots & \frac{\partial z}{\partial q_n} \end{bmatrix}$$

From the forward kinematics, we can extract the symbolic tip position vector from the last column of the homogeneous transformation matrix:

$$d_2^0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_2 c_{12} + a_1 c_1 \\ a_2 s_{12} + a_1 s_1 \\ 0 \end{bmatrix}$$

Taking the partial derivative with respect to each joint variable produces the linear velocity Jacobian:

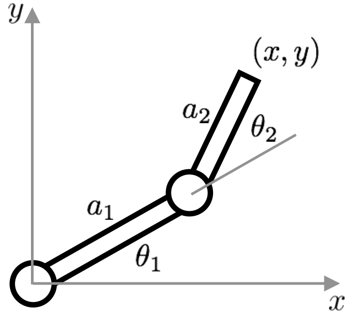
$$J_v(\vec{q}) = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} \\ 0 & 0 \end{bmatrix}$$

which relates instantaneous joint velocities to endpoint velocities:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

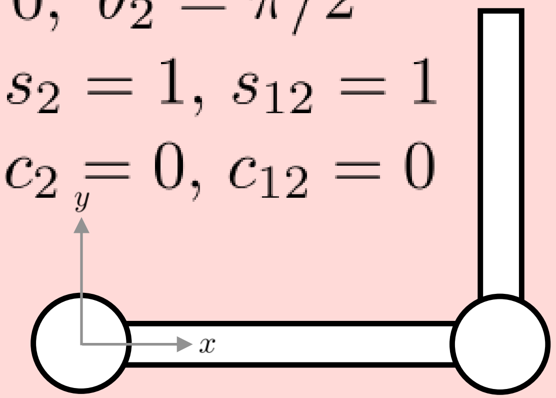
This mapping depends on the robot's current pose!

Example: Planar RR



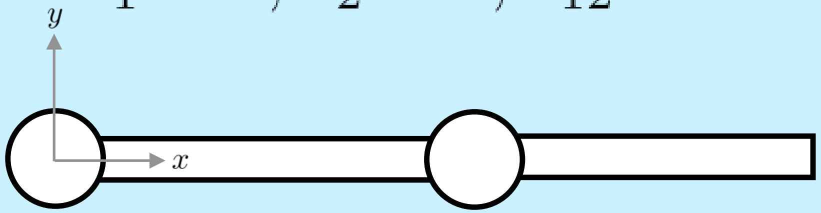
$$J_v(\vec{q}) = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} \\ 0 & 0 \end{bmatrix}$$

$$\theta_1 = 0,\ \theta_2 = \pi/2$$
$$s_1 = 0,\ s_2 = 1,\ s_{12} = 1$$
$$c_1 = 1,\ c_2 = 0,\ c_{12} = 0$$



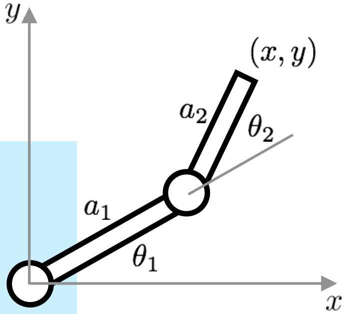
$$J_v([0 \ \ \pi/2]^T) = \begin{bmatrix} -a_2 & -a_2 \\ a_1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\theta_1 = 0,\ \theta_2 = 0$$
$$s_1 = 0,\ s_2 = 0,\ s_{12} = 0$$
$$c_1 = 1,\ c_2 = 1,\ c_{12} = 1$$

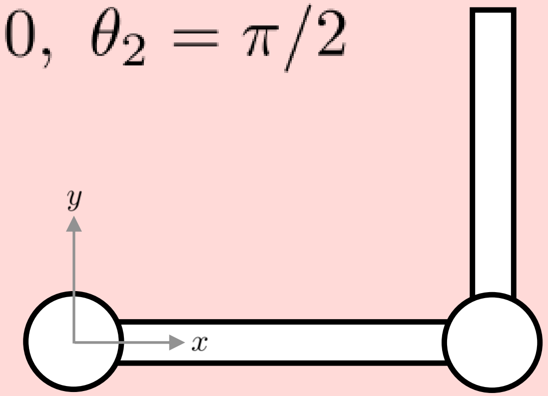


$$J_v([0 \ \ 0]^T) = \begin{bmatrix} 0 & 0 \\ a_1 + a_2 & a_2 \\ 0 & 0 \end{bmatrix}$$

Example: Planar RR

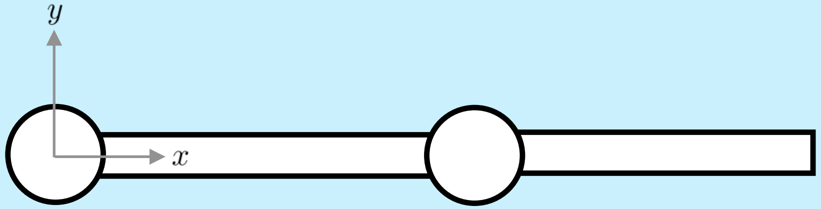


$$\theta_1 = 0,\ \theta_2 = \pi/2$$



$$J_v([0 \quad \pi/2]^T) = \begin{bmatrix} -a_2 & -a_2 \\ a_1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\theta_1 = 0,\ \theta_2 = 0$$



$$J_v([0 \quad 0]^T) = \begin{bmatrix} 0 & 0 \\ a_1 + a_2 & a_2 \\ 0 & 0 \end{bmatrix}$$

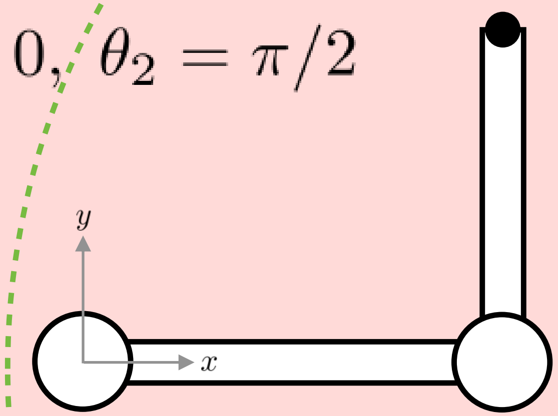
$$\dot{\vec{p}} = J_v(\vec{q})\,\dot{\vec{q}}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -a_2\dot{\theta}_1 - a_2\dot{\theta}_2 \\ a_1\dot{\theta}_1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ (a_1 + a_2)\dot{\theta}_1 + a_2\dot{\theta}_2 \\ 0 \end{bmatrix}$$

Example: Planar RR

$\theta_1 = 0,\ \theta_2 = \pi/2$

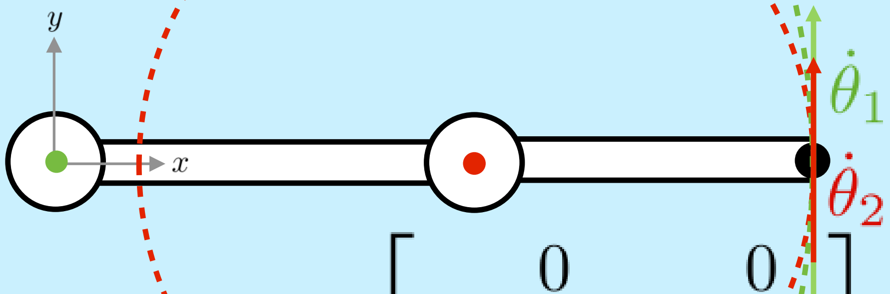


$$J_v([0 \quad \pi/2]^T) = \begin{bmatrix} -a_2 & -a_2 \\ a_1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -a_2\dot{\theta}_1 - a_2\dot{\theta}_2 \\ a_1\dot{\theta}_1 \\ 0 \end{bmatrix}$$

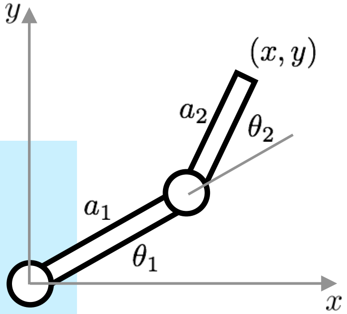
$$\dot{\vec{p}} = J_v(\vec{q})\,\dot{\vec{q}}$$

$\theta_1 = 0,\ \theta_2 = 0$



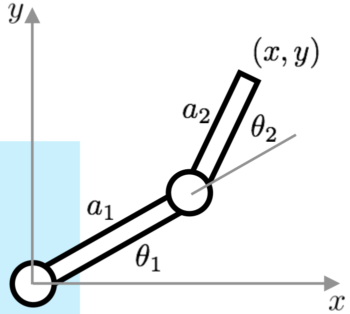
$$J_v([0 \quad 0]^T) = \begin{bmatrix} 0 & 0 \\ a_1 + a_2 & a_2 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ \boxed{(a_1 + a_2)\dot{\theta}_1} \boxed{+ a_2\dot{\theta}_2} \\ 0 \end{bmatrix}$$

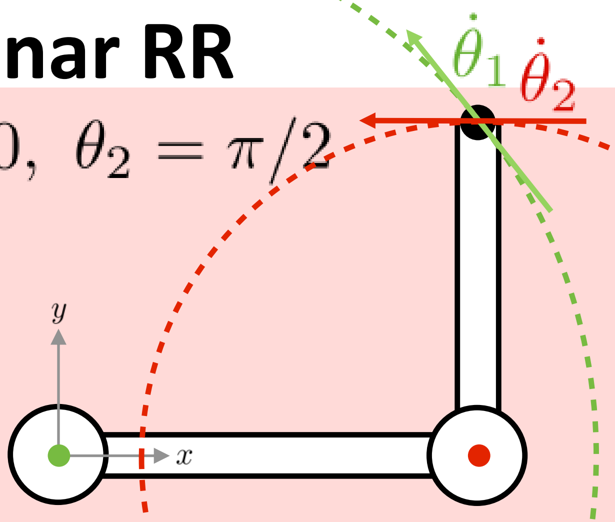


Notice anything else about this configuration?
The robot's tip cannot move in the x or z directions…

Example: Planar RR



$\theta_1 = 0,\ \theta_2 = \pi/2$



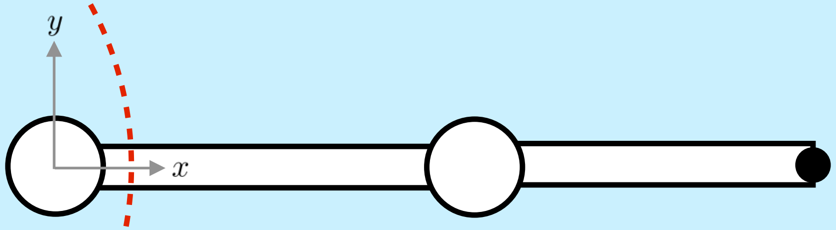
$$J_v([0 \quad \pi/2]^T) = \begin{bmatrix} -a_2 & -a_2 \\ a_1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \boxed{-a_2\dot{\theta}_1} \boxed{- a_2\dot{\theta}_2} \\ \boxed{a_1\dot{\theta}_1} \\ 0 \end{bmatrix}$$

The robot's tip cannot move in the z direction, but it can move in both x and y directions...

$$\dot{\vec{p}} = J_v(\vec{q})\,\dot{\vec{q}}$$

$\theta_1 = 0,\ \theta_2 = 0$

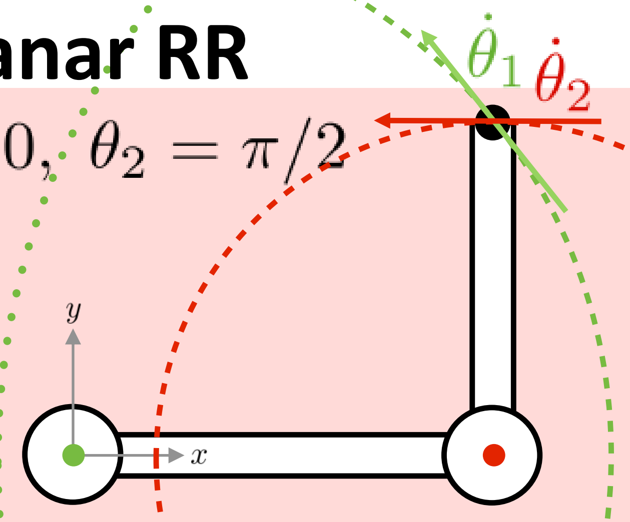


$$J_v([0 \quad 0]^T) = \begin{bmatrix} 0 & 0 \\ a_1 + a_2 & a_2 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ (a_1 + a_2)\dot{\theta}_1 + a_2\dot{\theta}_2 \\ 0 \end{bmatrix}$$

Example: Planar RR

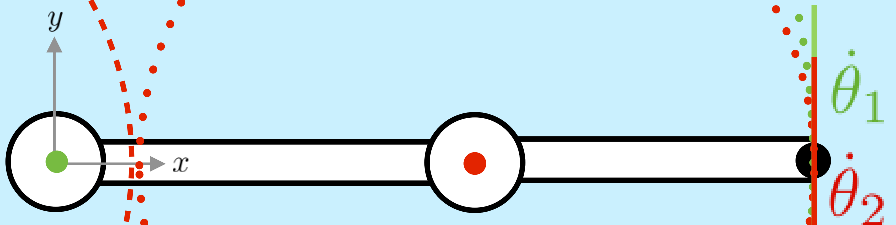
$\theta_1 = 0, \; \theta_2 = \pi/2$



$$J_v([0 \;\; \pi/2]^T) = \begin{bmatrix} -a_2 & -a_2 \\ a_1 & 0 \\ 0 & 0 \end{bmatrix}$$

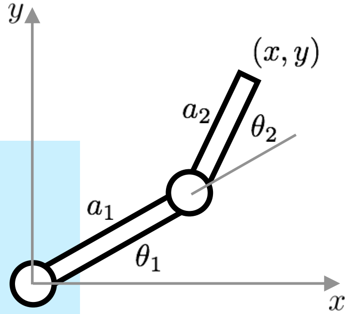
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \boxed{-a_2\dot{\theta}_1} \boxed{- a_2\dot{\theta}_2} \\ \boxed{a_1\dot{\theta}_1} \\ 0 \end{bmatrix}$$

$\theta_1 = 0, \; \theta_2 = 0$



$$J_v([0 \;\; 0]^T) = \begin{bmatrix} 0 & 0 \\ a_1 + a_2 & a_2 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ (a_1 + a_2)\dot{\theta}_1 + a_2\dot{\theta}_2 \\ 0 \end{bmatrix}$$



$$\dot{\vec{p}} = J_v(\vec{q})\,\dot{\vec{q}}$$

What questions do you have?

Another Way to Calculate the Linear Velocity Jacobian (SHV 4.6.2)

Prismatic Joints

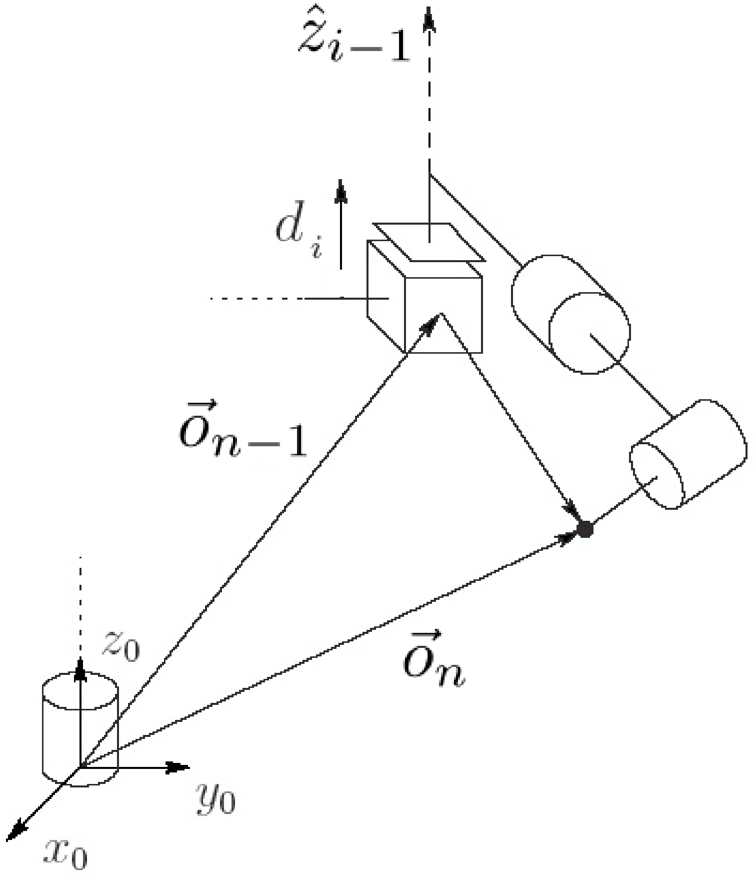


Figure 4.1: Motion of the end effector due to primsmatic joint i.

$$\dot{o}_n^0 = \dot{d}_i R_{i-1}^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \dot{d}_i z_{i-1}^0$$

$$J_{v_i} = \hat{z}_{i-1}$$

The orientation of a z-axis depends on the robot's pose if there are any revolute joints before it in the chain.

Another Way to Calculate the Linear Velocity Jacobian (SHV 4.6.2)

Revolute Joints

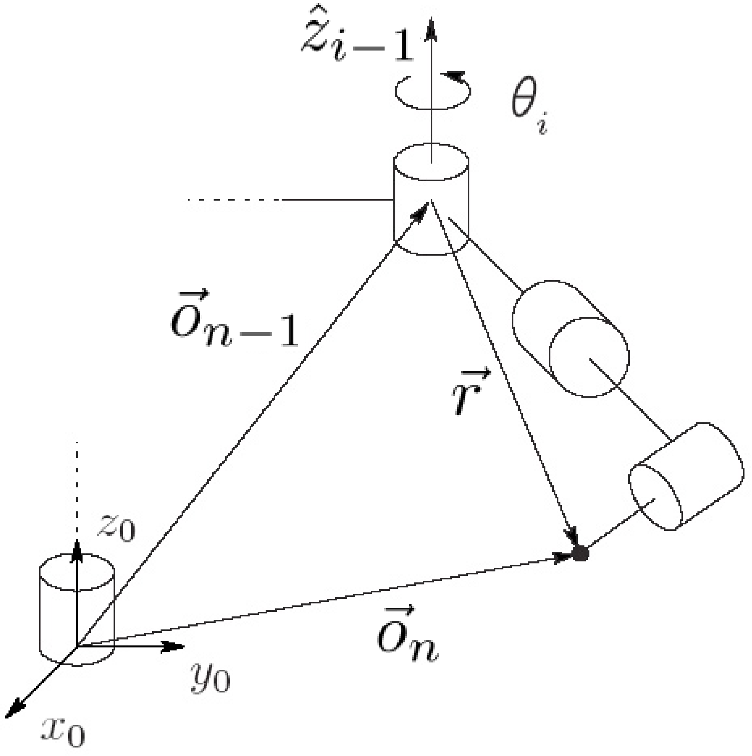


Figure 4.2: Motion of the end effector due to revolute joint i.

$$\vec{v} = \vec{\omega} \times \vec{r} = S(\vec{\omega})\vec{r}$$

$$\vec{\omega} = \dot{\theta}_i \hat{z}_{i-1}$$

$$\vec{r} = \vec{o}_n - \vec{o}_{i-1}$$

$$J_{v_i} = \hat{z}_{i-1} \times (\vec{o}_n - \vec{o}_{i-1})$$

Make sure these vectors are all expressed in the same frame before manipulating them!

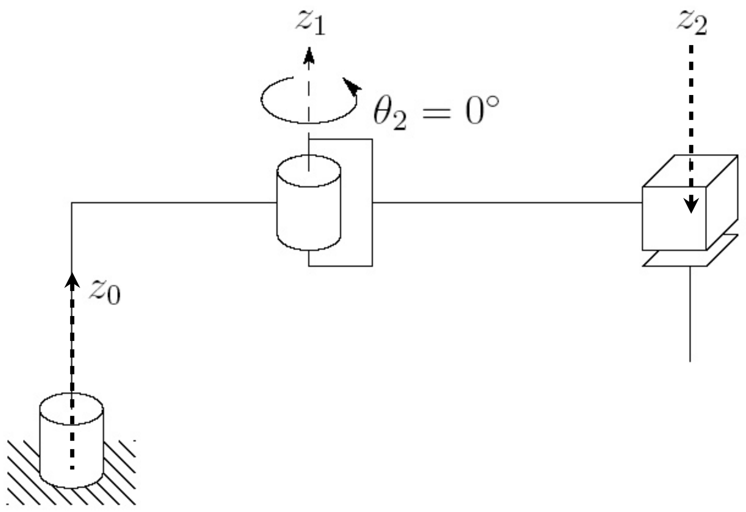
Example: SCARA

Prismatic Joints

$$J_{v_i} = \hat{z}_{i-1}$$

Revolute Joints

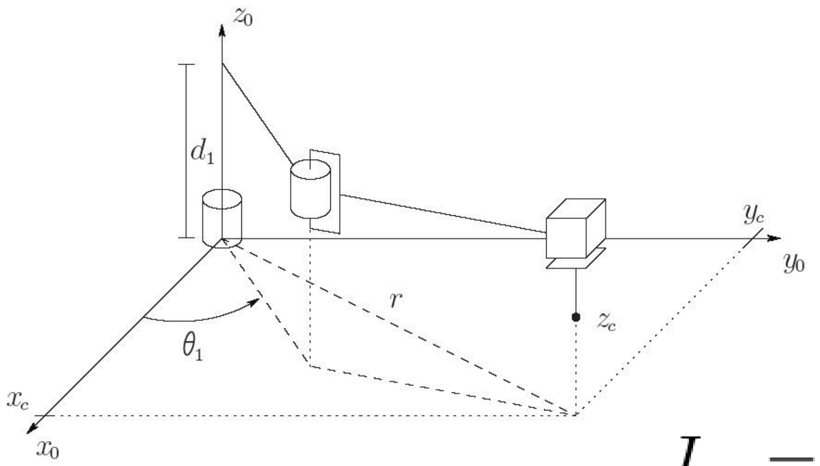
$$J_{v_i} = \hat{z}_{i-1} \times (\vec{o}_n - \vec{o}_{i-1})$$



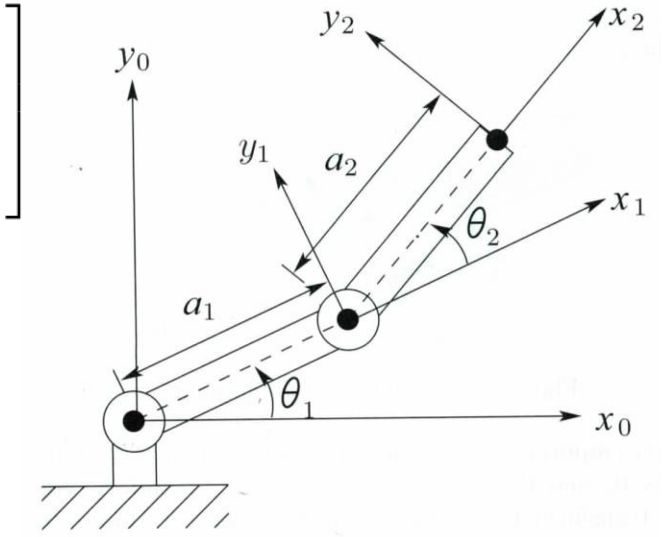
What is the SCARA's Jv?

$$J_v = \left[\quad \right]$$

Example: SCARA



$$T_3^0 = \begin{bmatrix} c_{12}^* & s_{12}^* & 0 & a_1 c_1^* + a_2 c_{12}^* \\ s_{12}^* & -c_{12}^* & 0 & a_1 s_1^* + a_2 s_{12}^* \\ 0 & 0 & -1 & -d_3^* \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$J_{v_i} = \hat{z}_{i-1} \times (\vec{o}_n - \vec{o}_{i-1})$$

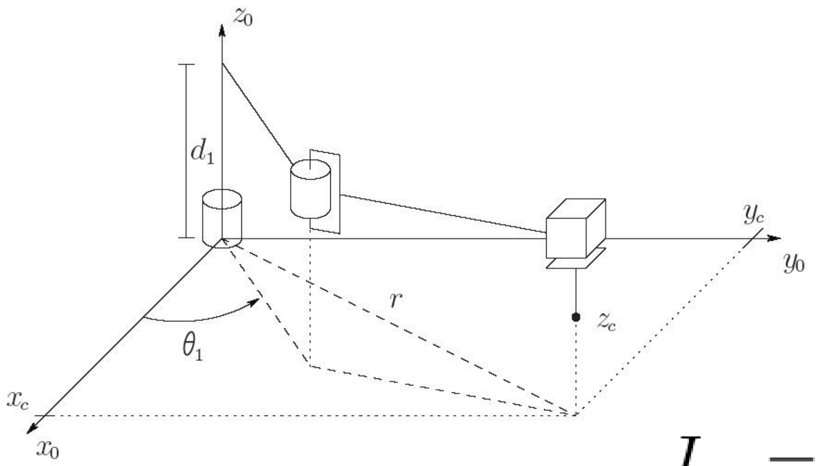
$$J_{v_2} = \hat{z}_1 \times (\vec{o}_3 - \vec{o}_1)$$

write all vectors in frame zero!

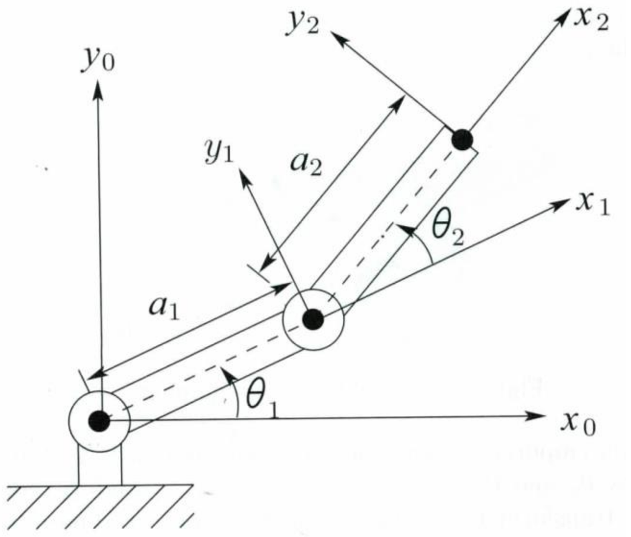
$$J_{v_2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} a_1 c_1 + a_2 c_{12} \\ a_1 s_1 + a_2 s_{12} \\ -d_3 \end{bmatrix} - \begin{bmatrix} a_1 c_1 \\ a_1 s_1 \\ 0 \end{bmatrix} \right)$$

$$J_{v_2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} a_2 c_{12} \\ a_2 s_{12} \\ -d_3 \end{bmatrix} = \begin{bmatrix} -a_2 s_{12} \\ a_2 c_{12} \\ 0 \end{bmatrix}$$

Example: SCARA



$$T_3^0 = \begin{bmatrix} c_{12}^* & s_{12}^* & 0 & a_1c_1^* + a_2c_{12}^* \\ s_{12}^* & -c_{12}^* & 0 & a_1s_1^* + a_2s_{12}^* \\ 0 & 0 & -1 & -d_3^* \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$J_{v_i} = \hat{z}_{i-1} \times (\vec{o}_n - \vec{o}_{i-1})$$

$$J_{v_1} = \hat{z}_0 \times (\vec{o}_3 - \vec{o}_0)$$

write all vectors in frame zero!

$$J_{v_1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left(\begin{bmatrix} a_1c_1 + a_2c_{12} \\ a_1s_1 + a_2s_{12} \\ -d_3 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right)$$

$$J_{v_1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} a_1c_1 + a_2c_{12} \\ a_1s_1 + a_2s_{12} \\ -d_3 \end{bmatrix} = \begin{bmatrix} -a_1s_1 - a_2s_{12} \\ a_1c_1 + a_2c_{12} \\ 0 \end{bmatrix}$$

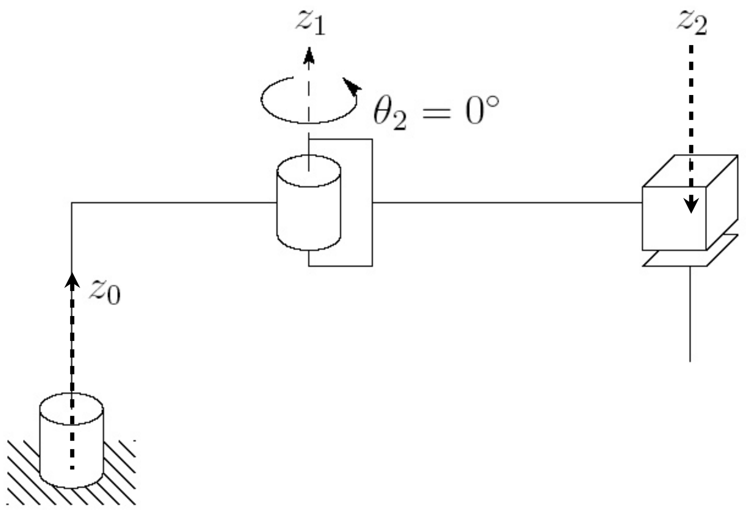
Example: SCARA

Prismatic Joints

$$J_{v_i} = \hat{z}_{i-1}$$

Revolute Joints

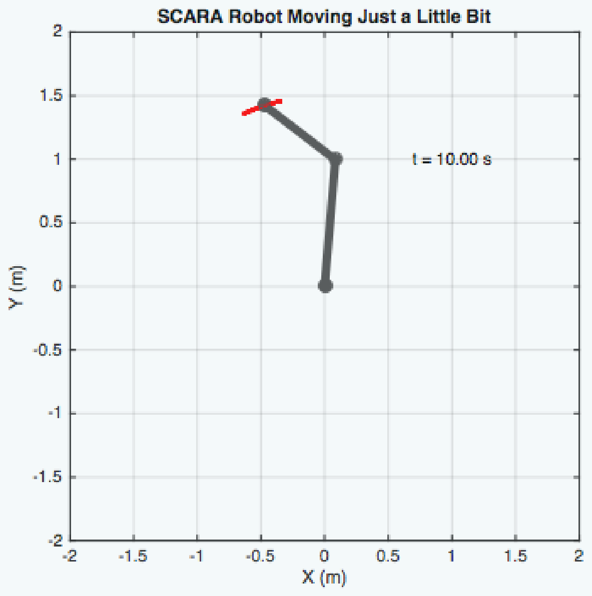
$$J_{v_i} = \hat{z}_{i-1} \times (\vec{o}_n - \vec{o}_{i-1})$$



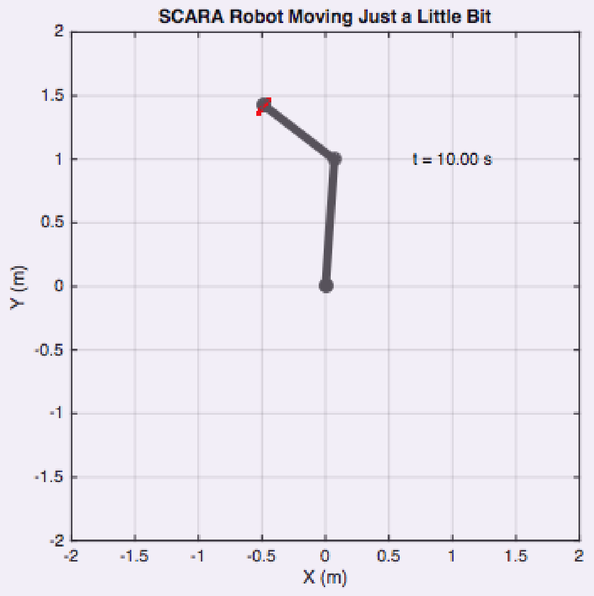
What is the SCARA's Jv?

$$J_v = \left[\begin{array}{ccc} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} & 0 \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} & 0 \\ 0 & 0 & -1 \end{array} \right]$$

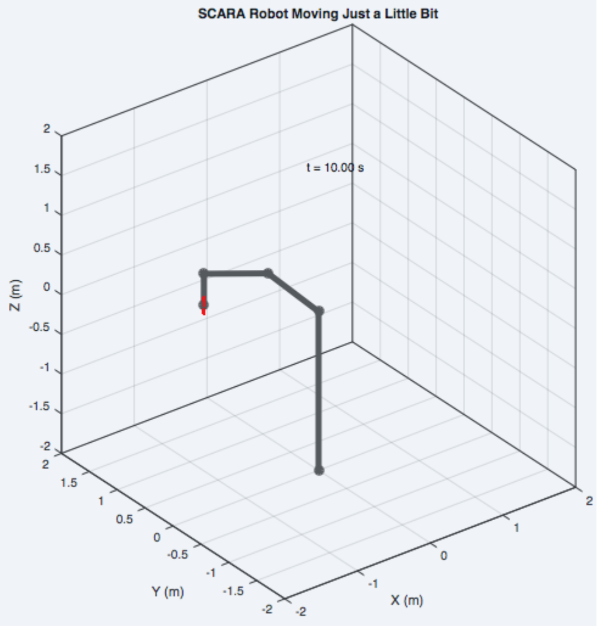
$$J_v = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} & 0 \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} & 0 \\ 0 & 0 & -1 \end{bmatrix}$$



Joint 1

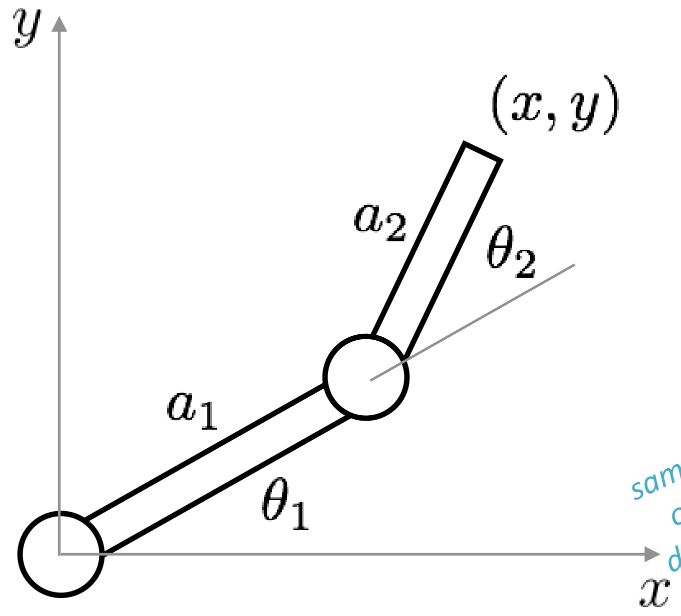


Joint 2



Joint 3

Example: Planar RR



$$J_v(\vec{q}) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \cdots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \cdots & \frac{\partial y}{\partial q_n} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \cdots & \frac{\partial z}{\partial q_n} \end{bmatrix}$$

From the forward kinematics, we can extract the symbolic tip position vector from the last column of the homogeneous transformation matrix:

$$d_2^0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_2 c_{12} + a_1 c_1 \\ a_2 s_{12} + a_1 s_1 \\ 0 \end{bmatrix}$$

same as the first two columns of the Jv we calculated via differentiation for a planar RR

Taking the partial derivative with respect to each joint variable produces the linear velocity Jacobian:

$$J_v(\vec{q}) = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} \\ 0 & 0 \end{bmatrix}$$

which relates instantaneous joint velocities to endpoint velocities:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

This mapping depends on the robot's current pose!

Linear Velocity Jacobians: Methods

How do the velocities of the joints affect the linear velocity of the end-effector?

$$J_v(\vec{q}) = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \cdots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \cdots & \frac{\partial y}{\partial q_n} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \cdots & \frac{\partial z}{\partial q_n} \end{bmatrix}$$

$$v_n^0 = J_v \dot{q}$$

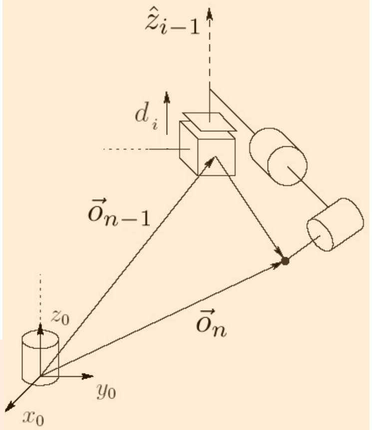
(3 x 1) (3 x n) (n x 1)

n joints

Two ways to get J_v

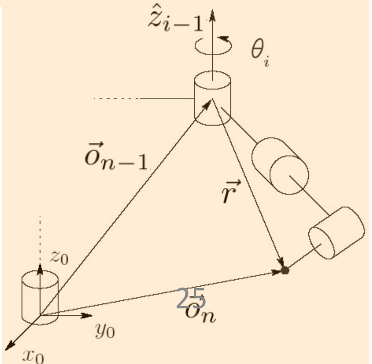
partial derivatives of the tip position with respect to the joint variables

geometric construction of the columns of Jv using the robot's forward kinematics



Prismatic $J_{v_i} = z_{i-1}$

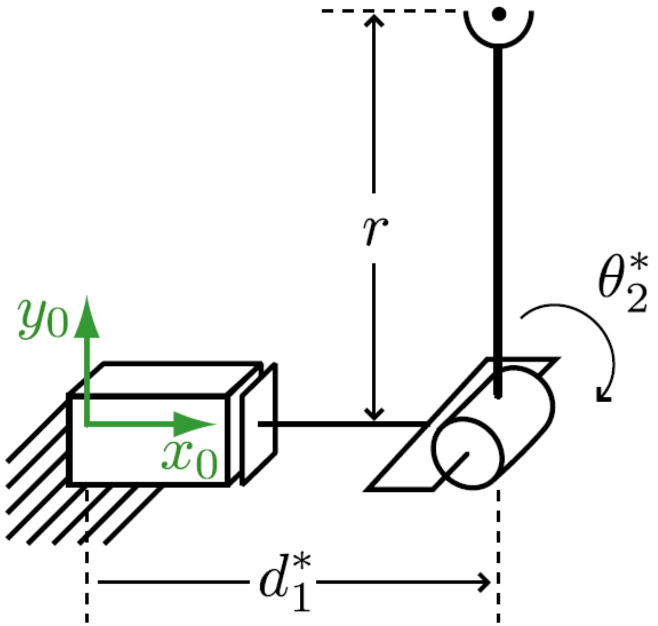
Revolute $J_{v_i} = z_{i-1} \times (o_n - o_{i-1})$



Both methods yield the same Jv matrix

I prefer to calculate linear velocity Jacobians by using geometric construction, but both approaches are valid, enabling you to check your work and increase your intuition.

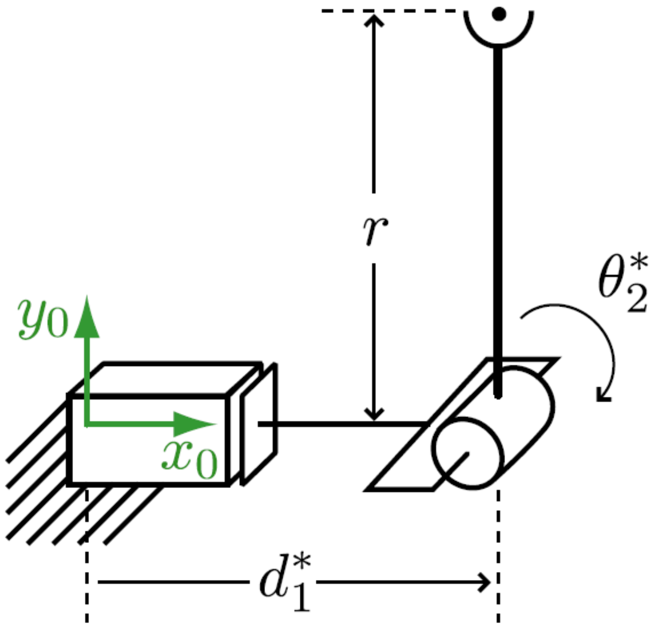
Example 2: PR Manipulator



$$p^0 = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} d_1^* + r\sin\theta_2^* \\ r\cos\theta_2^* \end{bmatrix}$$

Calculate the linear velocity Jacobian for this robot.

Example 2: PR Manipulator



$$p^0 = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} d_1^* + r\sin\theta_2^* \\ r\cos\theta_2^* \end{bmatrix}$$

Calculate the linear velocity Jacobian for this robot.

$$J_v = ?$$

$$J_v = \begin{bmatrix} \partial x/\partial d_1^* & \partial x/\partial \theta_2^* \\ \partial y/\partial d_1^* & \partial y/\partial \theta_2^* \end{bmatrix}$$

$$\boxed{J_v = \begin{bmatrix} 1 & r\cos\theta_2^* \\ 0 & -r\sin\theta_2^* \end{bmatrix}}$$

Check?

Prismatic
$J_{v_i} = z_{i-1}$

Revolute
$J_{v_i} = z_{i-1} \times (o_n - o_{i-1})$

Why Cynthia asks you not to use the syms toolbox

```matlab
% Create symbolic real-valued variables.
syms theta1 d2 a1 l2 real

% Enter x and y coordinates of tip position.
x = a1*cos(theta1) - (l2+d2)*sin(theta1)
y = a1*sin(theta1) + (l2+d2)*cos(theta1)

% Calculate linear velocity Jacobian from x and y
coordinates.
Jv = [diff(x,theta1) diff(x,d2);
      diff(y,theta1) diff(y,d2)]

% Evaluate
tic
for i = 1:1000
    subs(Jv,{theta1, d2, a1, l2}, [0, 1, 1, 1]);
end
toc
```

Elapsed time is 8.453685 seconds.

```matlab
% Evaluate
tic
for i = 1:1000
    linearJac(0, 1, 1, 1);
end
toc

% linear Jacobian written as a function
function Jv = linearJac(theta1, d2, a1, l2)

Jv = [-cos(theta1)*(d2+l2)-a1*sin(theta1), ...
      -sin(theta1);
      a1*cos(theta1)-sin(theta1)*(d2+l2), ...
      cos(theta1)];
end
```

Elapsed time is 0.003378 seconds.

As always, the syms toolbox will not be considered a valid methodological approach on Lab 4.

N-link Manipulators

$$\mathbf{T}_{\mathrm{n}}^{0} = \mathbf{T}_1^0(q_1)\mathbf{T}_2^1(q_2)\mathbf{T}_3^2(q_3)\ldots$$

i^{th} column of J_v is:

$$\frac{\partial}{\partial q_i}\left(\mathbf{T}_{\mathrm{n}}^{0}\begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right) = \frac{\partial}{\partial q_i}\left(\mathbf{T}_1^0(q_1)\mathbf{T}_2^1(q_2)\mathbf{T}_3^2(q_3)\ldots\begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right) = \boxed{\mathbf{T}_1^0\ldots\frac{\partial}{\partial q_i}\left(\mathbf{T}_i^{i-1}\right)\mathbf{T}_{i+1}^{i}\ldots\begin{bmatrix}0\\0\\0\\1\end{bmatrix}}$$

$$= \boxed{\frac{\partial}{\partial q_i}\left(\mathbf{T}_1^0(q_1)\right)}\mathbf{T}_2^1(q_2)\mathbf{T}_3^2(q_3)\ldots\begin{bmatrix}0\\0\\0\\1\end{bmatrix} + \mathbf{T}_1^0(q_1)\boxed{\frac{\partial}{\partial q_i}\left(\mathbf{T}_2^1(q_2)\right)}\mathbf{T}_3^2(q_3)\ldots\begin{bmatrix}0\\0\\0\\1\end{bmatrix} + \cdots$$

$$\cancel{+\mathbf{T}_1^0(q_1)\mathbf{T}_2^1(q_2)\mathbf{T}_3^2(q_3)\ldots\frac{\partial}{\partial q_i}\begin{bmatrix}0\\0\\0\\1\end{bmatrix}}$$

Manipulator Jacobian

explore how changes in joint values affect the end-effector movement

could have N joints, but only six end-effector velocity terms

$$(v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$$

The Jacobian matrix lets us calculate how joint velocities turn into end-effector velocities; this mapping strongly depends on the robot's current configuration!

look at it in two parts: linear velocity and angular velocity

$$v_n^0 = J_v \dot{q}$$

$$\omega_n^0 = J_\omega \dot{q}$$

How do we calculate the angular velocity Jacobian?

Angular Velocity

angular velocity
Jacobian,
evaluated at
the robot's
current pose

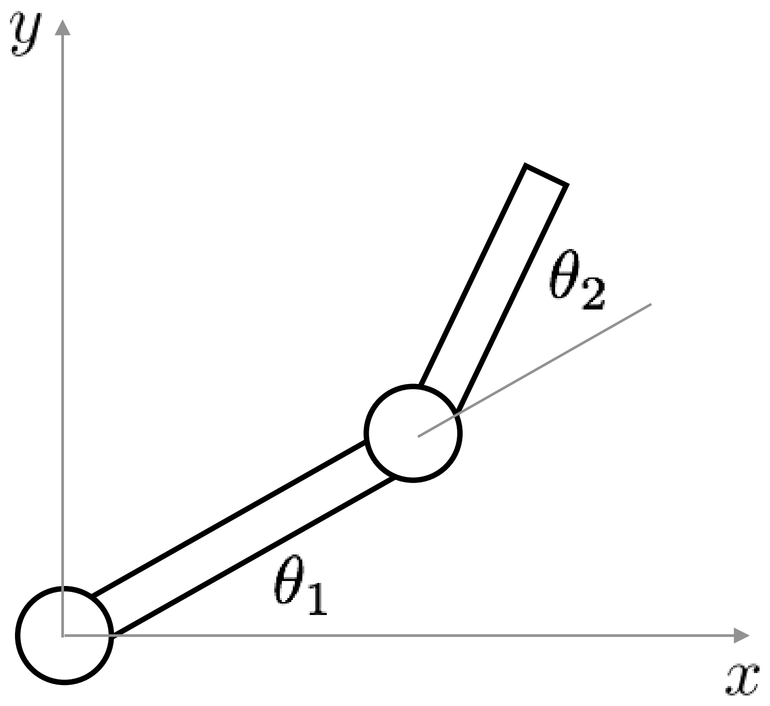
$$\omega = \mathbf{J}_\omega(q)\ \dot{\mathbf{q}}$$

final frame
angular
velocity

joint
velocities

Notation: $\omega_{i,j}^{k}$ this is the angular velocity of frame j with respect to frame i, expressed in frame k

Angular Velocity of Connected Rigid Bodies



$$\omega_{0,1}^{0} = 0\,\hat{x}_0 + 0\,\hat{y}_0 + \dot{\theta}_1\,\hat{z}_0$$

$$\omega_{1,2}^{1} = 0\,\hat{x}_1 + 0\,\hat{y}_1 + \dot{\theta}_2\,\hat{z}_1$$

$$\omega_{1,2}^{0} = \mathbf{R}_1^0\,\omega_{1,2}^{1}$$

$$\begin{aligned}\omega_{0,2}^{0} &= \omega_{0,1}^{0} + \mathbf{R}_1^0\,\omega_{1,2}^{1} \\ &= 0\,\hat{x}_0 + 0\,\hat{y}_0 + (\dot{\theta}_1 + \dot{\theta}_2)\,\hat{z}_0\end{aligned}$$

$$\omega_{0,n}^{0} = \sum_{i=1}^{n} \mathbf{R}_{i-1}^{0}\,\omega_{i-1,i}^{i-1}$$

$$\omega_{0,n}^{0} = \sum_{i=1}^{n} (\mathbf{R}_{i-1}^{0}\hat{\mathbf{z}})\,\dot{\theta}_i$$

note: this holds for revolute joints only (by definition, a prismatic joint cannot create angular velocity)

Angular Velocity Jacobian

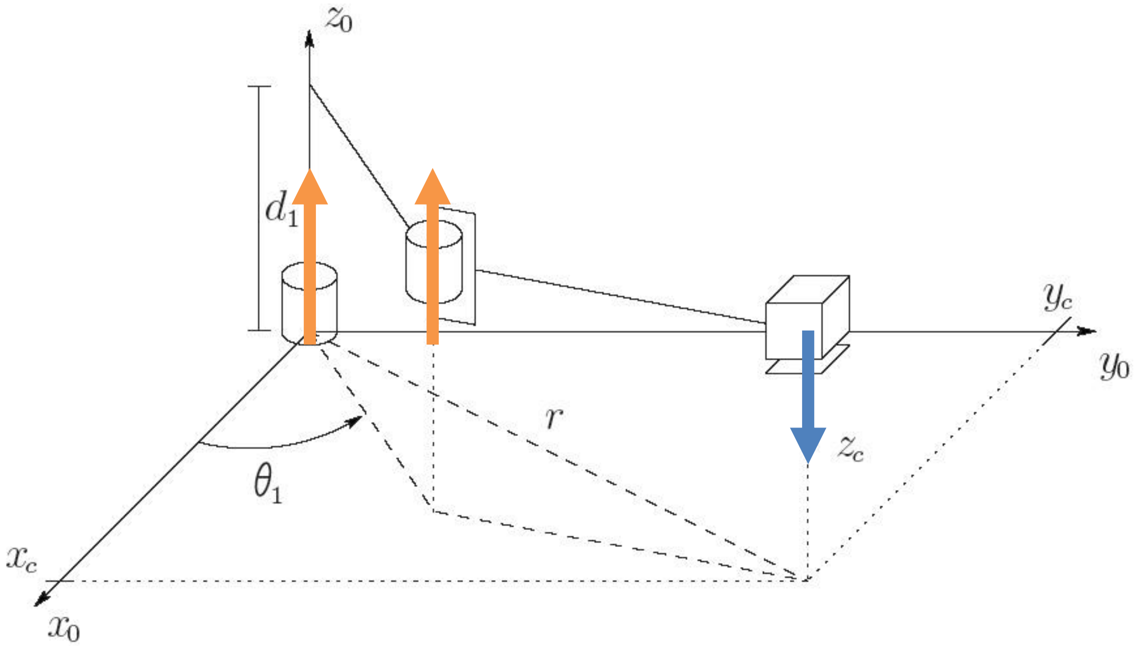
$$\omega_{0,n}^{0} = \sum_{i=1}^{n} \rho_i (\mathbf{R}_{i-1}^{0} \hat{z})\, \dot{\theta}_i$$

$\rho_i =$ 0 for prismatic
1 for revolute

$$\omega_{0,n}^{0} = [\rho_1 \hat{\mathbf{z}} \quad \rho_2 \mathbf{R}_1^0 \hat{\mathbf{z}} \quad \rho_3 \mathbf{R}_2^0 \hat{\mathbf{z}} \quad \cdots \quad \rho_n \mathbf{R}_{n-1}^0 \hat{\mathbf{z}}] \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_n \end{bmatrix}$$

$$\omega = J_\omega(q)\dot{q}$$

Example: SCARA

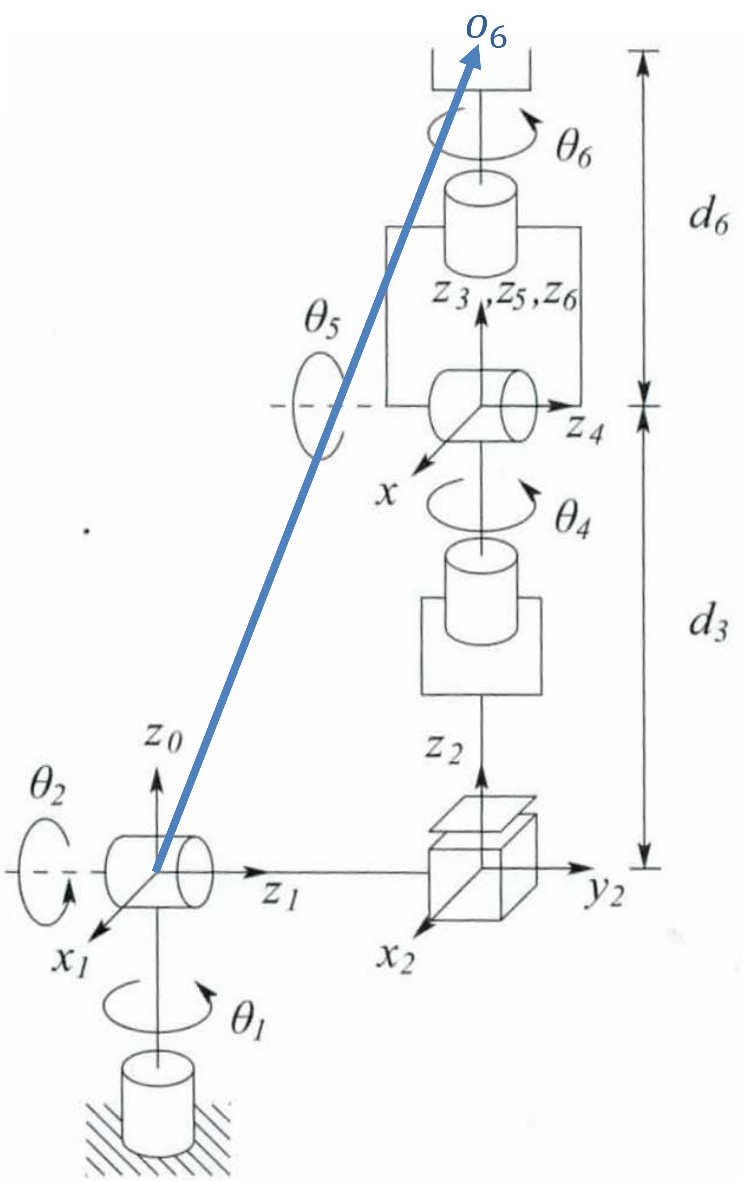


$$J_\omega = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

Axes of rotation for revolute joints

No angular velocity resulting from prismatic joints

Example: Stanford Arm



Prismatic

$$J_{vi} = z_{i-1}^0, J_{\omega i} = 0$$

Revolute

$$J_{vi} = S(z_{i-1}^0)(o_n^0 - o_{i-1}^0), J_{\omega i} = z_{i-1}^0$$

Link	d_i	a_i	α_i	θ_i
1	0	0	−90	$\theta^\star$
2	d_2	0	+90	$\theta^\star$
3	$d^\star$	0	0	0
4	0	0	−90	$\theta^\star$
5	0	0	+90	$\theta^\star$
6	d_6	0	0	$\theta^\star$

$$A_1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

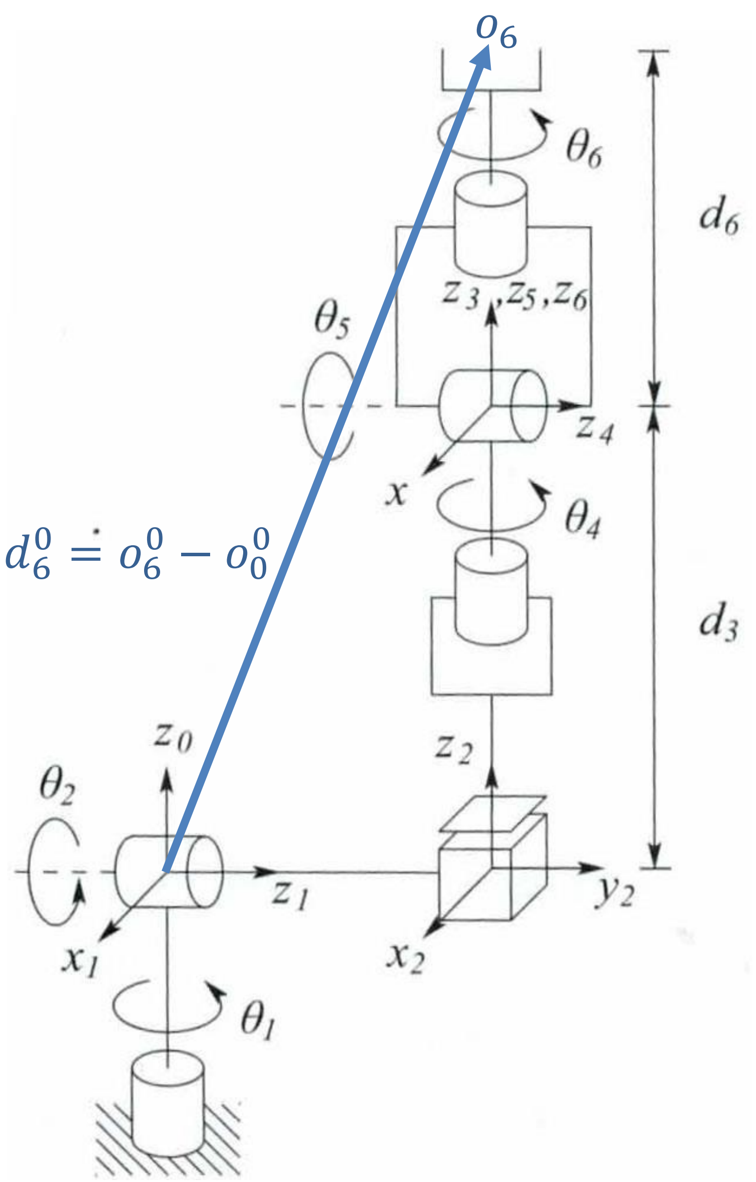
$$A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned}
r_{11} &= c_1[c_2(c_4c_5c_6 - s_4s_6) - s_2s_5c_6] - d_2(s_4c_5c_6 + c_4s_6) \\
r_{21} &= s_1[c_2(c_4c_5c_6 - s_4s_6) - s_2s_5c_6] + c_1(s_4c_5c_6 + c_4s_6) \\
r_{31} &= -s_2(c_4c_5c_6 - s_4s_6) - c_2s_5c_6 \\
r_{12} &= c_1[-c_2(c_4c_5s_6 + s_4c_6) + s_2s_5s_6] - s_1(-s_4c_5s_6 + c_4c_6) \\
r_{22} &= -s_1[-c_2(c_4c_5s_6 + s_4c_6) + s_2s_5s_6] + c_1(-s_4c_5s_6 + c_4c_6) \\
r_{32} &= s_2(c_4c_5s_6 + s_4c_6) + c_2s_5s_6 \\
r_{13} &= c_1(c_2c_4s_5 + s_2c_5) - s_1s_4s_5 \\
r_{23} &= s_1(c_2c_4s_5 + s_2c_5) + c_1s_4s_5 \\
r_{33} &= -s_2c_4s_5 + c_2c_5 \\
d_x &= c_1s_2d_3 - s_1d_2 + +d_6(c_1c_2c_4s_5 + c_1c_5s_2 - s_1s_4s_5) \\
d_y &= s_1s_2d_3 + c_1d_2 + d_6(c_1s_4s_5 + c_2c_4s_1s_5 + c_5s_1s_2) \\
d_z &= c_2d_3 + d_6(c_2c_5 - c_4s_2s_5)
\end{aligned}$$

Example: Stanford Arm



Prismatic

$$J_{vi} = z_{i-1}^0, J_{\omega i} = 0$$

Revolute

$$J_{vi} = S(z_{i-1}^0)(o_n^0 - o_{i-1}^0), J_{\omega i} = z_{i-1}^0$$

Link	d_i	a_i	α_i	θ_i
1	0	0	−90	$\theta^\star$
2	d_2	0	+90	$\theta^\star$
3	$d^\star$	0	0	0
4	0	0	−90	$\theta^\star$
5	0	0	+90	$\theta^\star$
6	d_6	0	0	$\theta^\star$

Joint 1

$$J_{v1} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} o_6^0 \qquad J_{\omega 1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

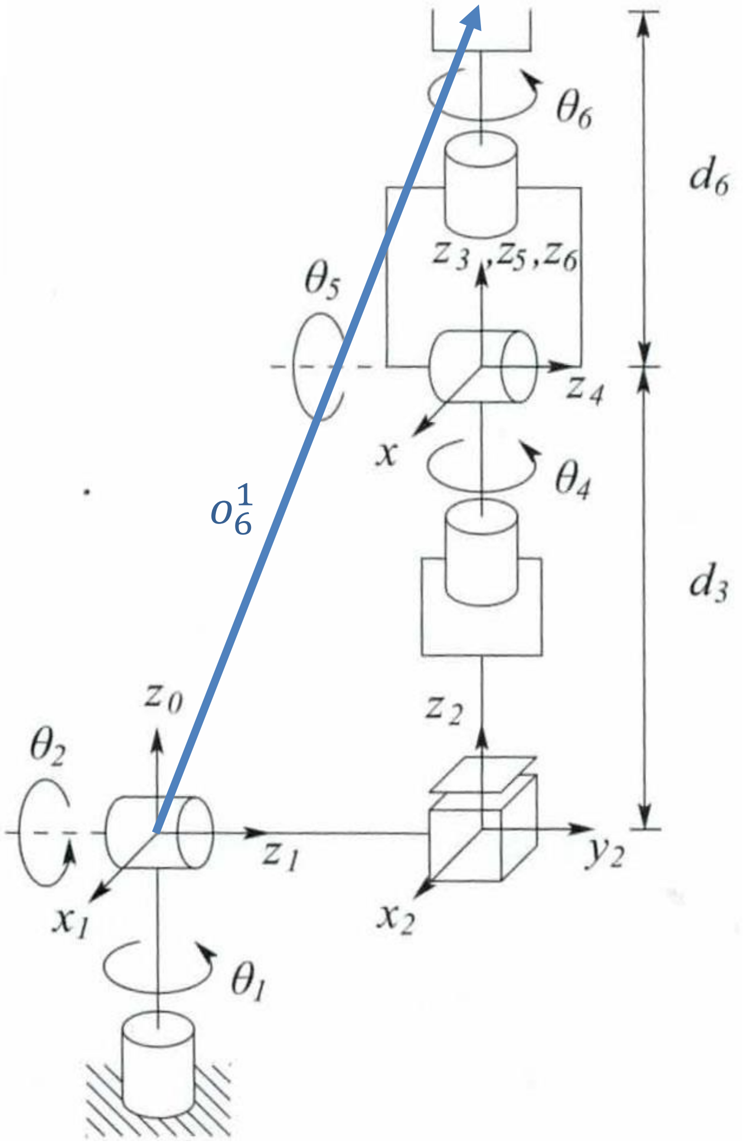
$$A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} d_x &=& c_1 s_2 d_3 - s_1 d_2 + + d_6(c_1 c_2 c_4 s_5 + c_1 c_5 s_2 - s_1 s_4 s_5) \\ d_y &=& s_1 s_2 d_3 + c_1 d_2 + d_6(c_1 s_4 s_5 + c_2 c_4 s_1 s_5 + c_5 s_1 s_2) \\ d_z &=& c_2 d_3 + d_6(c_2 c_5 - c_4 s_2 s_5) \end{bmatrix}$$

Example: Stanford Arm



Prismatic

$$J_{vi} = z_{i-1}^0, J_{\omega i} = 0$$

Revolute

$$J_{vi} = S(z_{i-1}^0)(o_n^0 - o_{i-1}^0), J_{\omega i} = z_{i-1}^0$$

Link	d_i	a_i	α_i	θ_i
1	0	0	−90	$\theta^\star$
2	d_2	0	+90	$\theta^\star$
3	$d^\star$	0	0	0
4	0	0	−90	$\theta^\star$
5	0	0	+90	$\theta^\star$
6	d_6	0	0	$\theta^\star$

Joint 1

$$J_{v1} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} o_6^0 \qquad J_{\omega 1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Joint 2

$$J_{v2} = \begin{bmatrix} 0 & 0 & c_1 \\ 0 & 0 & s_1 \\ -c_1 & -s_1 & 0 \end{bmatrix} R_1^0 o_6^1 \qquad J_{\omega 2} = R_1^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} d_x &= c_1 s_2 d_3 - s_1 d_2 + +d_6(c_1 c_2 c_4 s_5 + c_1 c_5 s_2 - s_1 s_4 s_5) \\ d_y &= s_1 s_2 d_3 + c_1 d_2 + d_6(c_1 s_4 s_5 + c_2 c_4 s_1 s_5 + c_5 s_1 s_2) \\ d_z &= c_2 d_3 + d_6(c_2 c_5 - c_4 s_2 s_5) \end{aligned}$$

Example: Stanford Arm

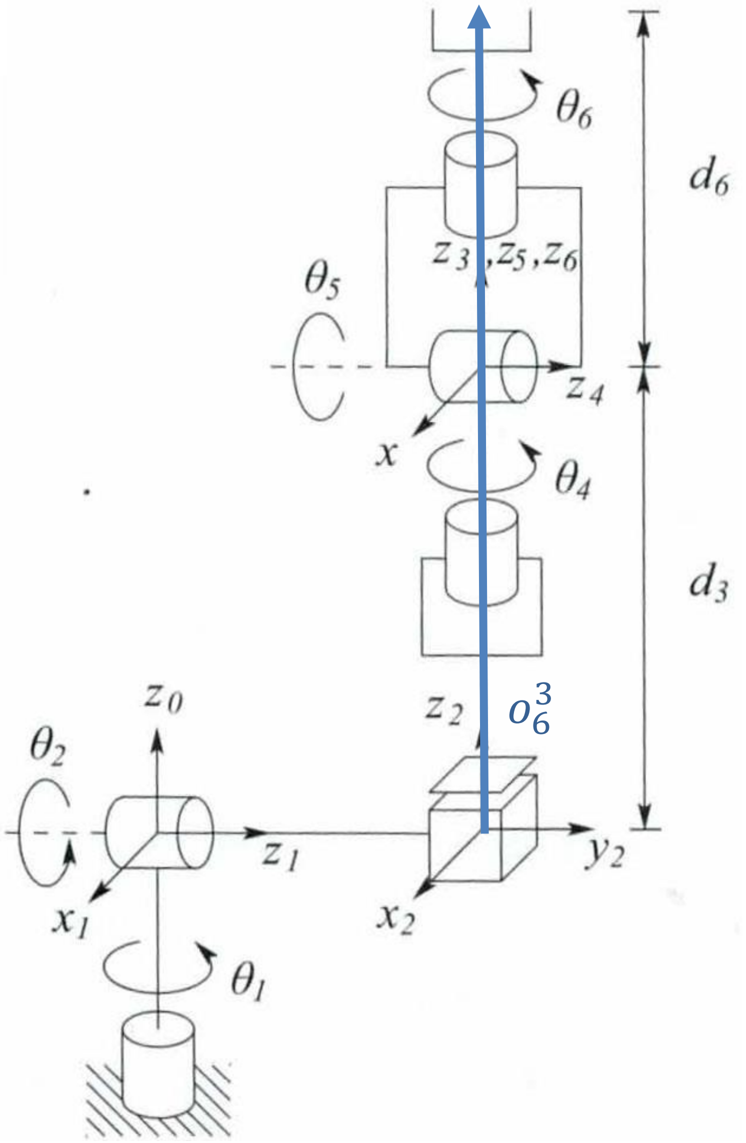
Prismatic

$$J_{vi} = z_{i-1}^0, J_{\omega i} = 0$$

Revolute

$$J_{vi} = S(z_{i-1}^0)(o_n^0 - o_{i-1}^0), J_{\omega i} = z_{i-1}^0$$

Link	d_i	a_i	α_i	θ_i
1	0	0	-90	$\theta^\star$
2	d_2	0	$+90$	$\theta^\star$
3	$d^\star$	0	0	0
4	0	0	-90	$\theta^\star$
5	0	0	$+90$	$\theta^\star$
6	d_6	0	0	$\theta^\star$



Joint 1

$$J_{v1} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} o_6^0 \qquad J_{\omega 1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Joint 2

$$J_{v2} = \begin{bmatrix} 0 & 0 & c_1 \\ 0 & 0 & s_1 \\ -c_1 & -s_1 & 0 \end{bmatrix} R_1^0 o_6^1 \qquad J_{\omega 2} = R_1^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Joint 3

$$J_{v3} = R_3^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad J_{\omega 3} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned} d_x &= c_1 s_2 d_3 - s_1 d_2 + +d_6(c_1 c_2 c_4 s_5 + c_1 c_5 s_2 - s_1 s_4 s_5) \\ d_y &= s_1 s_2 d_3 + c_1 d_2 + d_6(c_1 s_4 s_5 + c_2 c_4 s_1 s_5 + c_5 s_1 s_2) \\ d_z &= c_2 d_3 + d_6(c_2 c_5 - c_4 s_2 s_5) \end{aligned}$$

$$A_1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

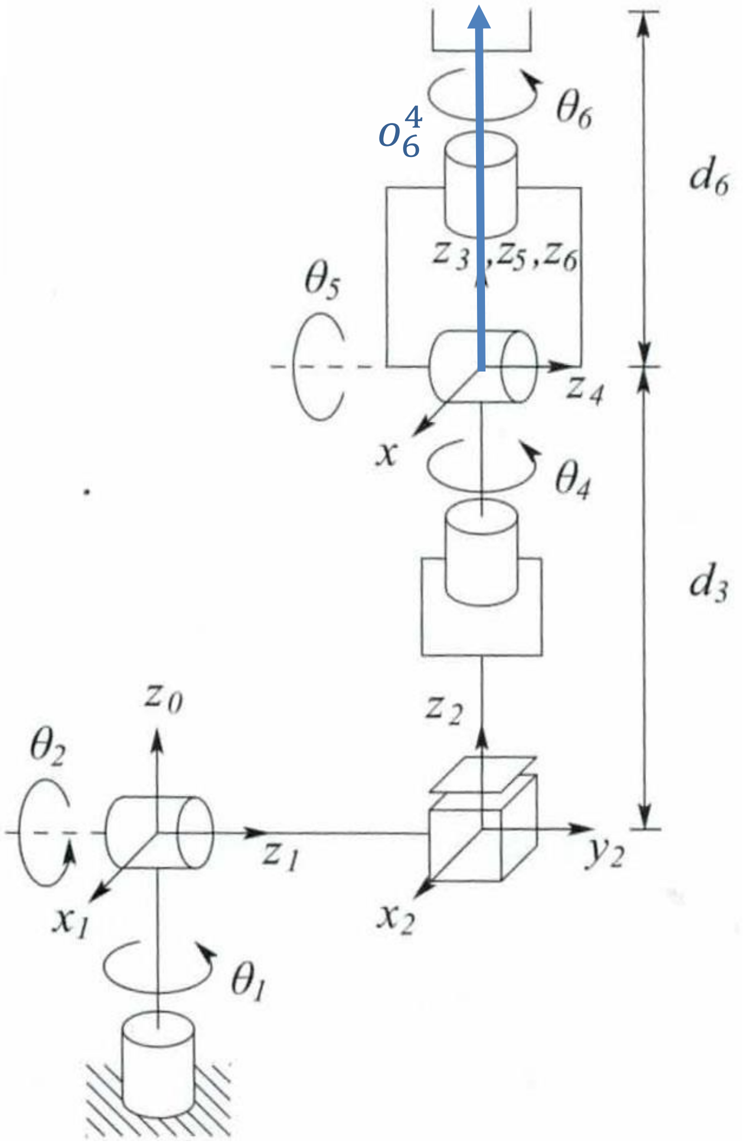
$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example: Stanford Arm



Prismatic

$$J_{vi} = z_{i-1}^0, J_{\omega i} = 0$$

Revolute

$$J_{vi} = S(z_{i-1}^0)(o_n^0 - o_{i-1}^0), J_{\omega i} = z_{i-1}^0$$

Link	d_i	a_i	α_i	θ_i
1	0	0	-90	$\theta^\star$
2	d_2	0	$+90$	$\theta^\star$
3	$d^\star$	0	0	0
4	0	0	-90	$\theta^\star$
5	0	0	$+90$	$\theta^\star$
6	d_6	0	0	$\theta^\star$

$$A_1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Joint 1

$$J_{v1} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} o_6^0 \qquad J_{\omega 1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Joint 2

$$J_{v2} = \begin{bmatrix} 0 & 0 & c_1 \\ 0 & 0 & s_1 \\ -c_1 & -s_1 & 0 \end{bmatrix} R_1^0 o_6^1 \qquad J_{\omega 2} = R_1^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Joint 3

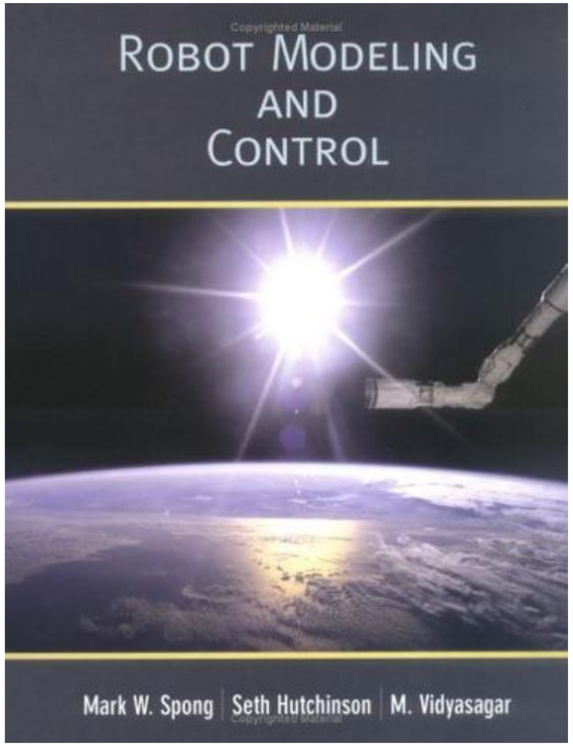
$$J_{v3} = R_3^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad J_{\omega 3} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Joint 4

$$J_{v4} = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix} R_4^0 o_6^4 \qquad J_{\omega 1} = R_4^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

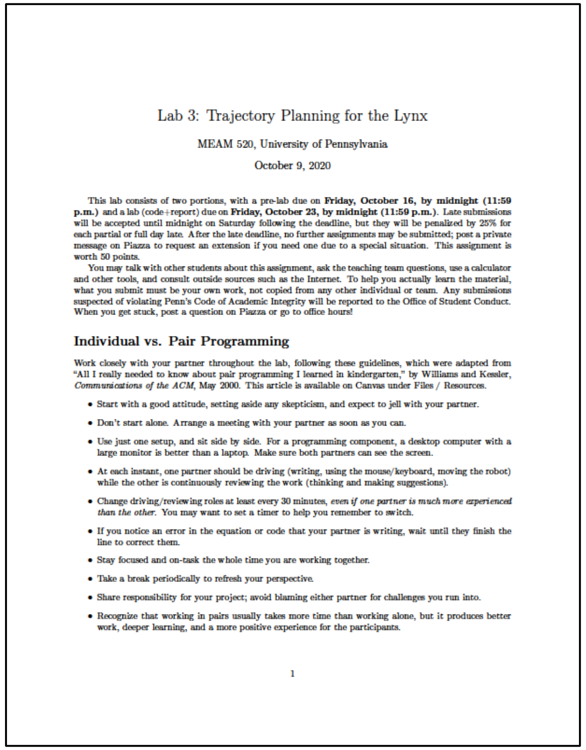
$$\begin{aligned} d_x &= c_1 s_2 d_3 - s_1 d_2 + + d_6(c_1 c_2 c_4 s_5 + c_1 c_5 s_2 - s_1 s_4 s_5) \\ d_y &= s_1 s_2 d_3 + c_1 d_2 + d_6(c_1 s_4 s_5 + c_2 c_4 s_1 s_5 + c_5 s_1 s_2) \\ d_z &= c_2 d_3 + d_6(c_2 c_5 - c_4 s_2 s_5) \end{aligned}$$

Next time: Inverse Velocity Kinematics



Chapter 4: Velocity Kinematics

- Read 4.9, 4.11



Lab 3: Trajectory Planning for the Lynx

MEAM 520, University of Pennsylvania

October 9, 2020

This lab consists of two portions, with a pre-lab due on **Friday, October 16, by midnight (11:59 p.m.)** and a lab (code+report) due on **Friday, October 23, by midnight (11:59 p.m.)**. Late submissions will be accepted until midnight on Saturday following the deadline, but they will be penalized by 25% for each partial or full day late. After the late deadline, no further assignments may be submitted; post a private message on Piazza to request an extension if you need one due to a special situation. This assignment is worth 50 points.

You may talk with other students about this assignment, ask the teaching team questions, use a calculator and other tools, and consult outside sources such as the Internet. To help you actually learn the material, what you submit must be your own work, not copied from any other individual or team. Any submissions suspected of violating Penn's Code of Academic Integrity will be reported to the Office of Student Conduct. When you get stuck, post a question on Piazza or go to office hours!

Individual vs. Pair Programming

Work closely with your partner throughout the lab, following these guidelines, which were adapted from "All I really needed to know about pair programming I learned in kindergarten," by Williams and Kessler, *Communications of the ACM*, May 2000. This article is available on Canvas under Files / Resources.

- Start with a good attitude, setting aside any skepticism, and expect to jell with your partner.
- Don't start alone. Arrange a meeting with your partner as soon as you can.
- Use just one setup, and sit side by side. For a programming component, a desktop computer with a large monitor is better than a laptop. Make sure both partners can see the screen.
- At each instant, one partner should be driving (writing, using the mouse/keyboard, moving the robot) while the other is continuously reviewing the work (thinking and making suggestions).
- Change driving/reviewing roles at least every 30 minutes, *even if one partner is much more experienced than the other*. You may want to set a timer to help you remember to switch.
- If you notice an error in the equation or code that your partner is writing, wait until they finish the line to correct them.
- Stay focused and on-task the whole time you are working together.
- Take a break periodically to refresh your perspective.
- Share responsibility for your project; avoid blaming either partner for challenges you run into.
- Recognize that working in pairs usually takes more time than working alone, but it produces better work, deeper learning, and a more positive experience for the participants.

1

Lab 3: Trajectory Planning due Friday