

autorob.org

EECS 398

Intro. to Autonomous Robotics

ME/EECS 567 ROB 510

Robot Modeling and Control

Fall 2018

Inverse Kinematics

UM EECS 398/598 - autorob.github.io

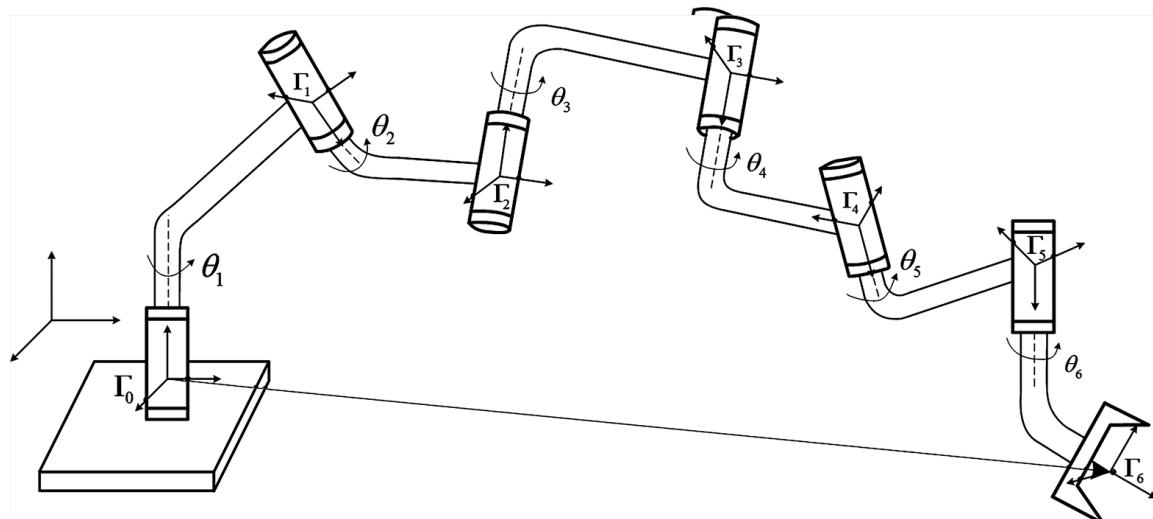
Objective (revisited)

Goal: Given the structure of a robot arm, compute

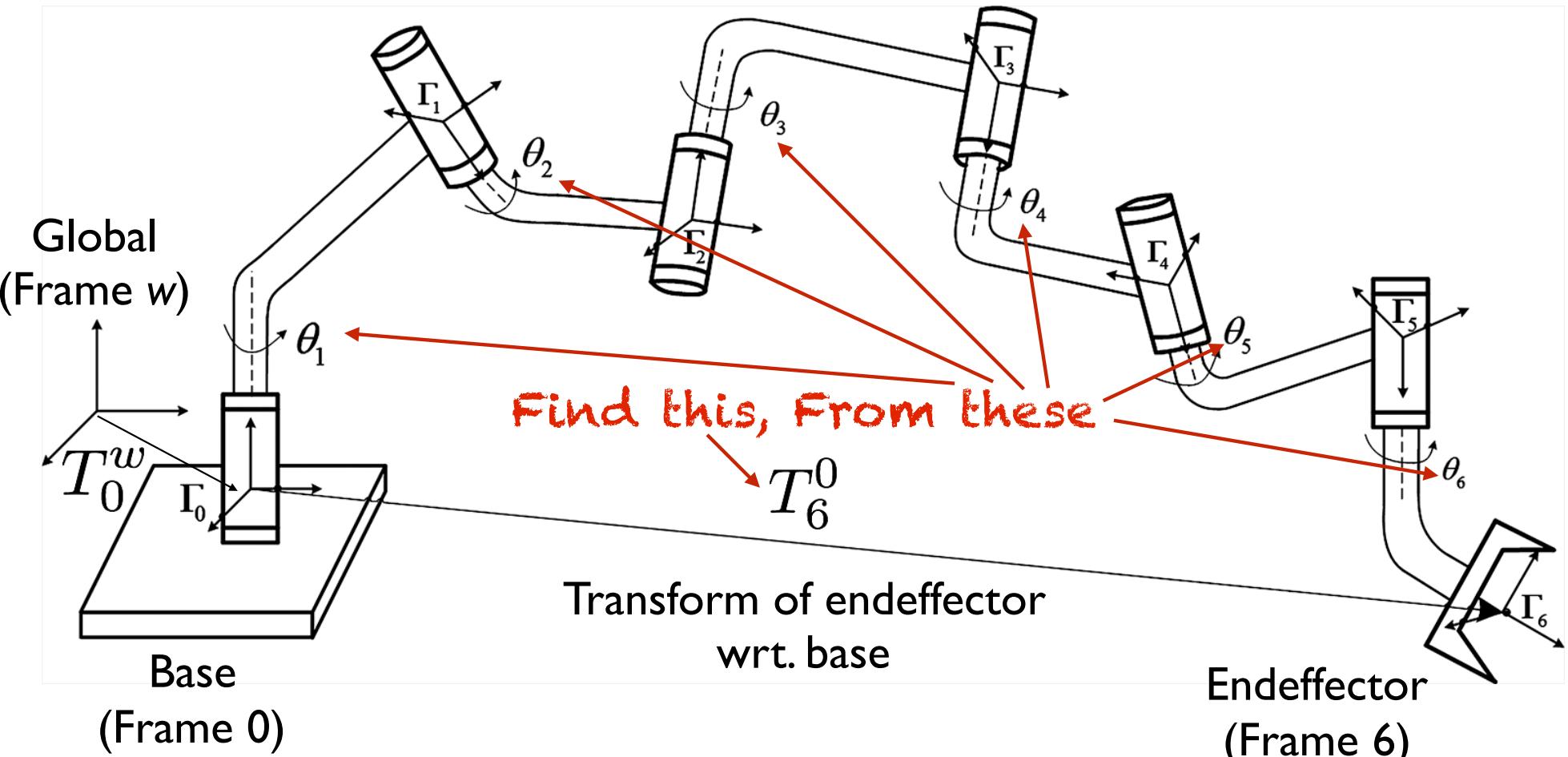
- **Forward kinematics:** predicting the pose of the end-effector, given joint positions.

- **Inverse kinematics:** inferring the joint positions necessary to reach a desired end-effector pose.

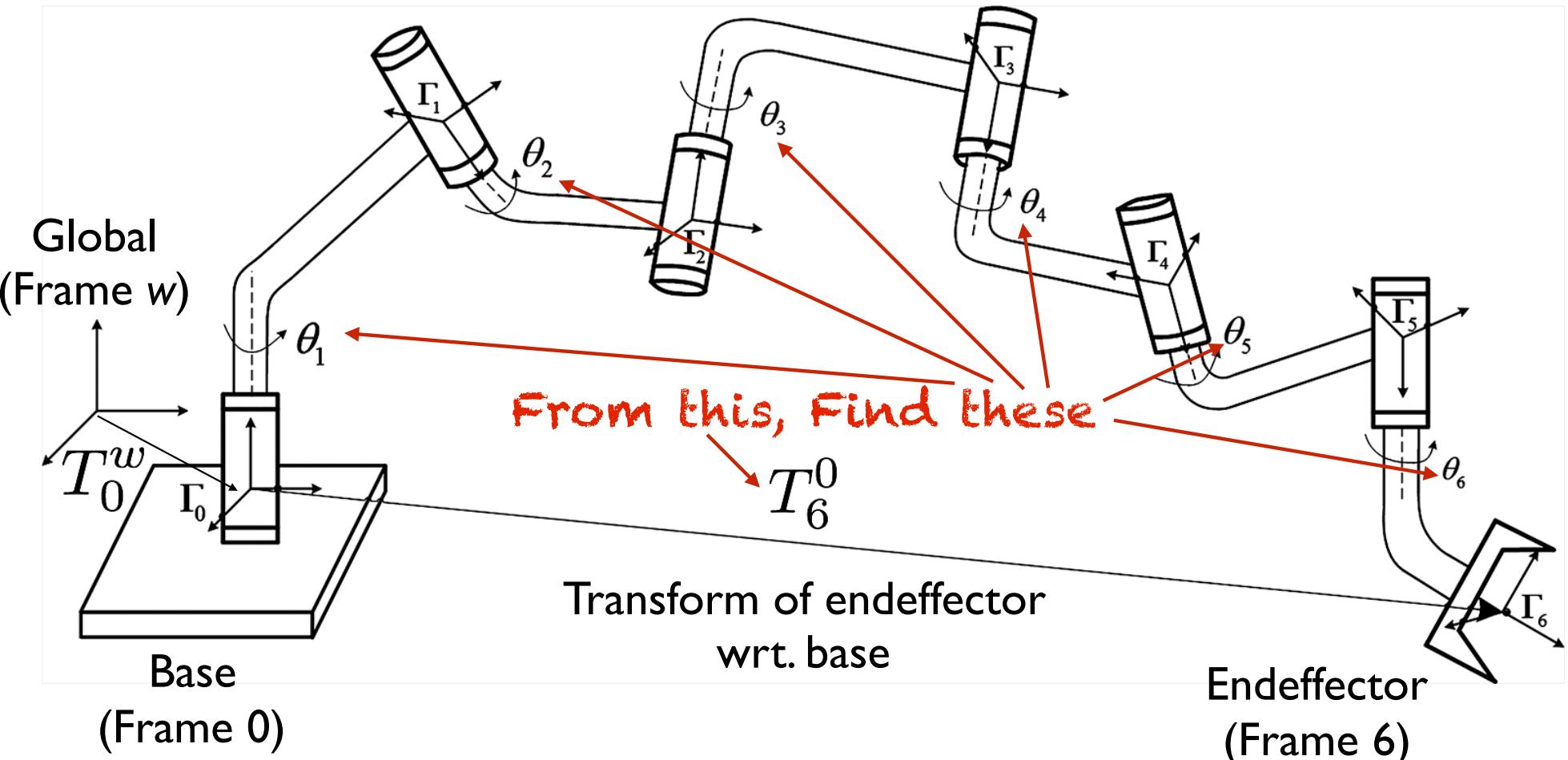
We need to solve for configuration from a transform between world and endeffector frames.



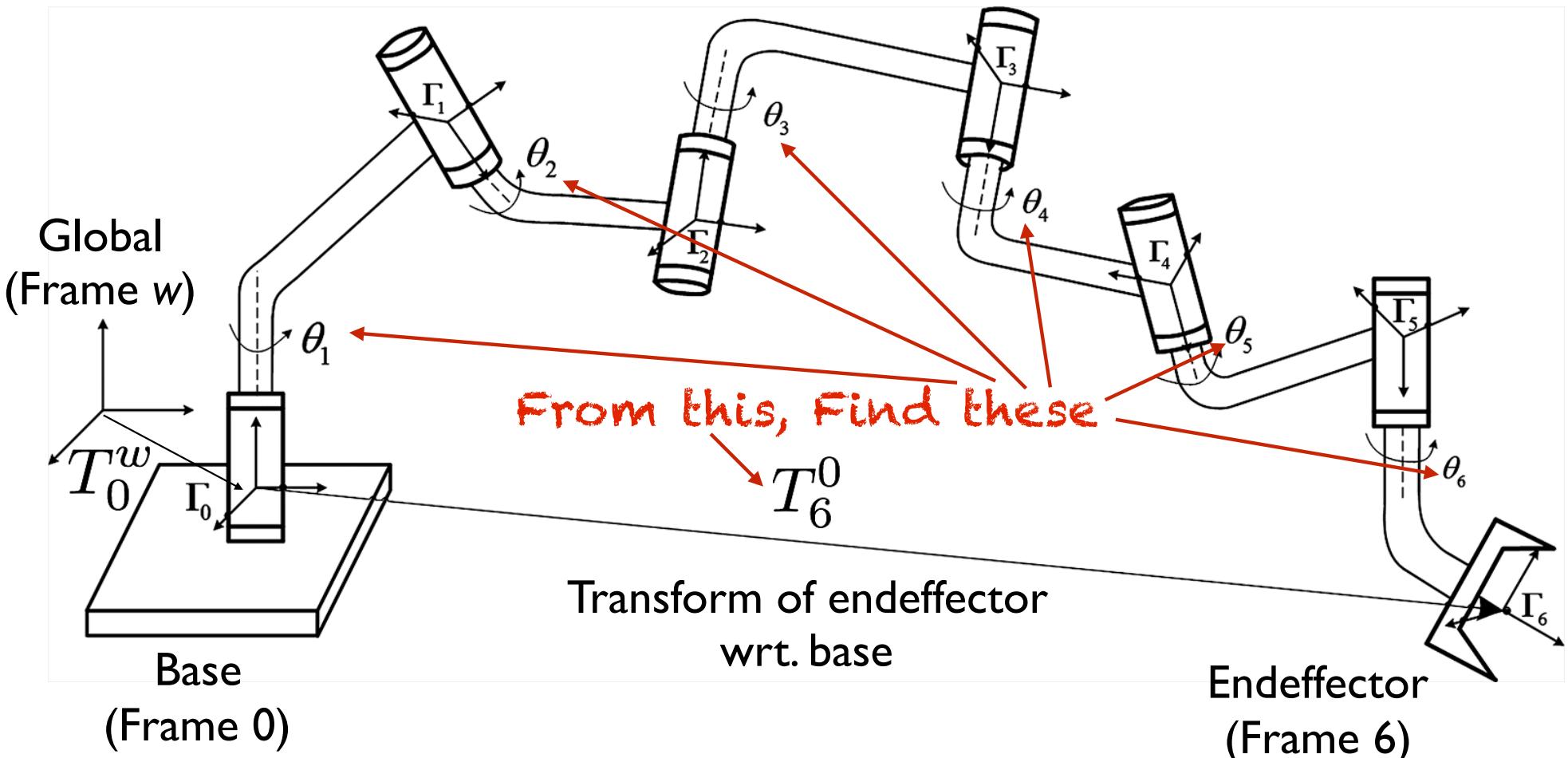
Forward kinematics: many-to-one mapping of robot configuration to reachable workspace endeffector poses



Inverse kinematics: one-to-many mapping of workspace endeffector pose to robot configuration

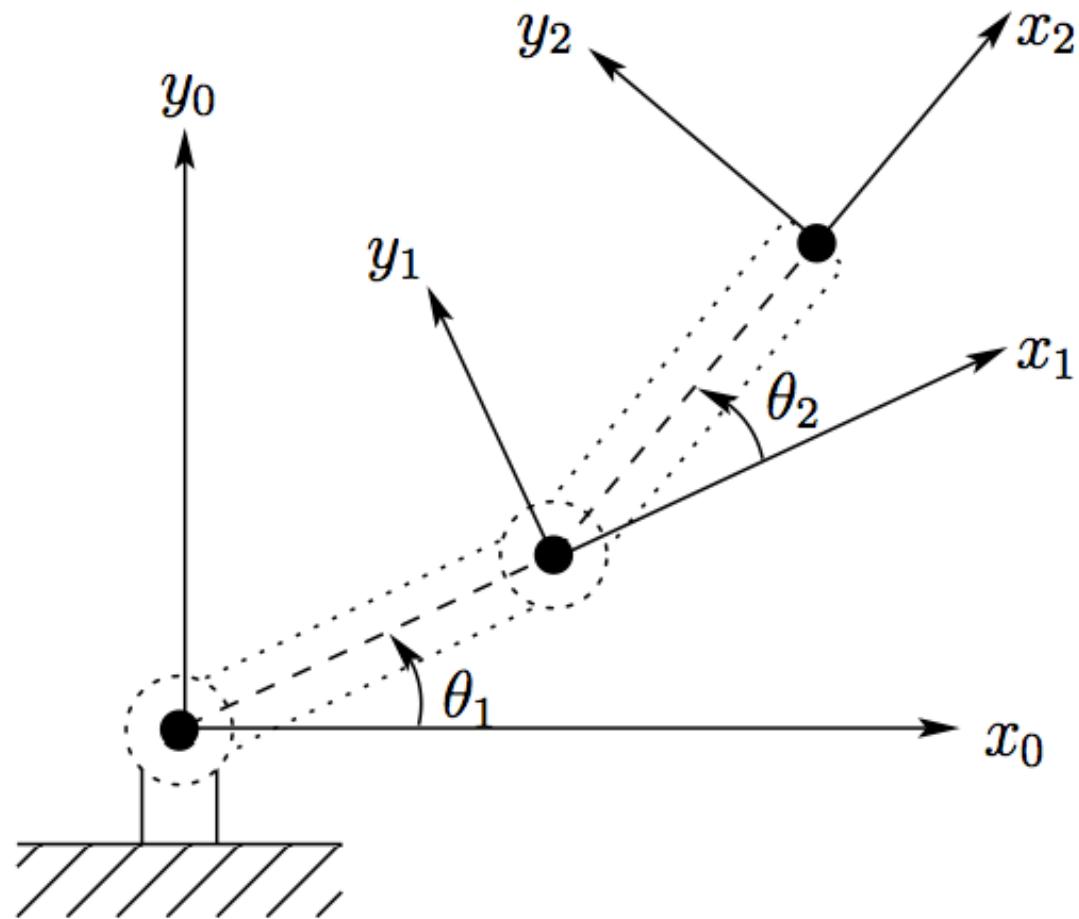


Inverse kinematics: how to solve for $q = \{\theta_1, \dots, \theta_N\}$ from T^0_N ?

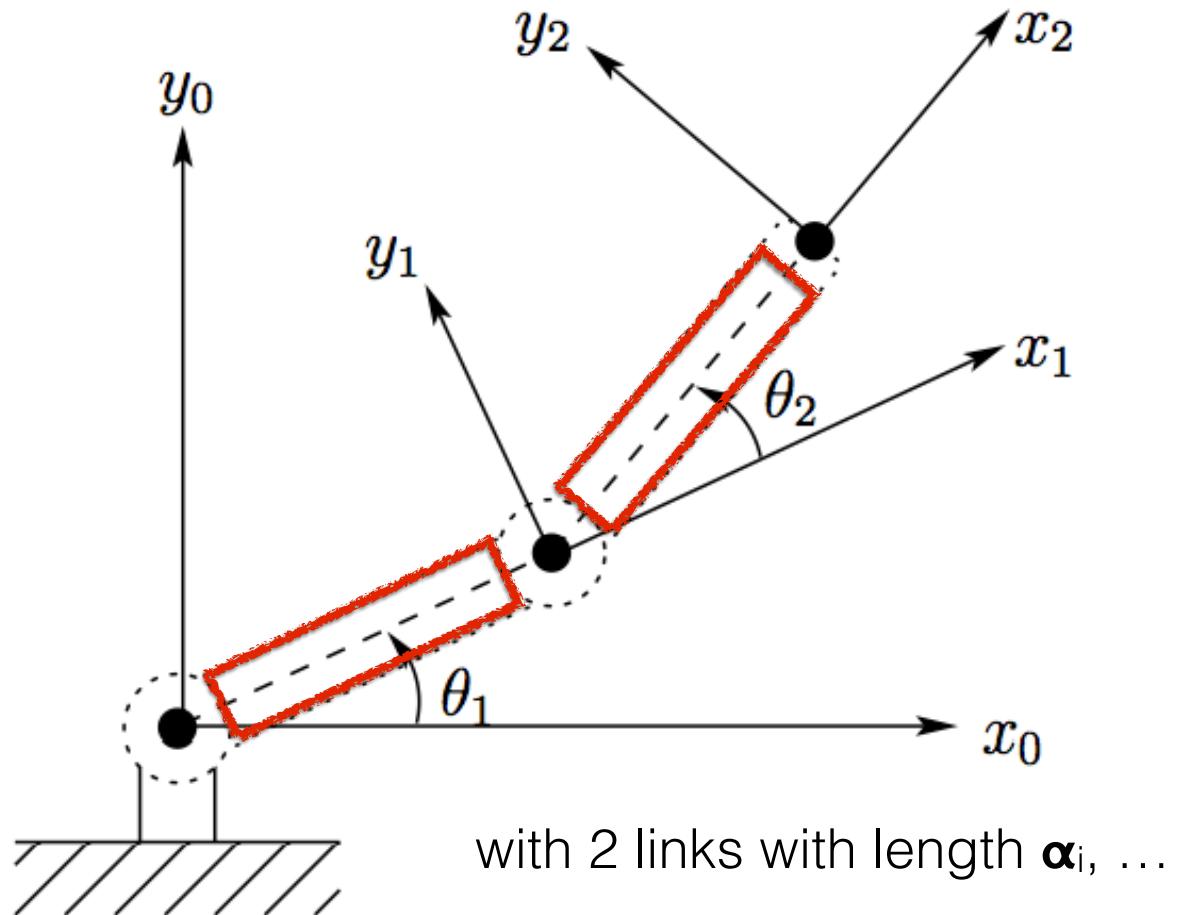


Let's define IK
starting from FK

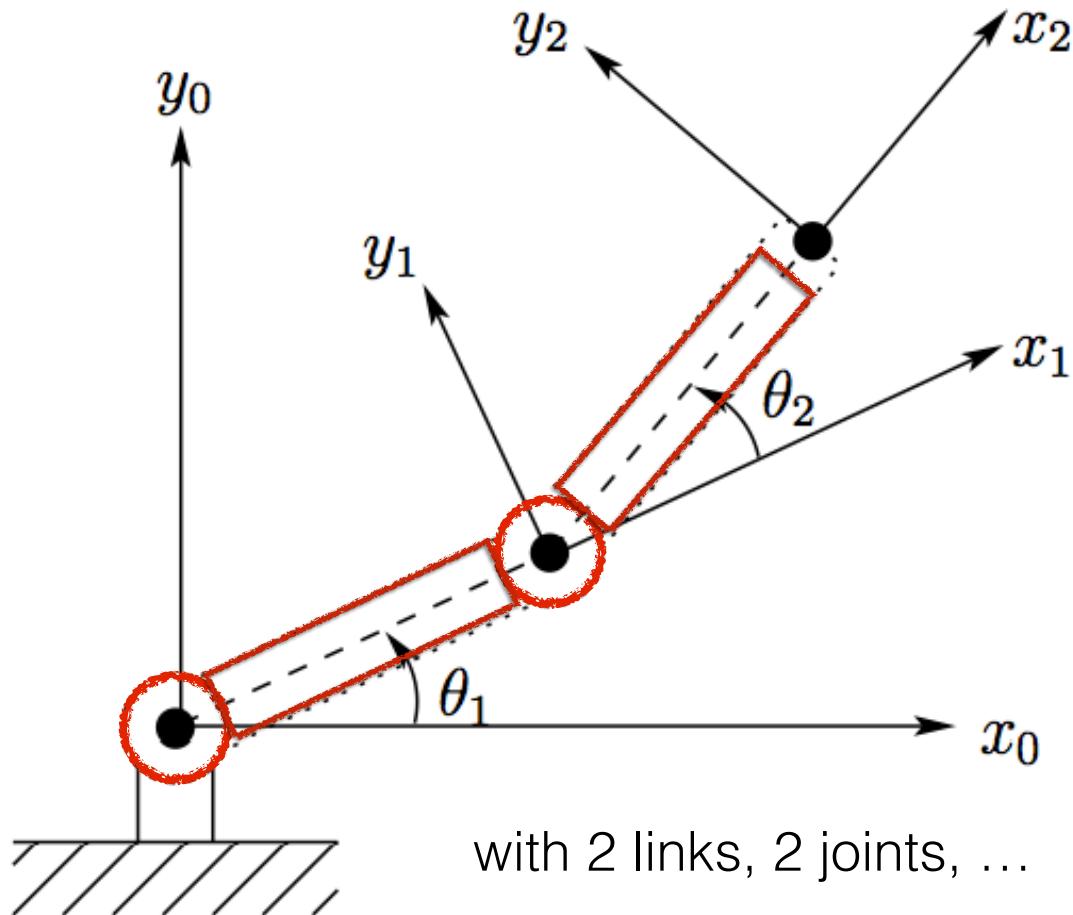
Consider a planar 2-link arm as an example



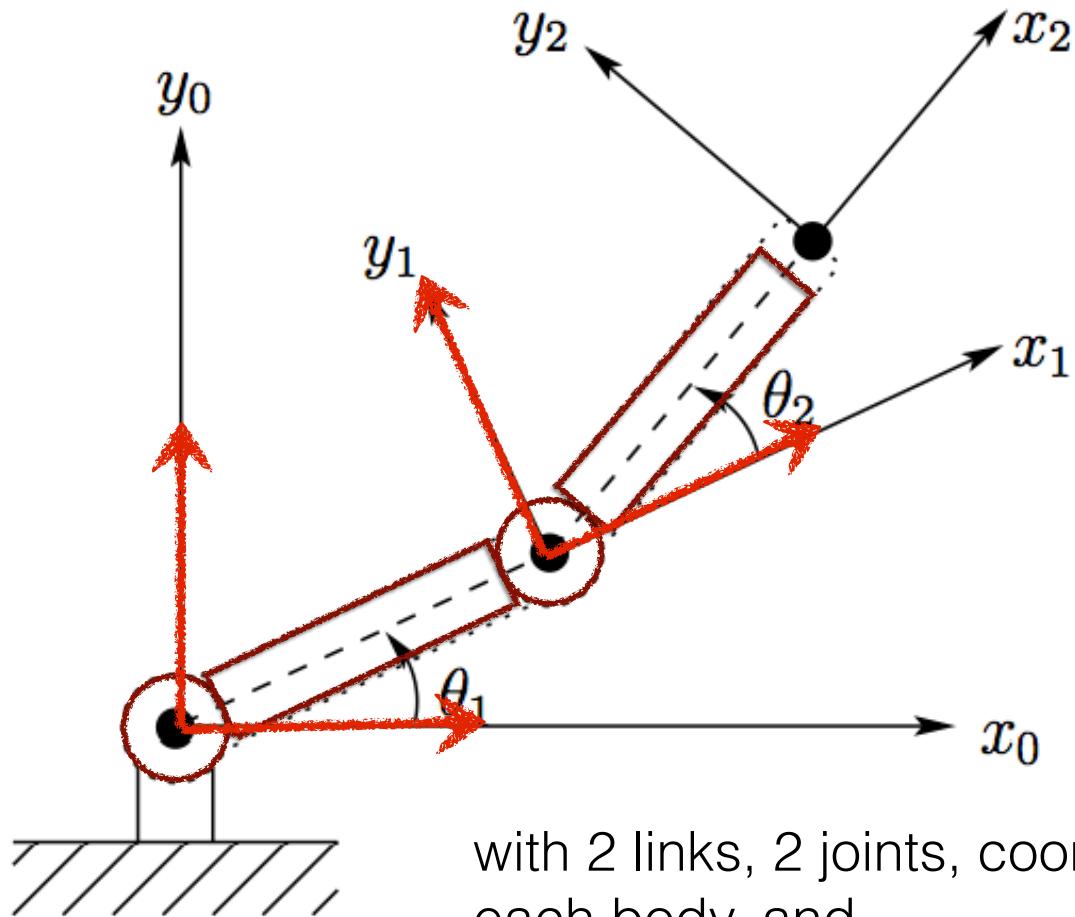
Consider a planar 2-link arm as an example



Consider a planar 2-link arm as an example

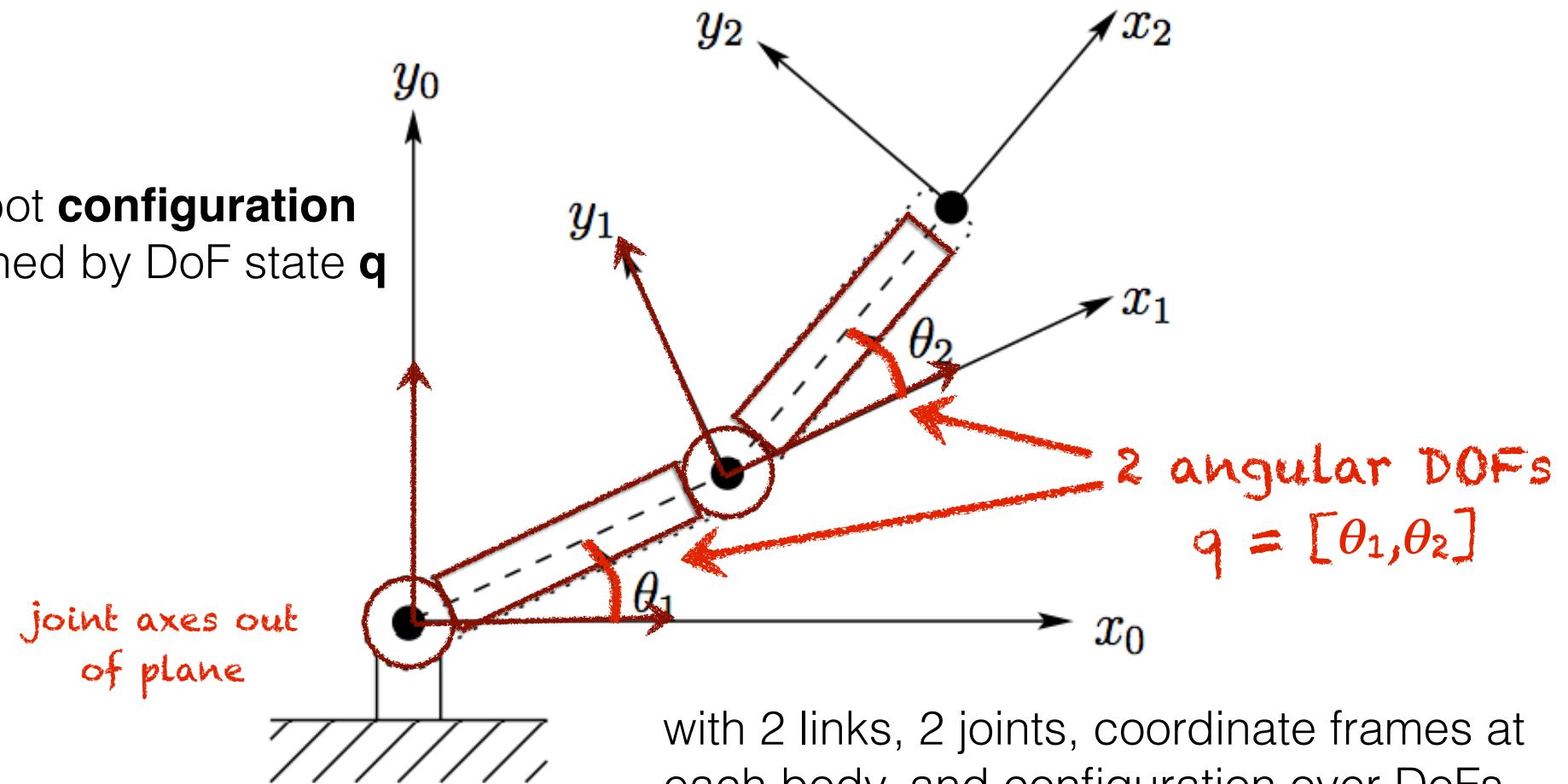


Consider a planar 2-link arm as an example



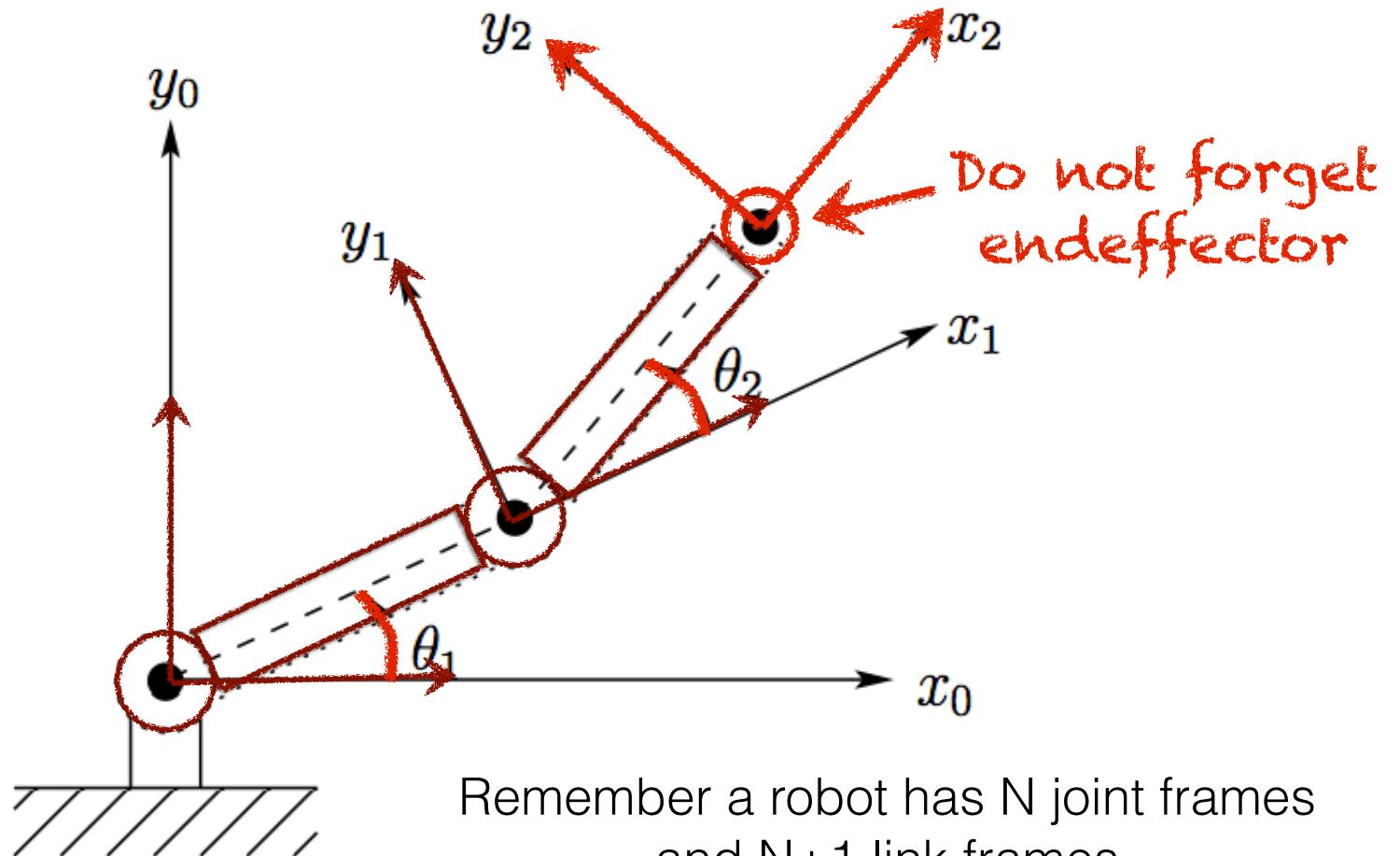
with 2 links, 2 joints, coordinate frames at each body, and ...

Consider a planar 2-link arm as an example



with 2 links, 2 joints, coordinate frames at each body, and configuration over DoFs

Consider a planar 2-link arm as an example

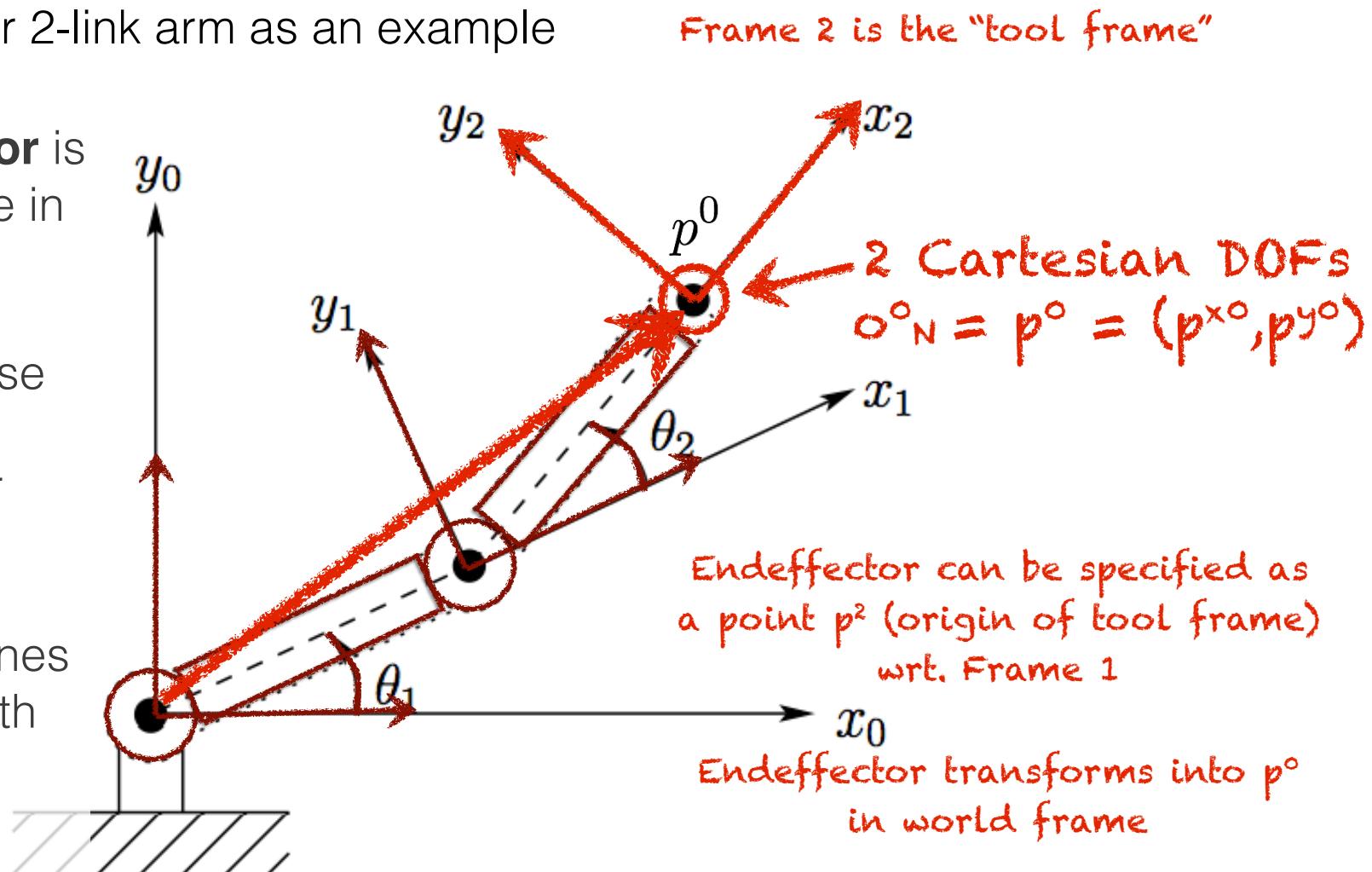


Consider a planar 2-link arm as an example

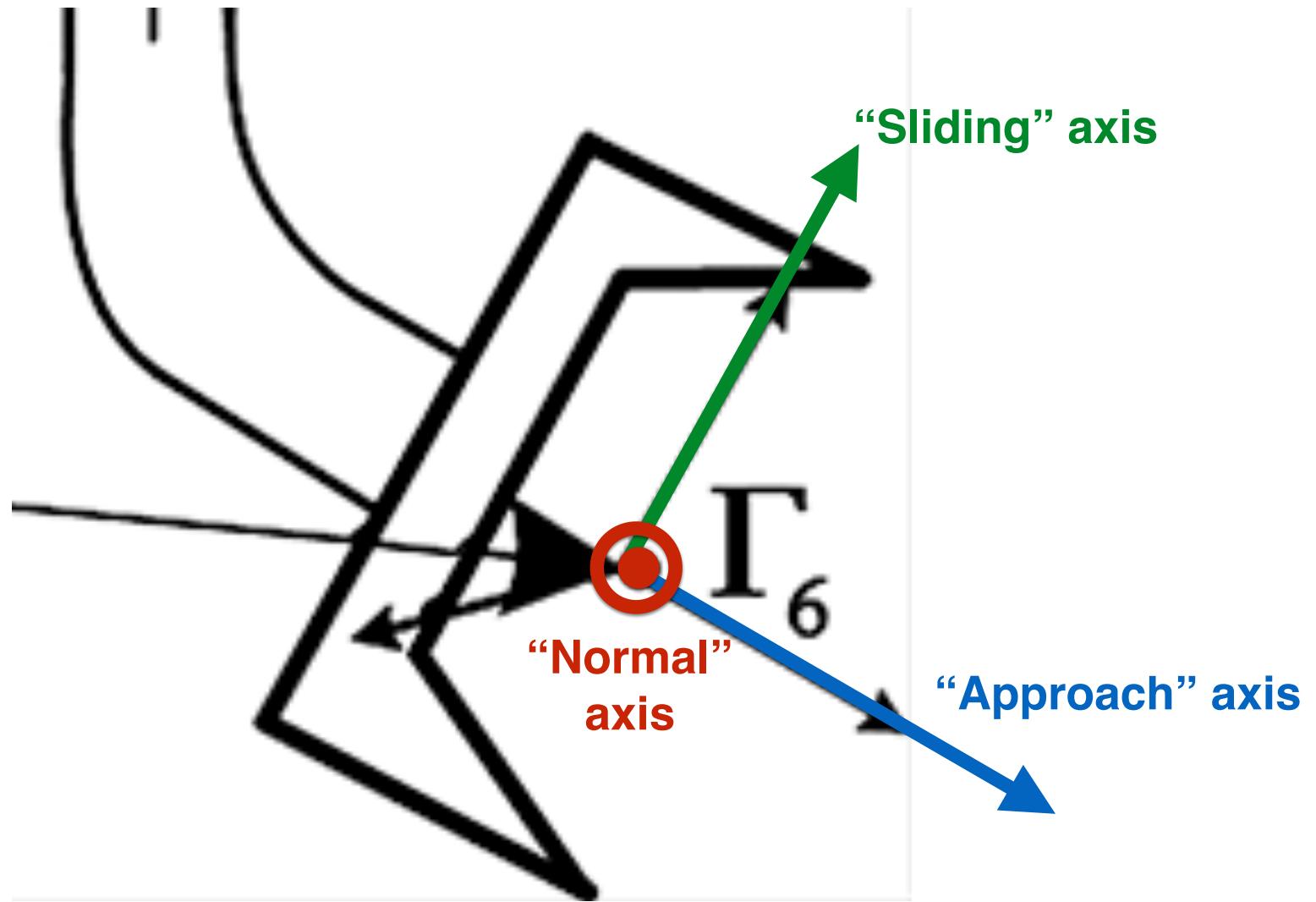
Robot **endeffector** is
the gripper pose in
world frame

Endeffector pose
has position
can consider
orientation

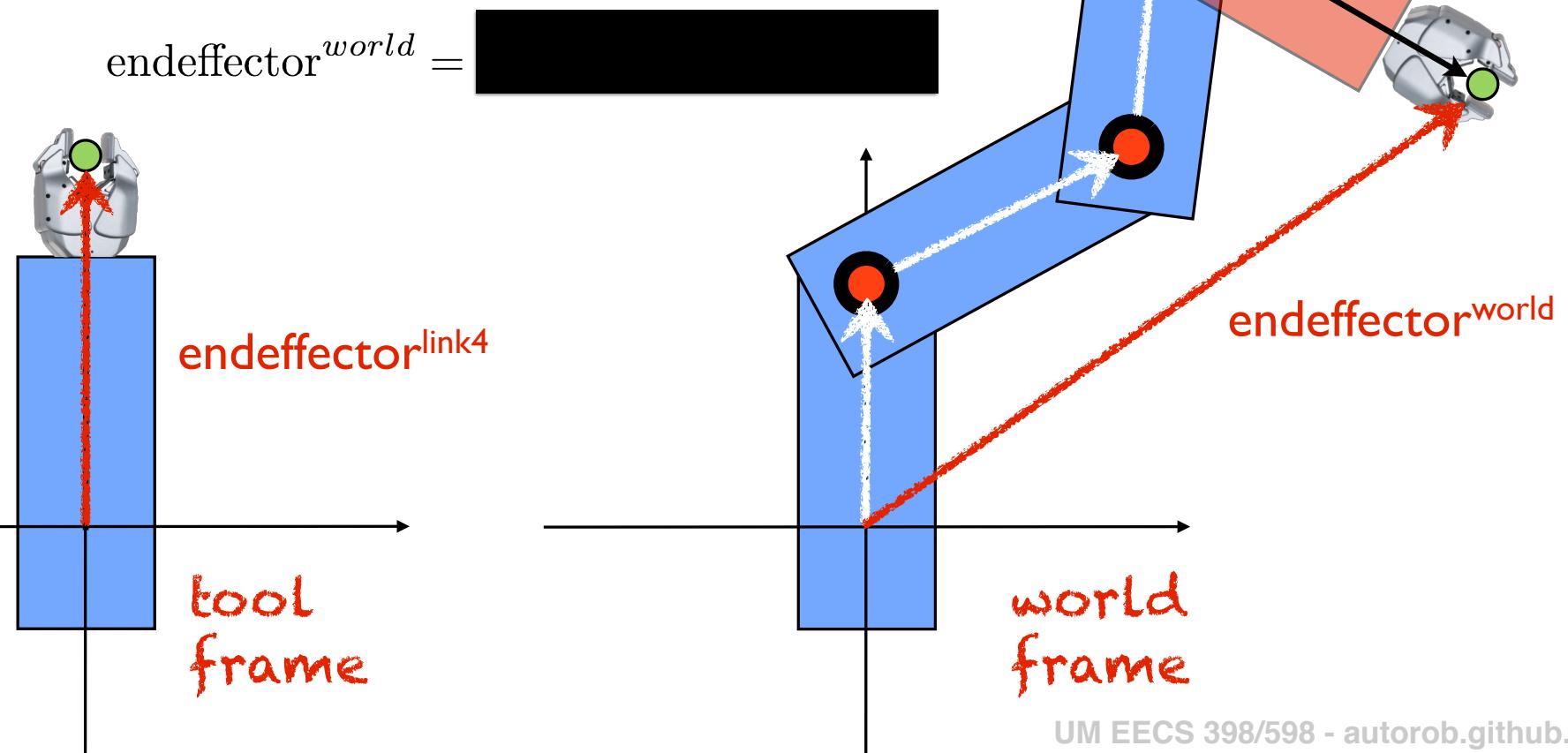
Endeffector defines
“tool frame” with
transform
world frame



Endeffector axes

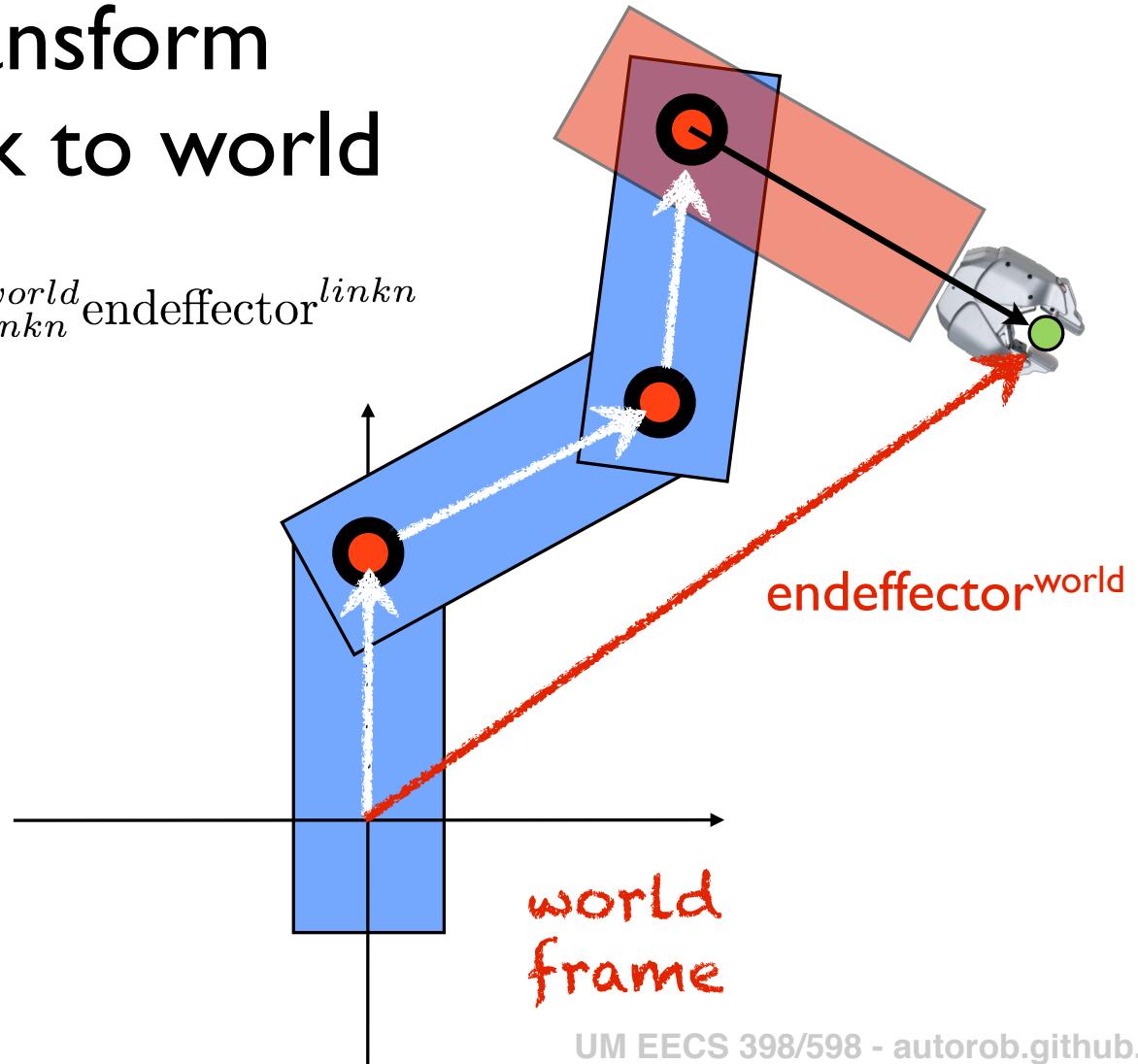
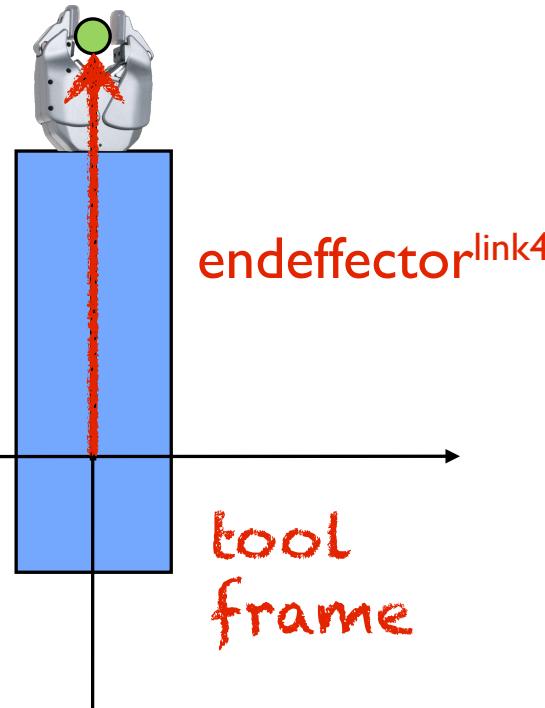


Checkpoint: Transform endeffector on link to world



Checkpoint: Transform endeffector on link to world

$$\text{endeffector}^{world} = T_{linkn}^{world} \text{endeffector}^{linkn}$$

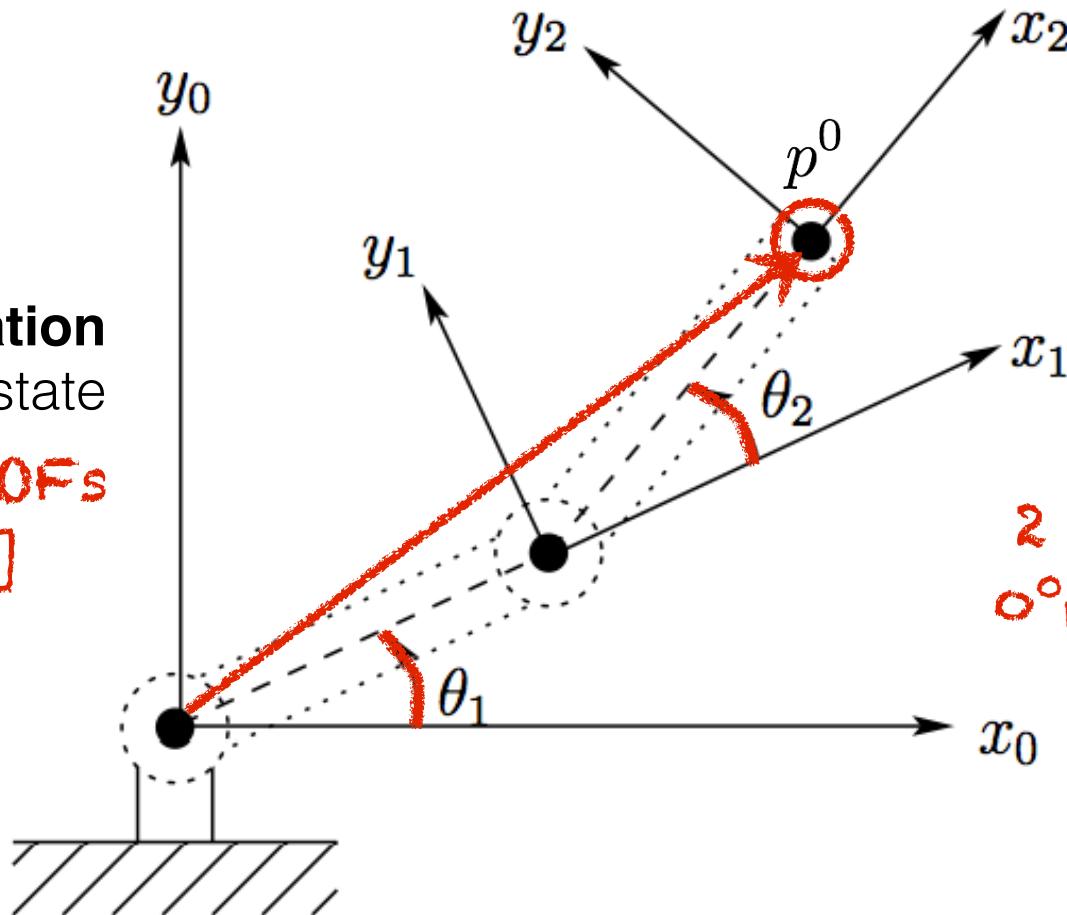


Forward kinematics: “given configuration, compute endeffector”

Robot **configuration**
defined by DoF state

2 angular DOFs

$$q = [\theta_1, \theta_2]$$



Robot **endeffector**
is the gripper pose
in world frame

2 Cartesian DOFs
 ${}^0_N = p^0 = (p^{x^0}, p^{y^0})$

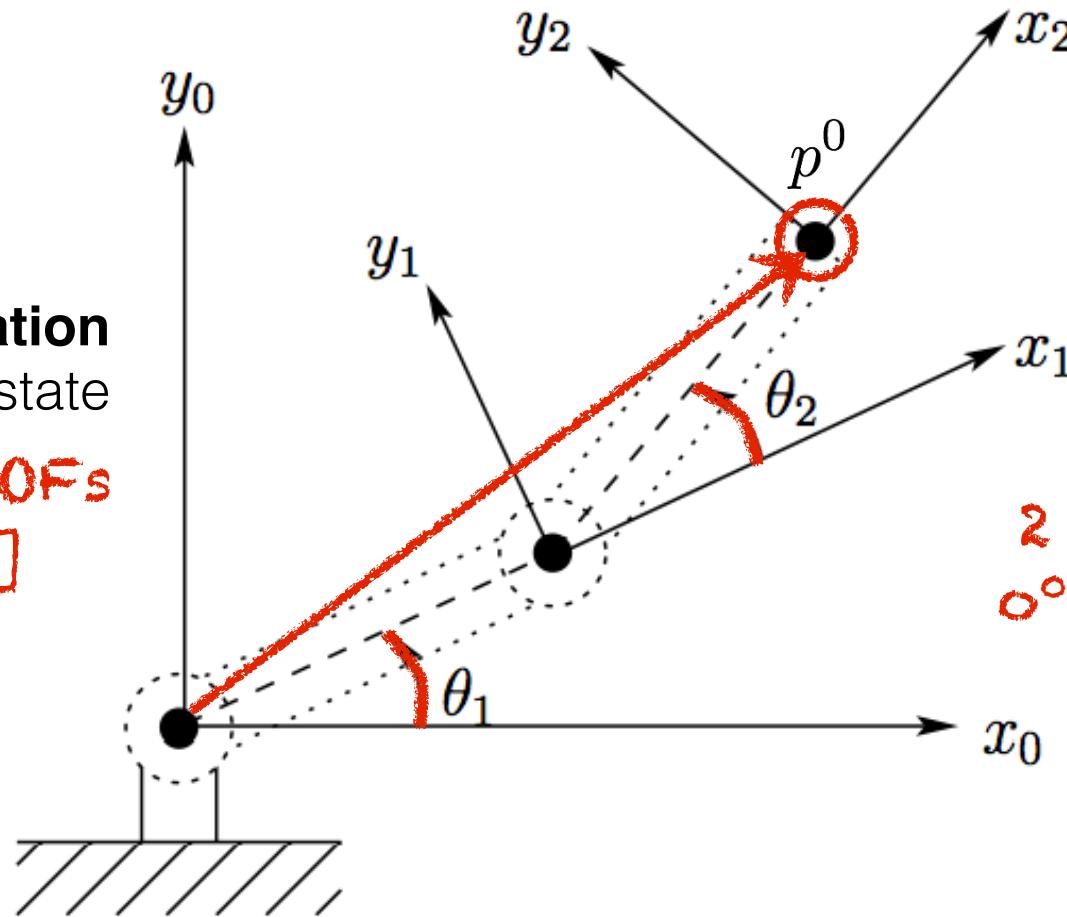
Forward kinematics: $[o^0_N, R^0_N] = f(q)$

$$p^o = f(\theta_1, \theta_2)$$

Robot **configuration**
defined by DoF state

2 angular DOFs

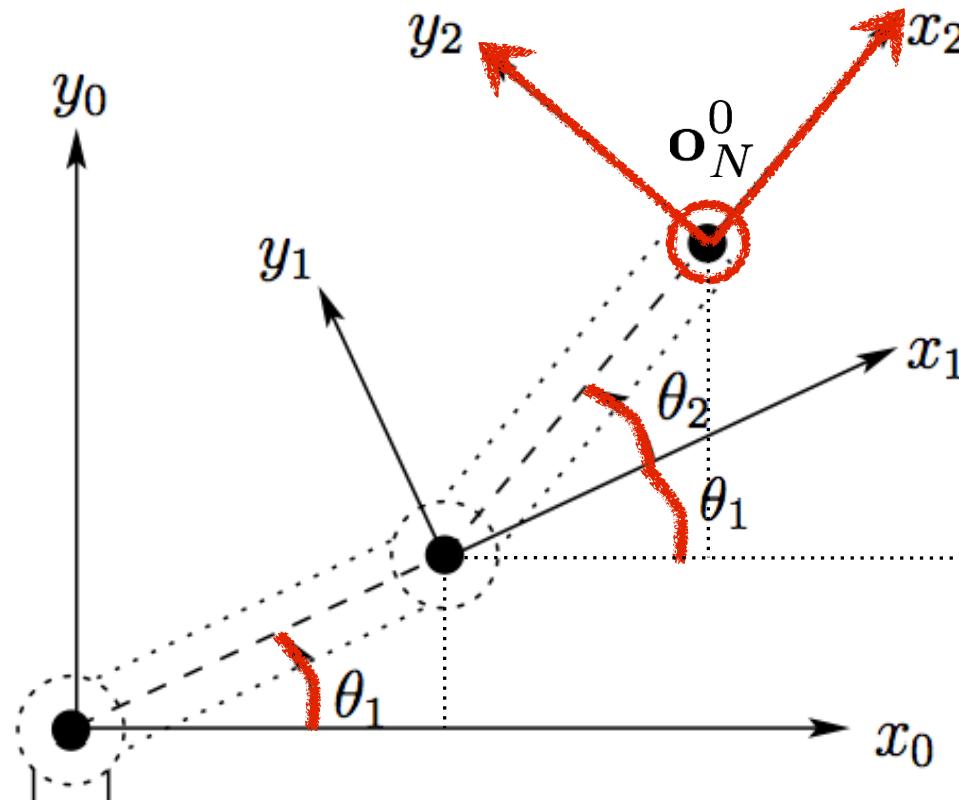
$$q = [\theta_1, \theta_2]$$



Robot **endeffector**
is the gripper pose
in world frame

2 Cartesian DOFs
 $o^0_N = p^o = (p^{x^o}, p^{y^o})$

Forward kinematics: $[\mathbf{o}_N^0, \mathbf{R}_N^0] = f(\mathbf{q})$



What is the position and orientation of the tool wrt. the world?

remember:
 $p^0 = T_1^0 T_2^1 p^2$

$$\mathbf{R}_N^0 = \left[\begin{array}{c} \text{What are the elements of this matrix?} \end{array} \right]$$

$$\mathbf{o}_N^0 = \left[\begin{array}{c} \text{What are the elements of this vector?} \end{array} \right]$$

Forward kinematics: $[\mathbf{o}_N^0, \mathbf{R}_N^0] = f(\mathbf{q})$

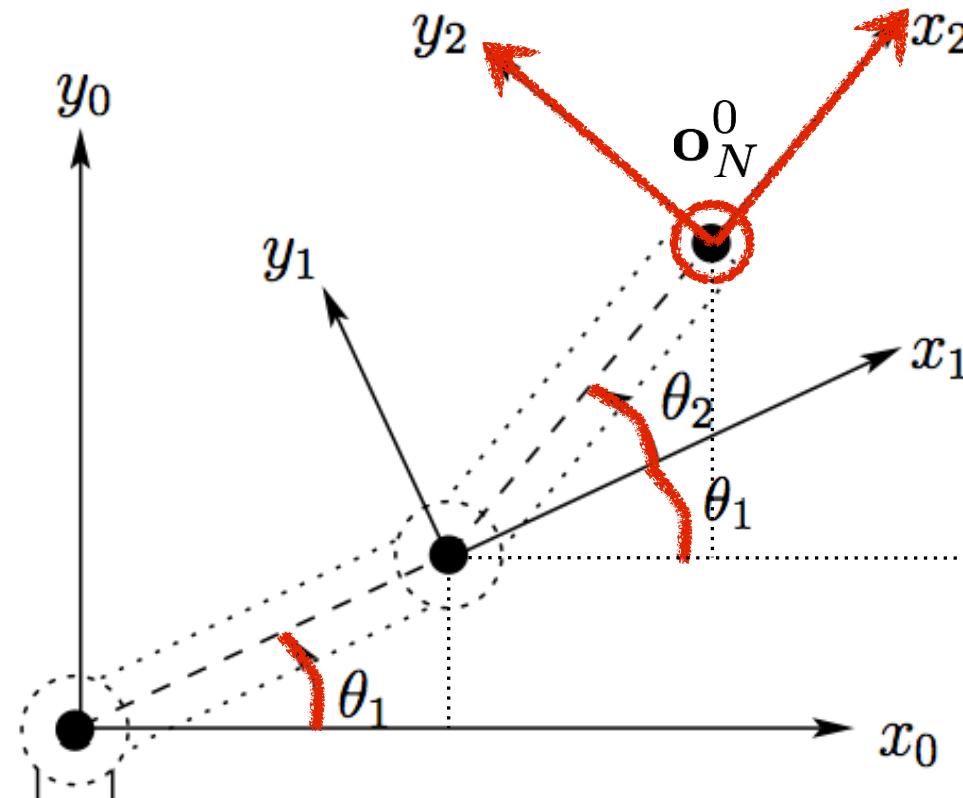
remember:

$$p^0 = T_1^0 T_2^1 p^2$$

$$p^0 = R_2^0 p^2 + d_2^0$$

where

$$R_2^0 = R_1^0 R_2^1$$



What is the position and orientation of the tool wrt. the world?

$$\mathbf{R}_N^0 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix} \quad \mathbf{o}_N^0 = \begin{bmatrix} \text{[Redacted]} \\ \text{[Redacted]} \end{bmatrix}$$

What are the elements of this vector?

Start with:

$$d_2^0 = R_1^0 d_2^1 + d_1^0$$

substitute in variables then perform operations:

$$\begin{bmatrix} \cos\theta_2 & -\sin\theta_2 \\ \sin\theta_2 & \cos\theta_2 \end{bmatrix} \begin{bmatrix} \alpha_2 \cos\theta_2 \\ \alpha_2 \sin\theta_2 \end{bmatrix} + \begin{bmatrix} \alpha_1 \cos\theta_1 \\ \alpha_1 \sin\theta_1 \end{bmatrix}$$

then substitute trig identities

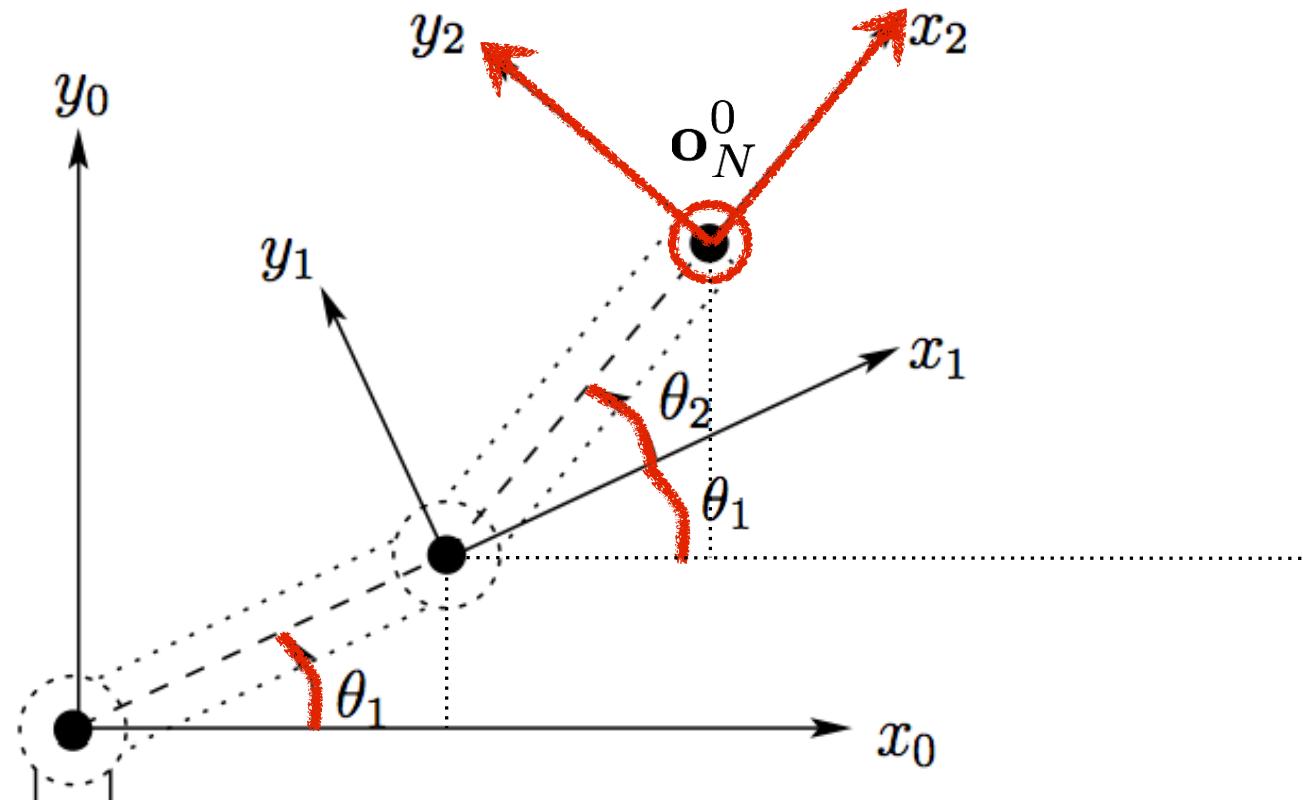
$$\cos(x + y) = \cos(x)\cos(y) - \sin(x)\sin(y)$$

$$\sin(x + y) = \sin(x)\cos(y) - \cos(x)\sin(y)$$

to get:

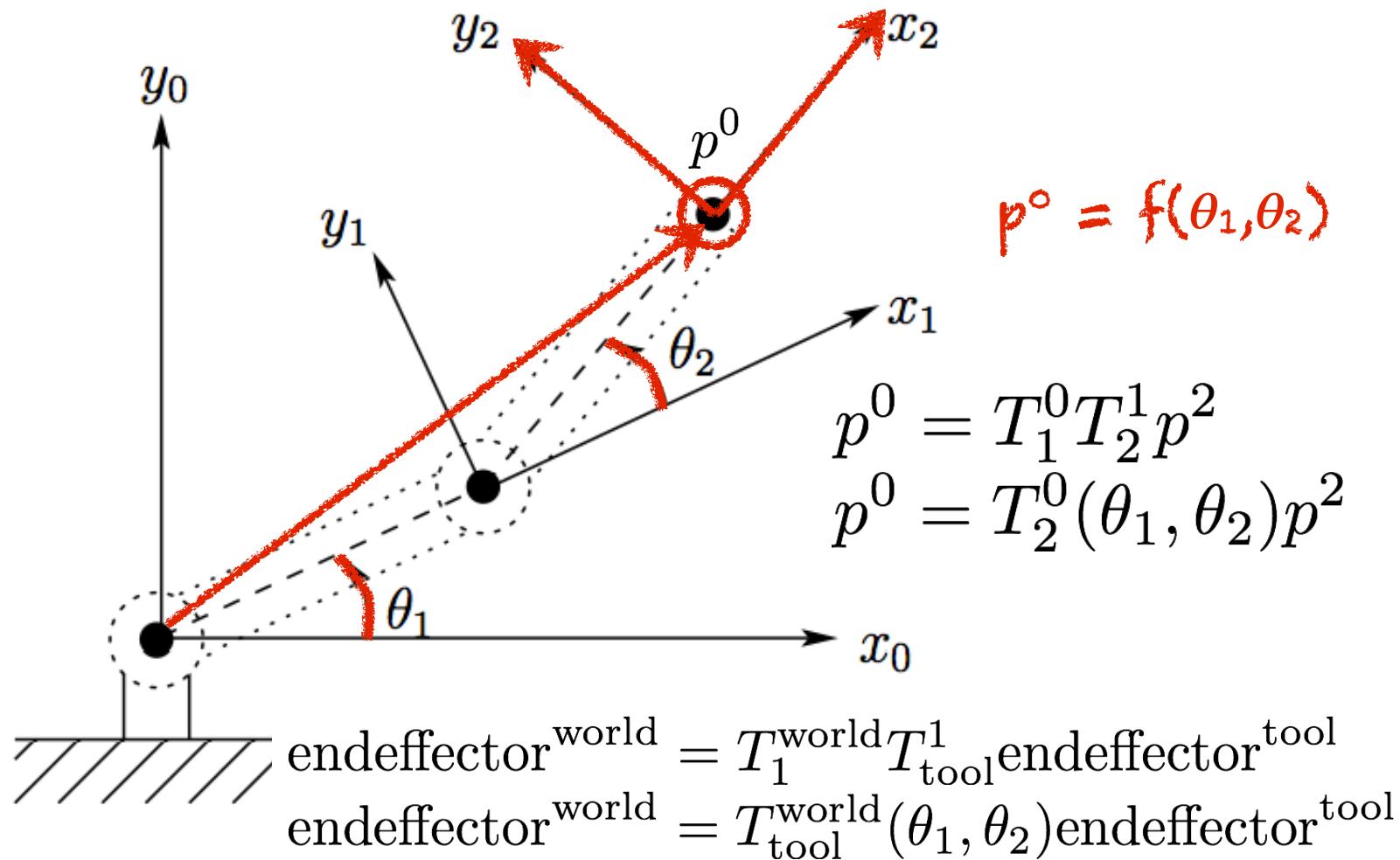
$$\mathbf{o}_N^0 = \left[\begin{array}{c} \text{What are the elements of this vector?} \end{array} \right]$$

Forward kinematics: $[\mathbf{o}_N^0, \mathbf{R}_N^0] = f(\mathbf{q})$

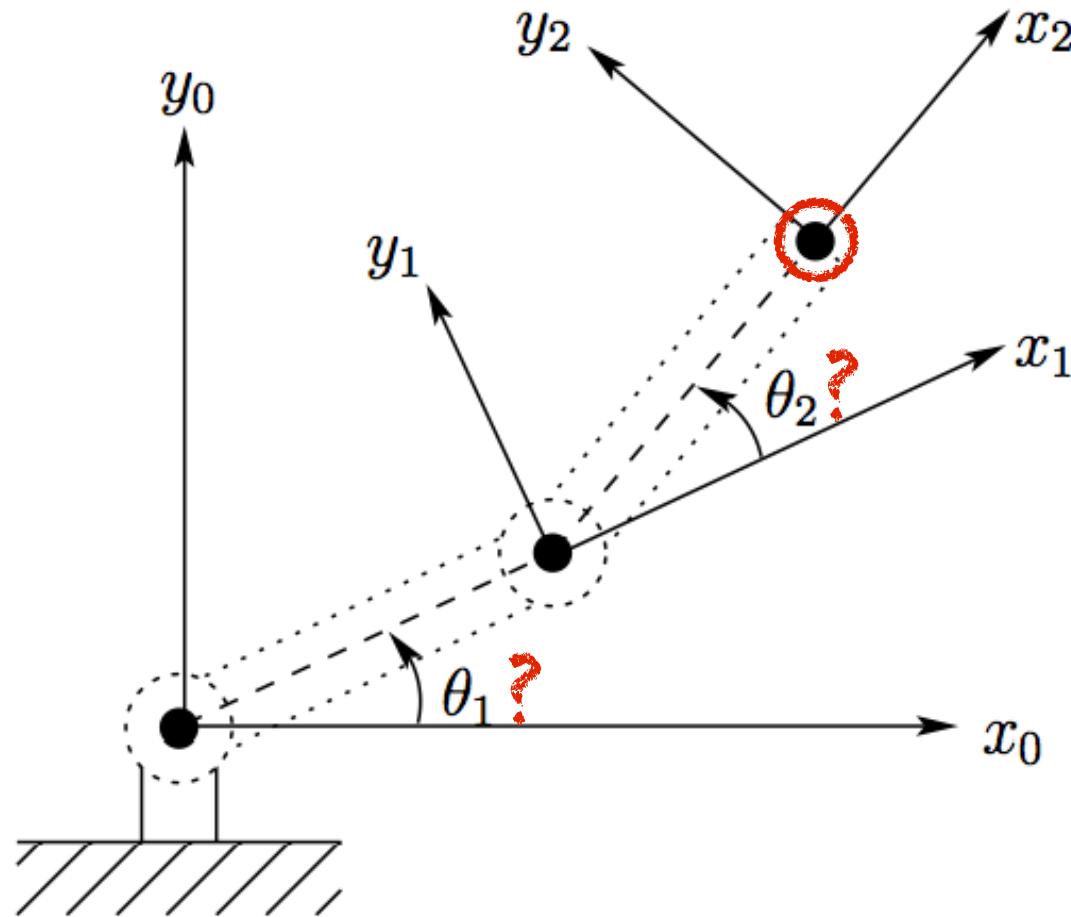


$$\mathbf{R}_N^0 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix} \quad \mathbf{o}_N^0 = \begin{bmatrix} \alpha_1 \cos \theta_1 + \alpha_2 \cos(\theta_1 + \theta_2) \\ \alpha_1 \sin \theta_1 + \alpha_2 \sin(\theta_1 + \theta_2) \end{bmatrix}$$

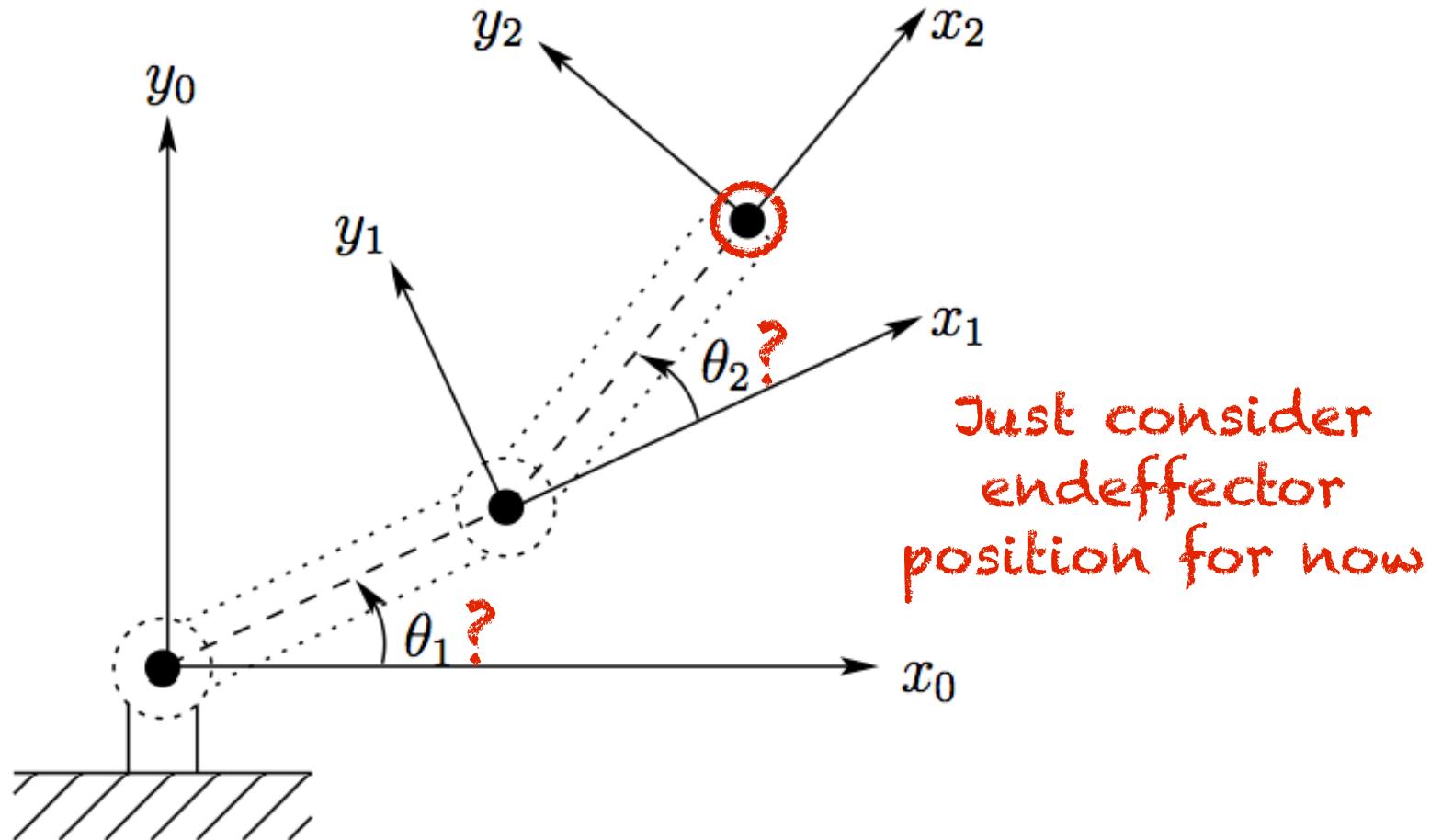
Forward kinematics: $[{\mathbf{o}}^0_N, {\mathbf{R}}^0_N] = f(\mathbf{q})$



Inverse kinematics: “given endeffector, compute configuration”



Inverse kinematics: $\mathbf{q} = f^{-1}([\mathbf{o}^0_N, \mathbf{R}^0_N])$ $[\theta_1, \theta_2] = f^{-1}(p^o)$

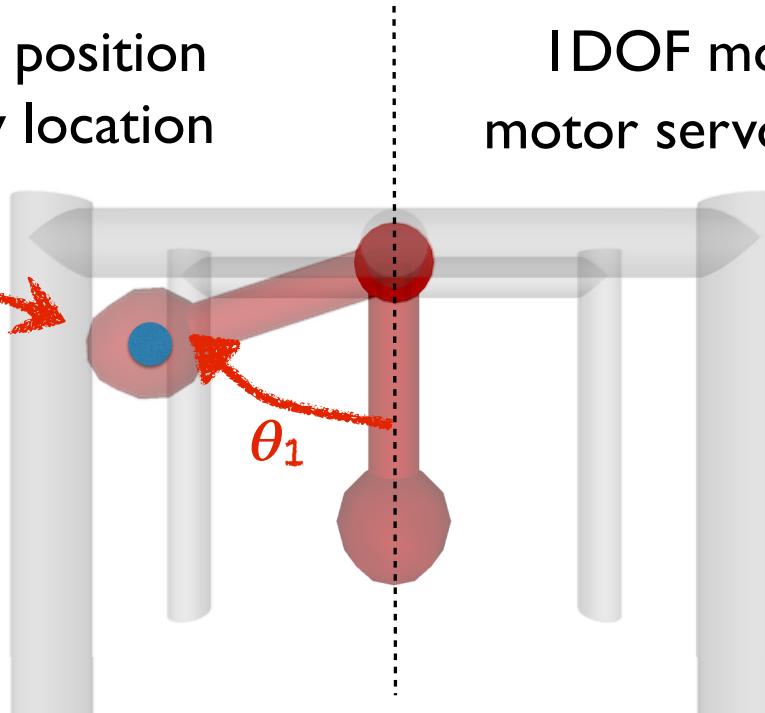


1DOF pendulum example



desired endeffector position
(\mathbf{o}_N^0) given as an x,y location

what is θ_1 ?



assume:
1DOF motor at pendulum axis,
motor servo moves arm to angle θ_1

1DOF pendulum example

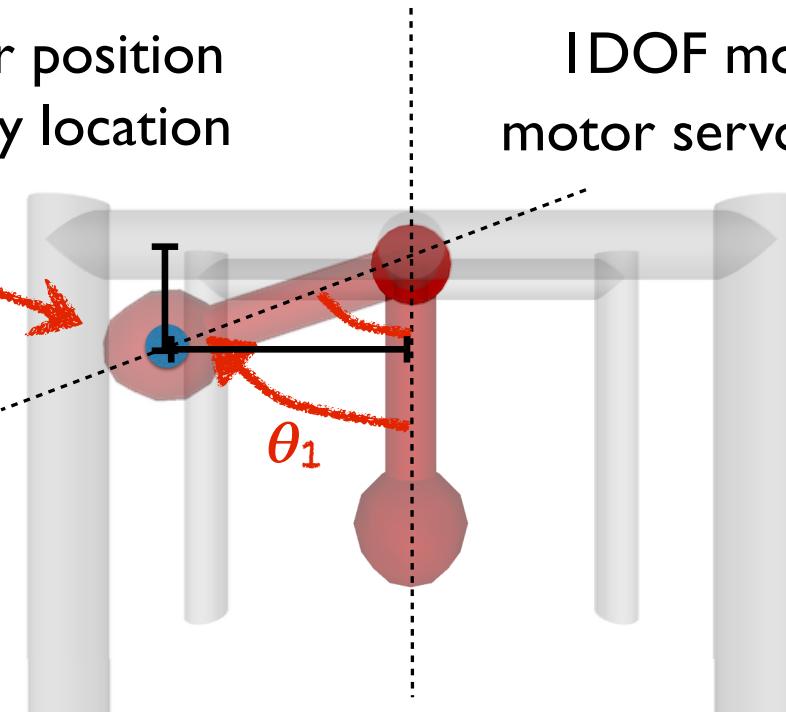


desired endeffector position
(\mathbf{o}_N^0) given as an x,y location

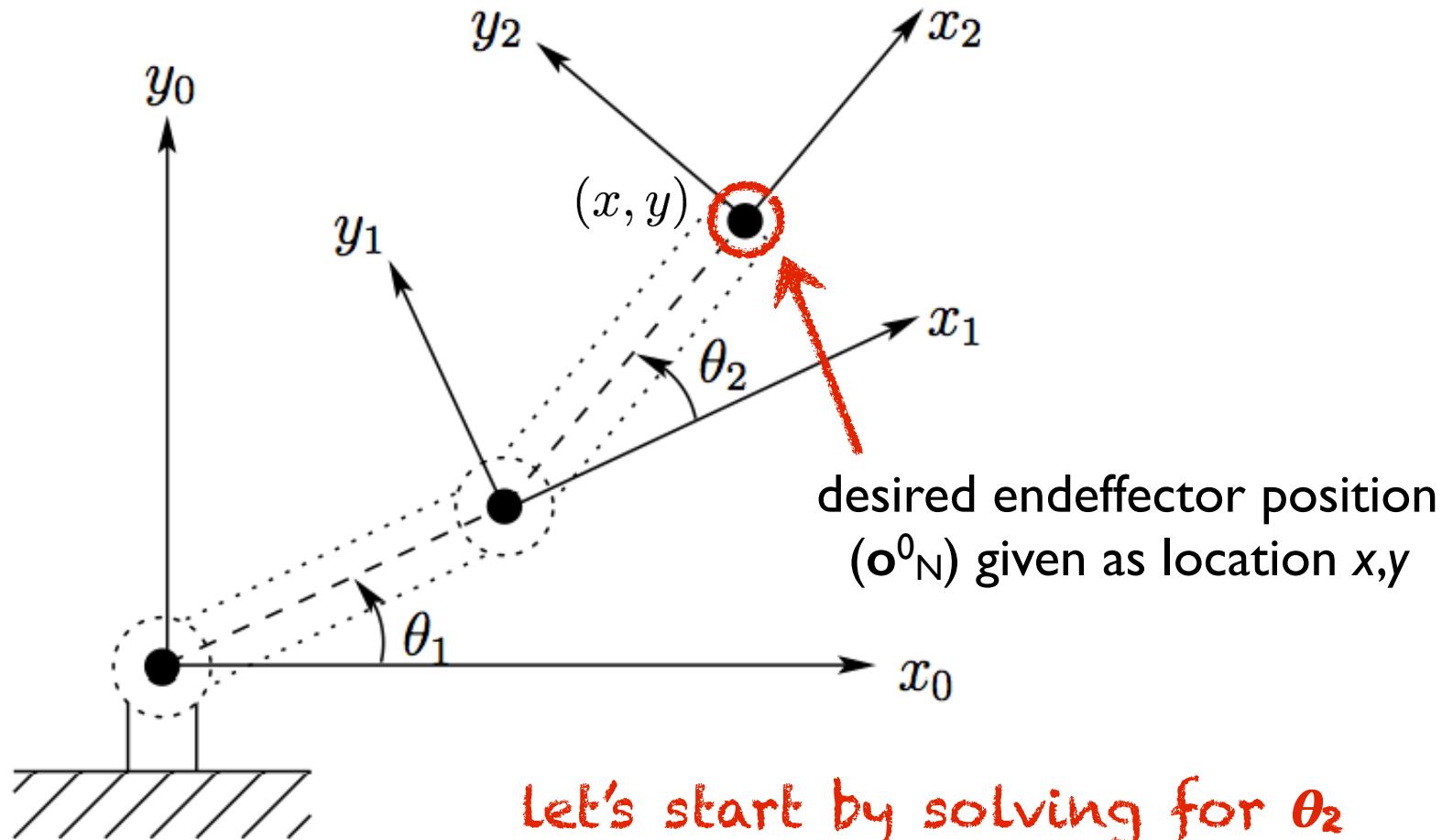
what is θ_1 ?

$$\theta_1 = \tan^{-1}(y/x)$$

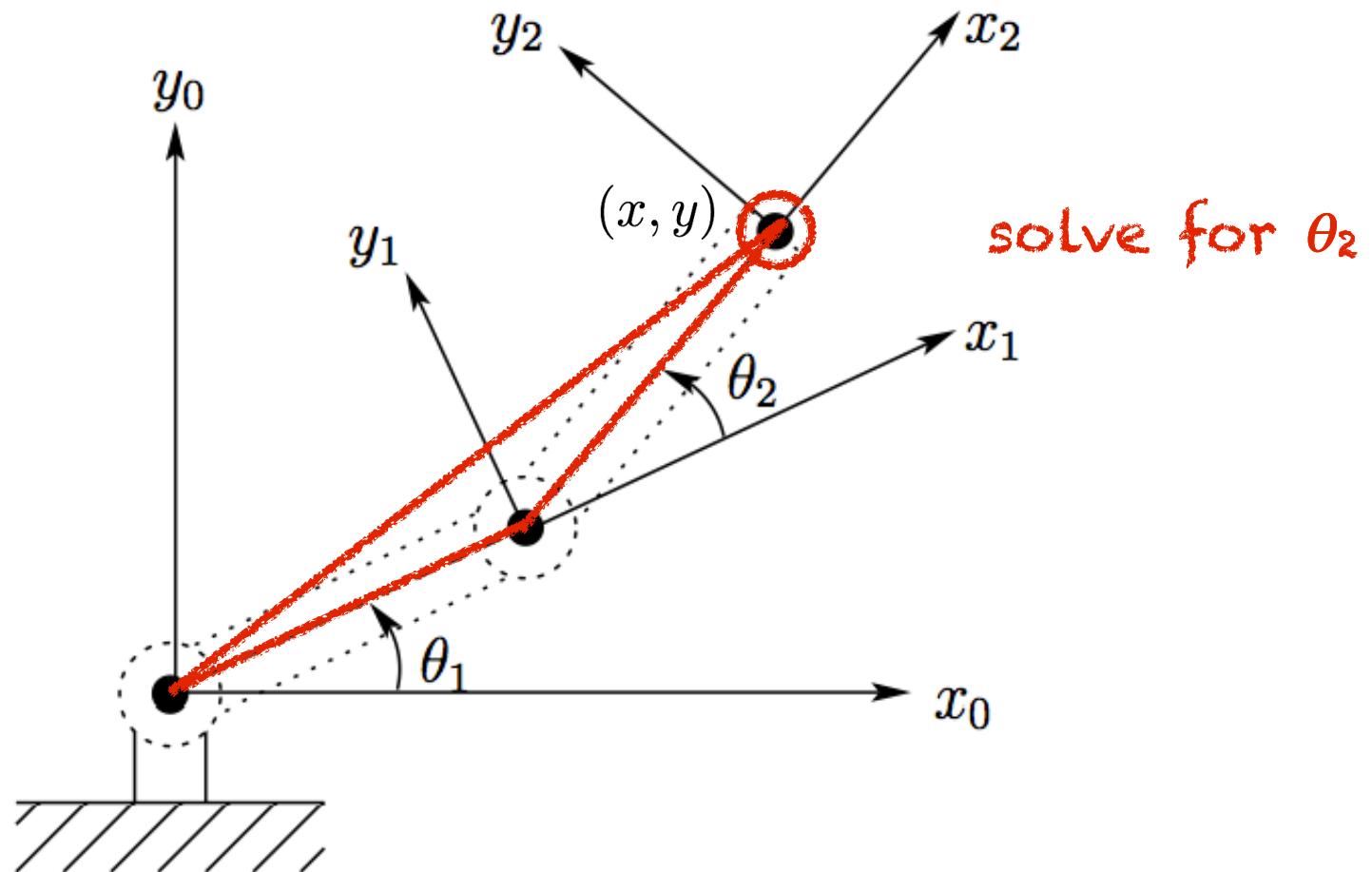
assume:
1DOF motor at pendulum axis,
motor servo moves arm to angle θ_1



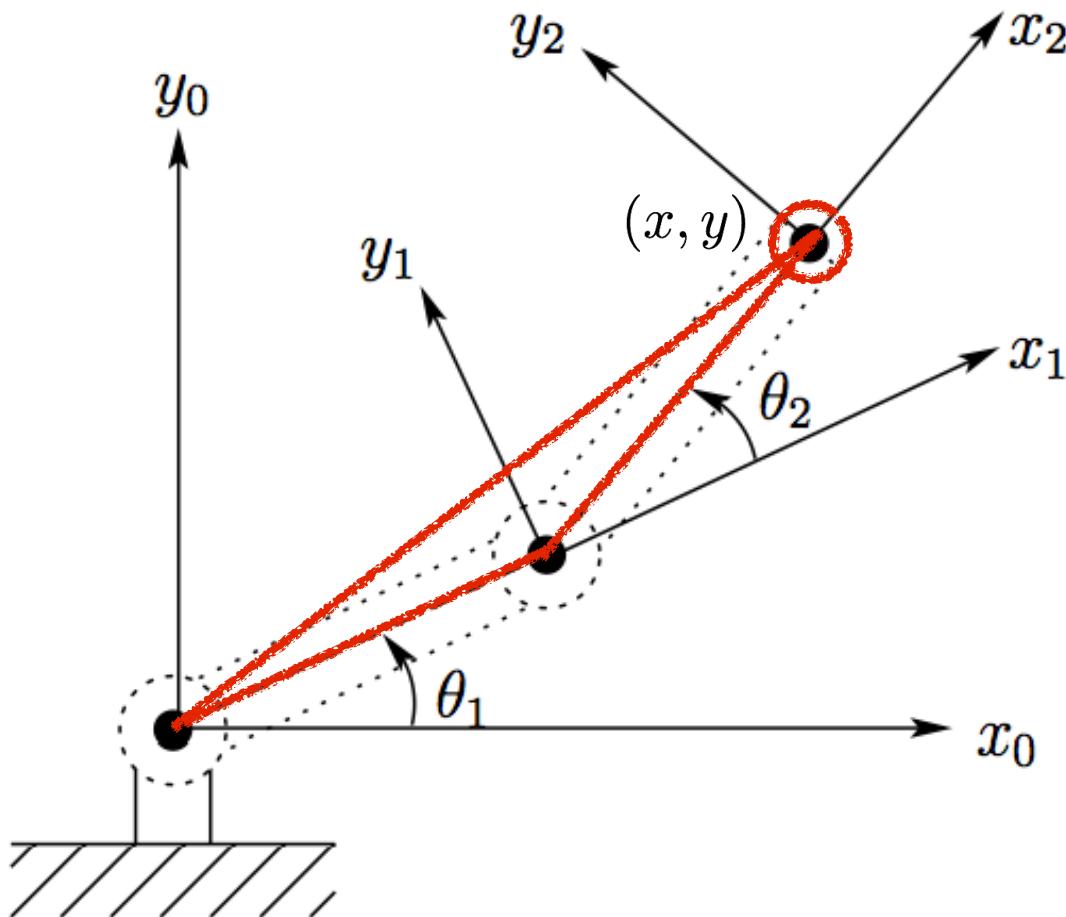
Inverse kinematics: $\mathbf{q} = f^{-1}([\mathbf{o}^0_N, \mathbf{R}^0_N])$ $[\theta_1, \theta_2] = f^{-1}(x, y)$



Inverse kinematics: $\mathbf{q} = f^{-1}([\mathbf{o}_N^0, \mathbf{R}_N^0])$ $[\theta_1, \theta_2] = f^{-1}(x, y)$



Inverse kinematics: $\mathbf{q} = f^{-1}([\mathbf{o}_N^0, \mathbf{R}_N^0])$

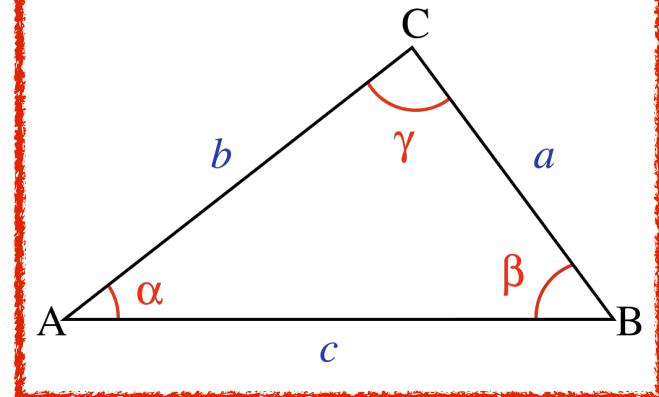


$$[\theta_1, \theta_2] = f^{-1}(x, y)$$

solve for θ_2

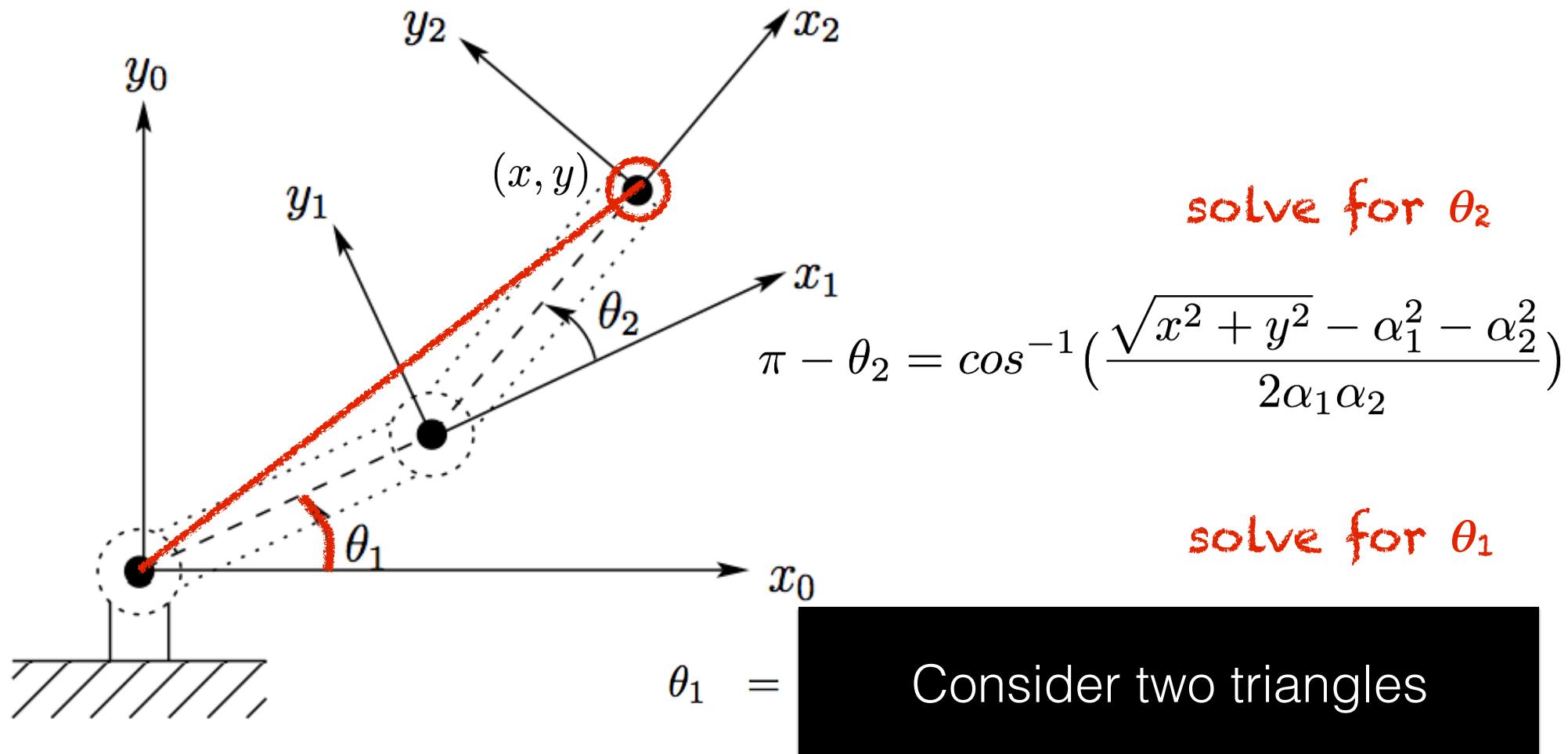
Law of Cosines

$$\gamma = \arccos \left(\frac{a^2 + b^2 - c^2}{2ab} \right)$$

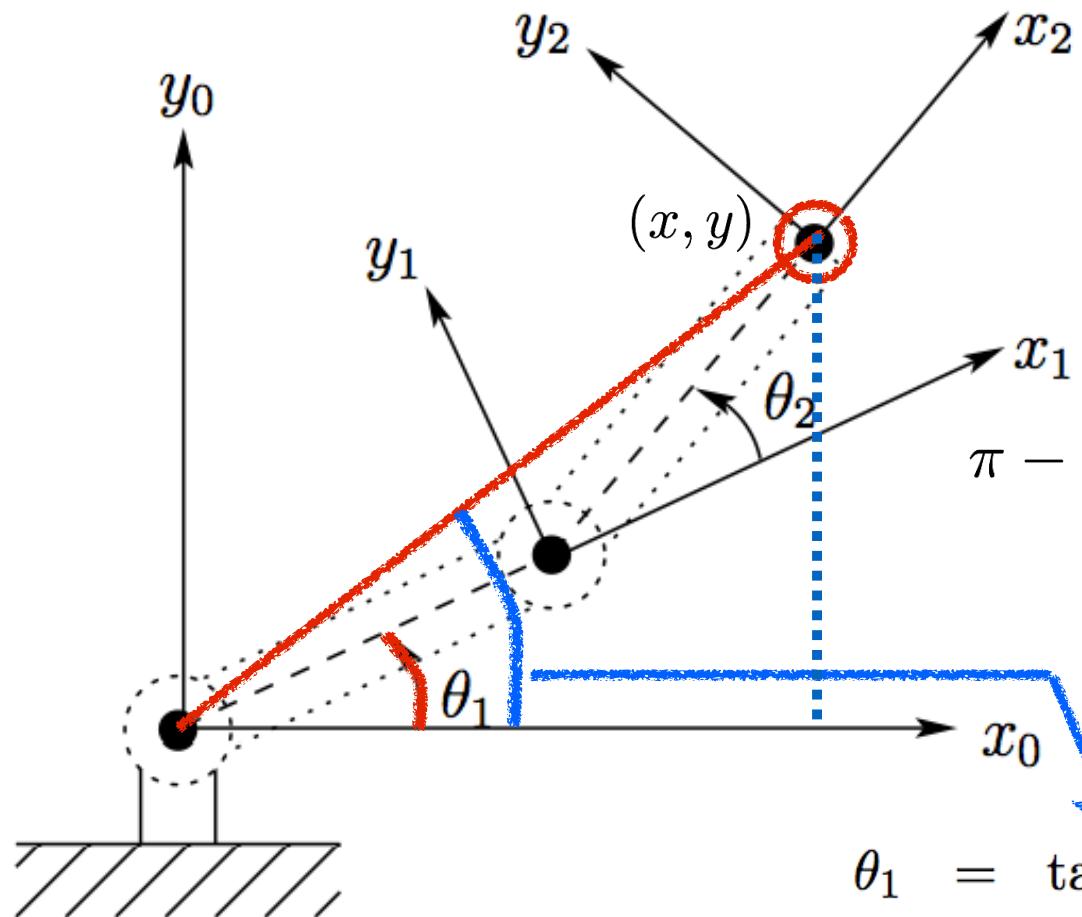


Inverse kinematics: $\mathbf{q} = f^{-1}([\mathbf{o}_N^0, \mathbf{R}_N^0])$

$$[\theta_1, \theta_2] = f^{-1}(x, y)$$



Inverse kinematics: $\mathbf{q} = f^{-1}([\mathbf{o}_N^0, \mathbf{R}_N^0])$ $[\theta_1, \theta_2] = f^{-1}(x, y)$



solve for θ_2

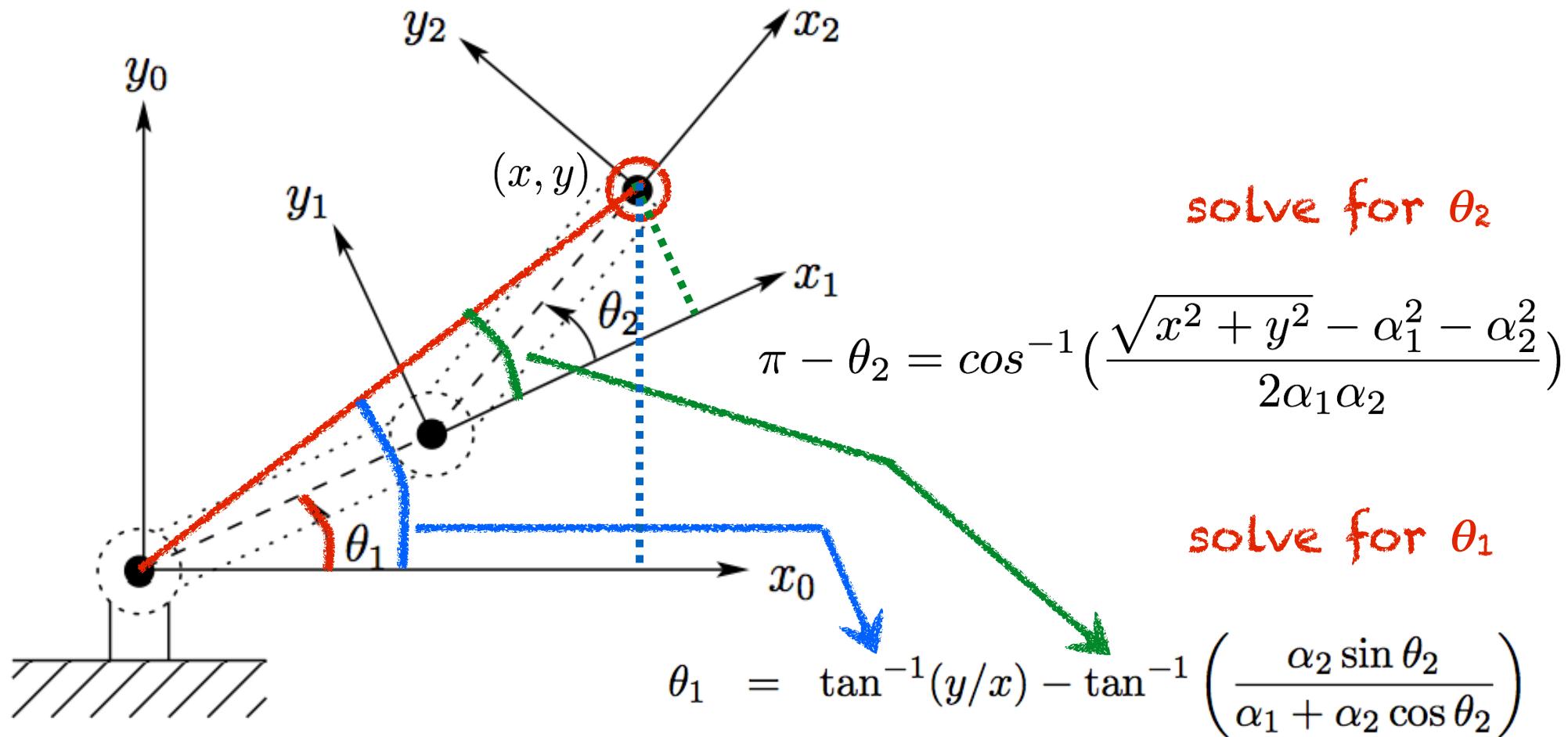
$$\pi - \theta_2 = \cos^{-1}\left(\frac{\sqrt{x^2 + y^2} - \alpha_1^2 - \alpha_2^2}{2\alpha_1\alpha_2}\right)$$

solve for θ_1

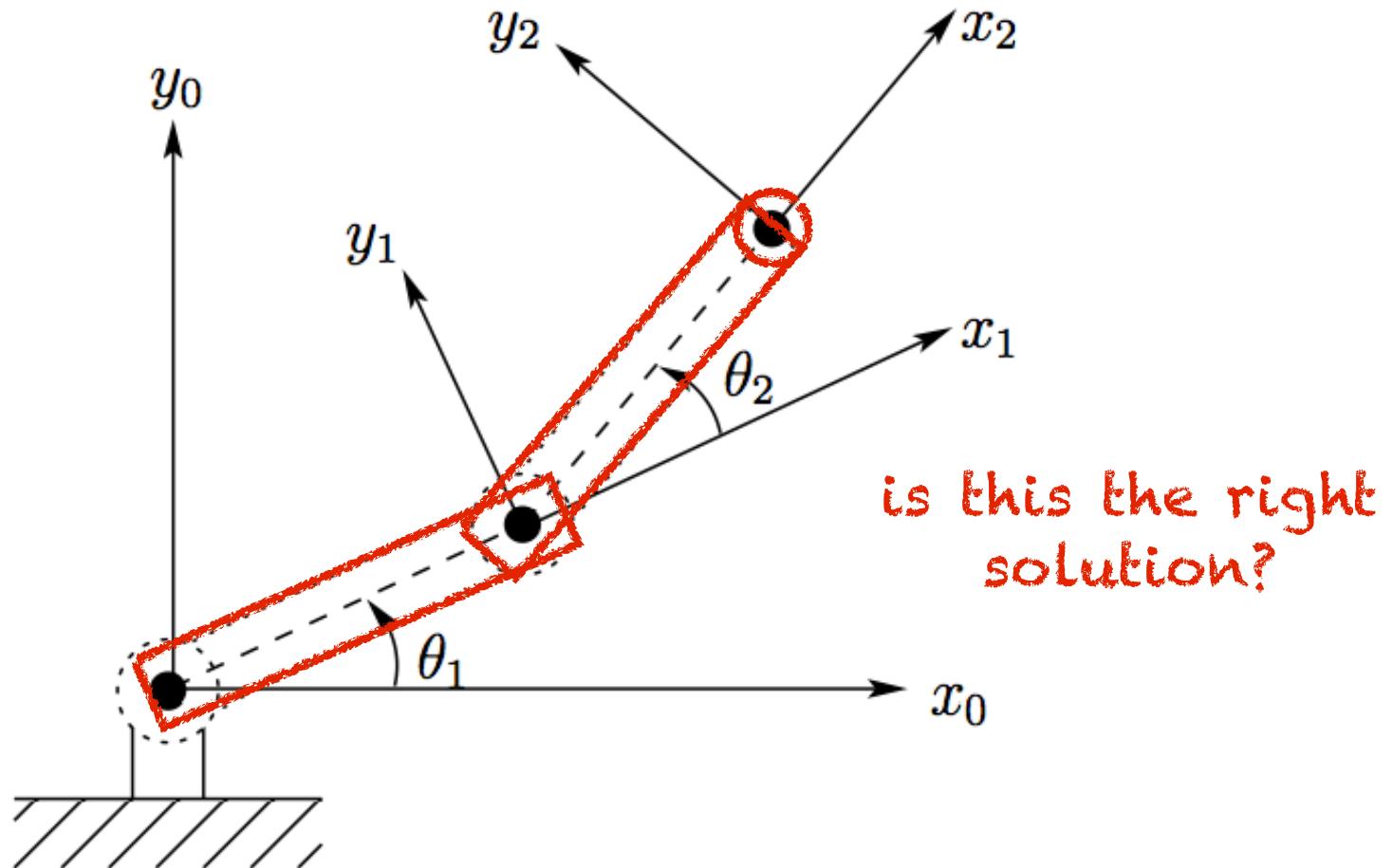
$$\theta_1 = \tan^{-1}(y/x) -$$

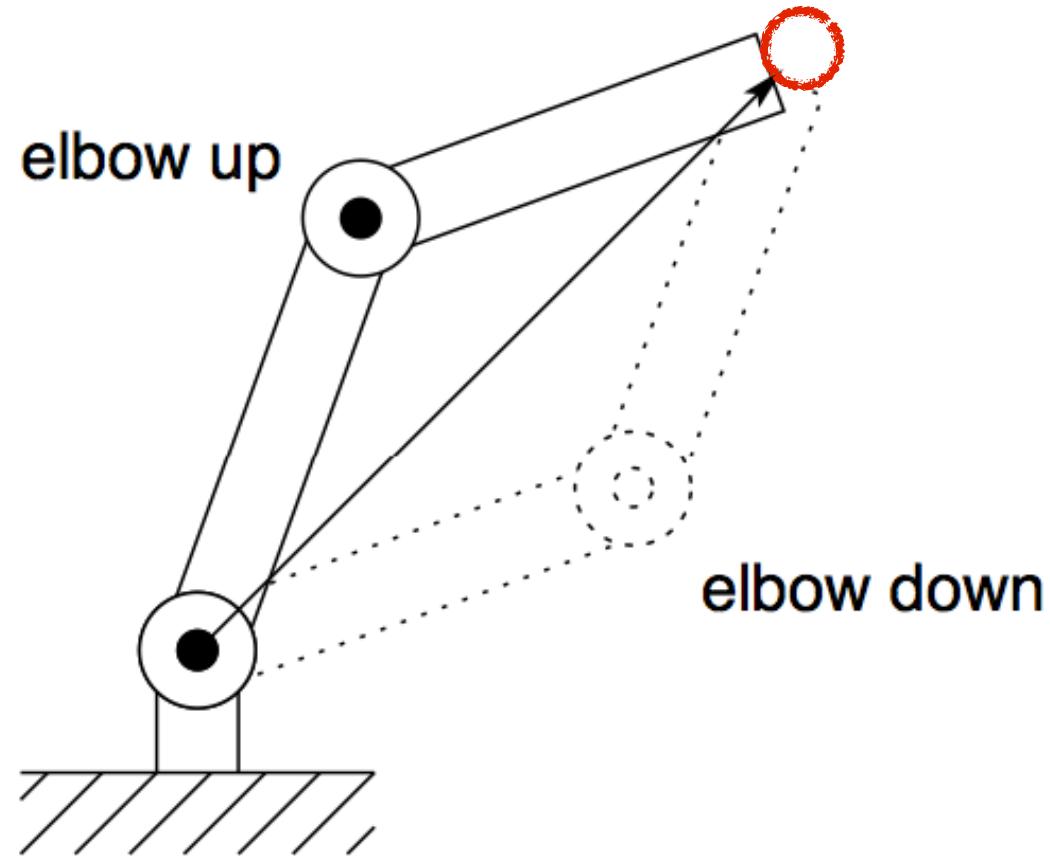
Consider two trian

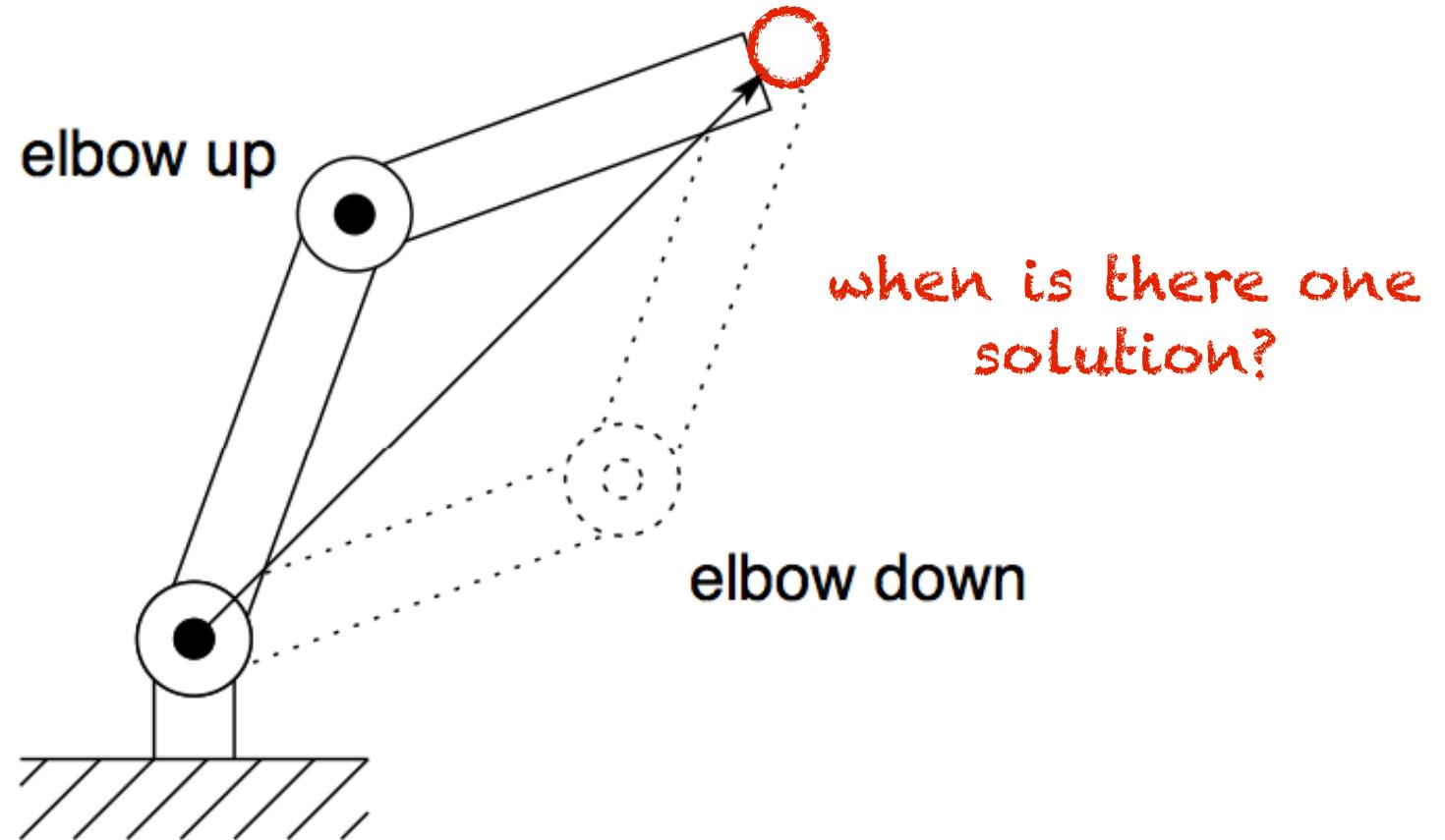
Inverse kinematics: $\mathbf{q} = f^{-1}([\mathbf{o}_N^0, \mathbf{R}_N^0])$ $[\theta_1, \theta_2] = f^{-1}(x, y)$



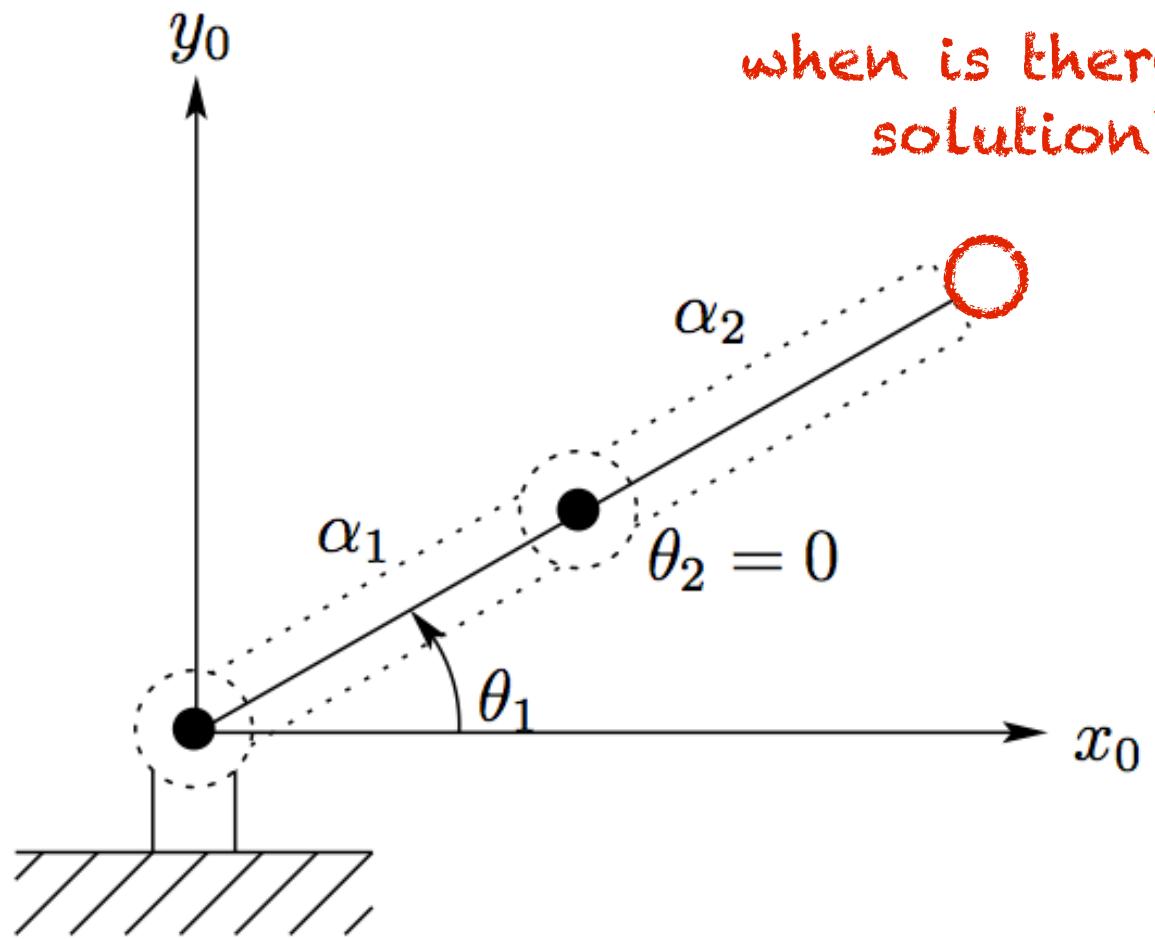
inverse kinematics: $(\theta_1, \theta_2) = f^{-1}(x, y)$

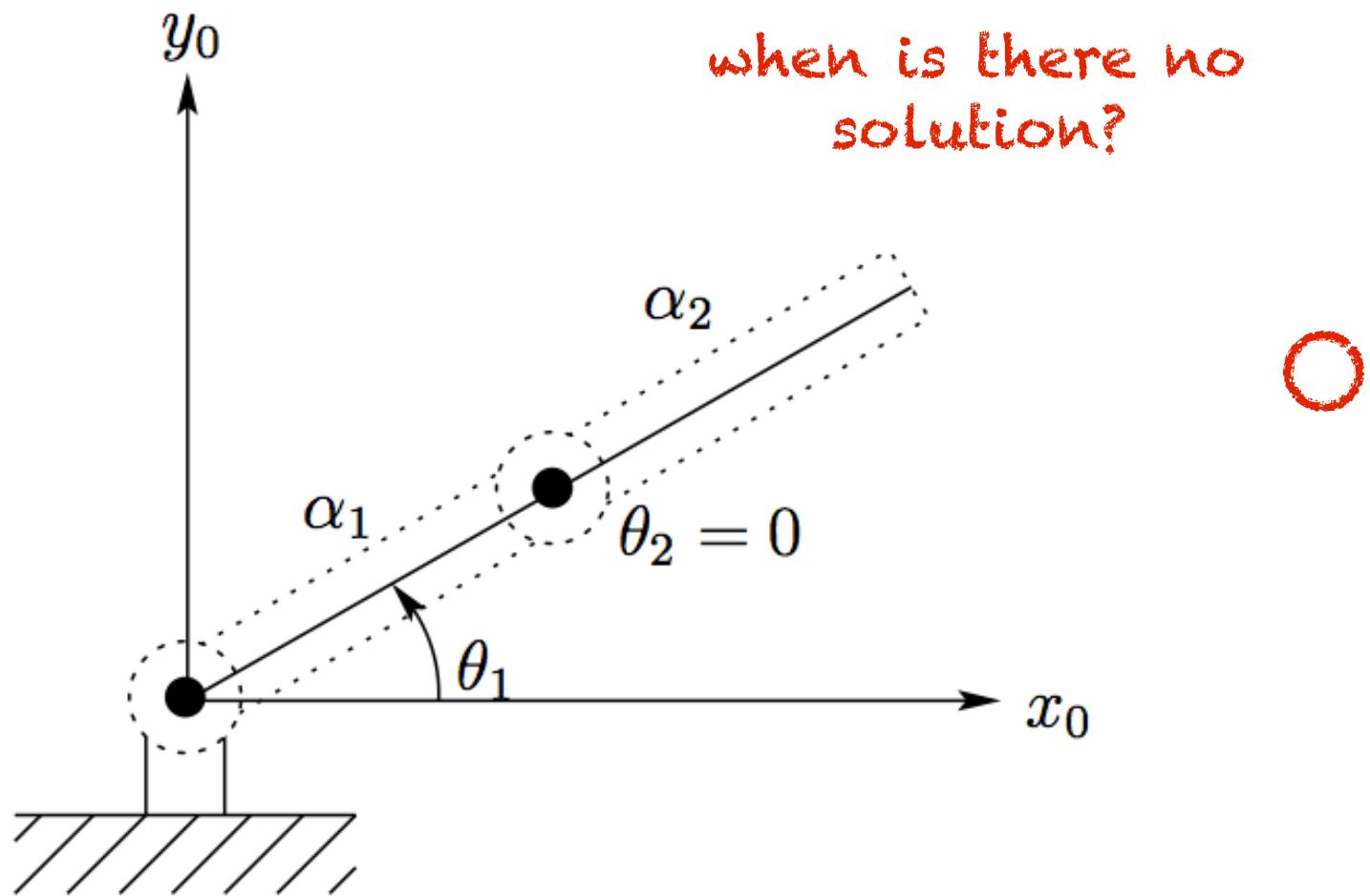






when is there no
solution?





Can we do IK for 3 links?

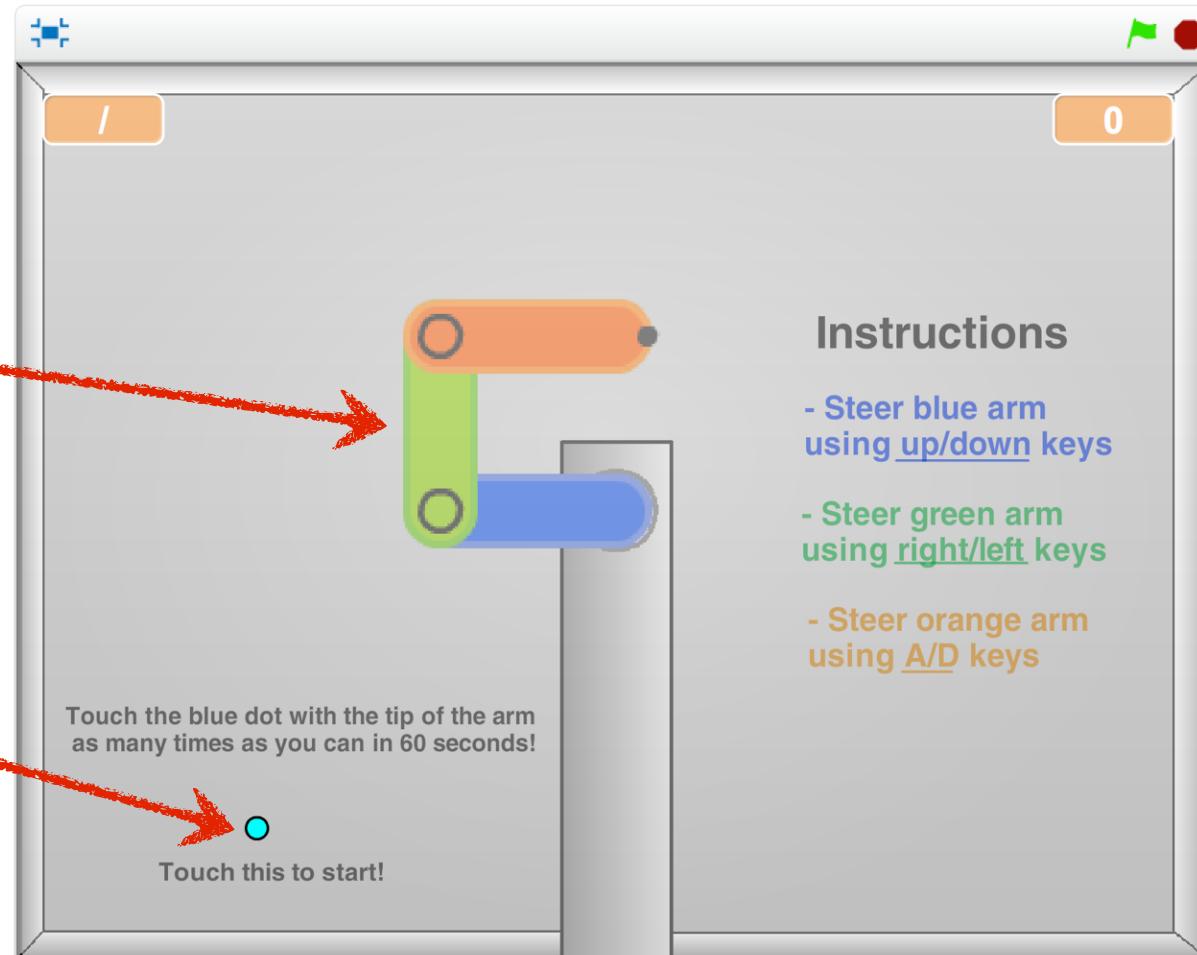
Try this



How many solutions for this arm?

3
unknowns

2
constraints



Remember:
 $Ax = b$

Inverse Kinematics: 2D

$$T_n^0(q_1, \dots, q_n) = H$$

Inverse Kinematics: 2D

Configuration

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \dots \\ q_n \end{bmatrix}$$

$$T_n^0(q_1, \dots, q_n)$$

$$= H$$

Transform from
endeffector frame
to world frame

$$H = \begin{bmatrix} R & o \\ 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} r_{11} & r_{12} & o_x \\ r_{21} & r_{22} & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

Inverse Kinematics: 2D

Configuration

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \dots \\ q_n \end{bmatrix}$$

$$T_n^0(q_1, \dots, q_n)$$

$$= H$$

Transform from
endeffector frame
to world frame

$$H = \begin{bmatrix} R & o \\ 0 & 1 \end{bmatrix}$$

Inverse orientation

$$R_n^0(q_1, \dots, q_n) = R$$

$$o_n^0(q_1, \dots, q_n) = o$$

$$H = \begin{bmatrix} r_{11} & r_{12} & o_x \\ r_{21} & r_{22} & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

Inverse position

Inverse Kinematics: 2D

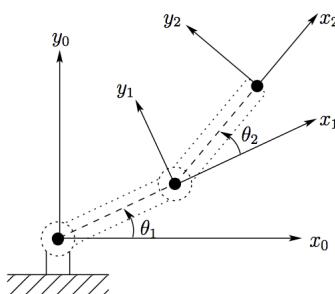
Configuration

$$T_n^0(q_1, \dots, q_n) = H$$

Transform from
endeffector

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

$$H = \begin{bmatrix} R & o \\ 0 & 1 \end{bmatrix}$$



Closed form solution

$$H = \begin{bmatrix} r_{11} & r_{12} & o_x \\ r_{21} & r_{22} & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

Inverse Kinematics: 2D

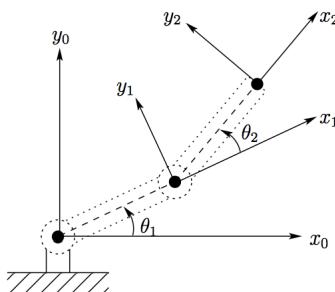
Configuration

$$T_n^0(q_1, \dots, q_n) = H$$

Transform from
endeffector

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

$$H = \begin{bmatrix} R & o \\ 0 & 1 \end{bmatrix}$$



Closed form solution

$$\theta_2 = \cos^{-1}\left(\frac{x^2 + y^2 - \alpha_1^2 - \alpha_2^2}{2\alpha_1\alpha_2}\right)$$

$$\theta_1 = \tan^{-1}(y/x) - \tan^{-1}\left(\frac{\alpha_2 \sin \theta_2}{\alpha_1 + \alpha_2 \cos \theta_2}\right)$$

$$H = \begin{bmatrix} r_{11} & r_{12} & o_x \\ r_{21} & r_{22} & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

Inverse Kinematics: 3D

Configuration

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \dots \\ q_n \end{bmatrix}$$

6 DOF position and orientation of endeffector

$$T_n^0(q_1, \dots, q_n)$$

$$H$$

Transform from endeffector

$$H = \begin{bmatrix} R & o \\ 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} r_{11} & r_{12} & r_{13} & o_x \\ r_{21} & r_{22} & r_{23} & o_y \\ r_{31} & r_{32} & r_{33} & o_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

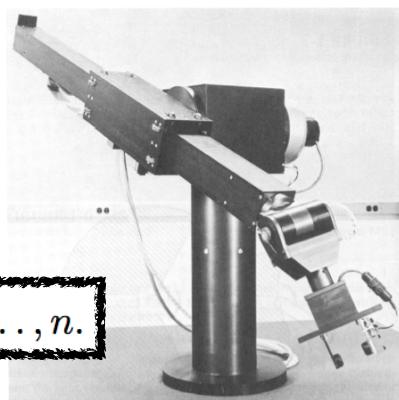
Inverse Kinematics: 3D

Configuration

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \dots \\ q_6 \end{bmatrix}$$

Closed form
solution?

$$q_k = f_k(h_{11}, \dots, h_{34}), \quad k = 1, \dots, n.$$



$$T_n^0(q_1, \dots, q_n)$$

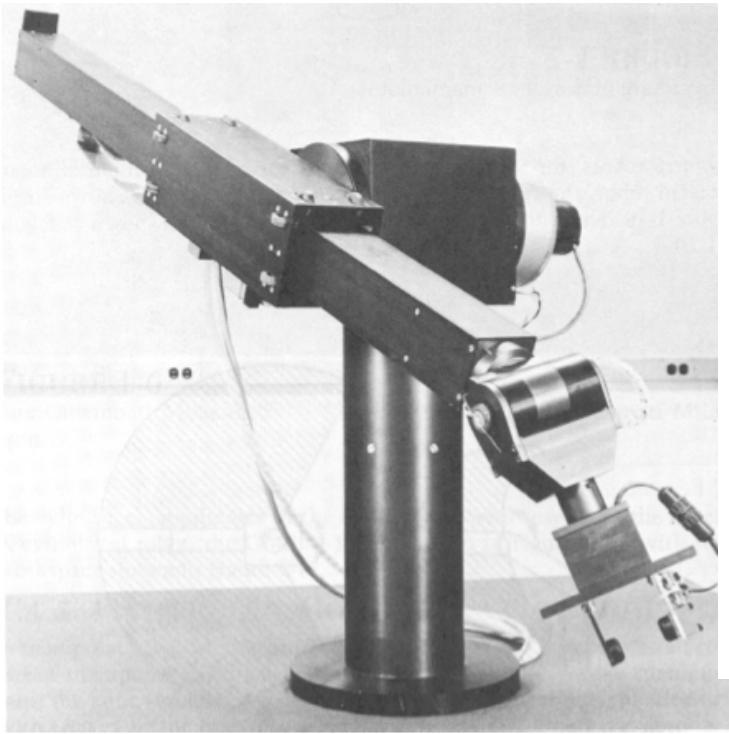
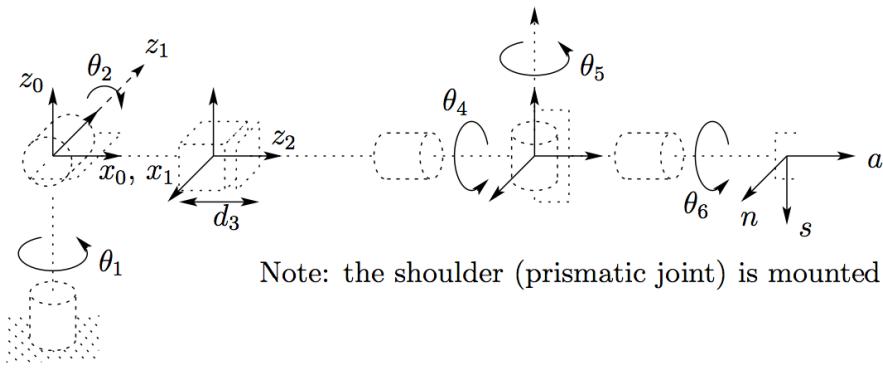
$$= H$$

Transform from
endeffector

$$H = \begin{bmatrix} R & o \\ 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} r_{11} & r_{12} & r_{13} & o_x \\ r_{21} & r_{22} & r_{23} & o_y \\ r_{31} & r_{32} & r_{33} & o_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

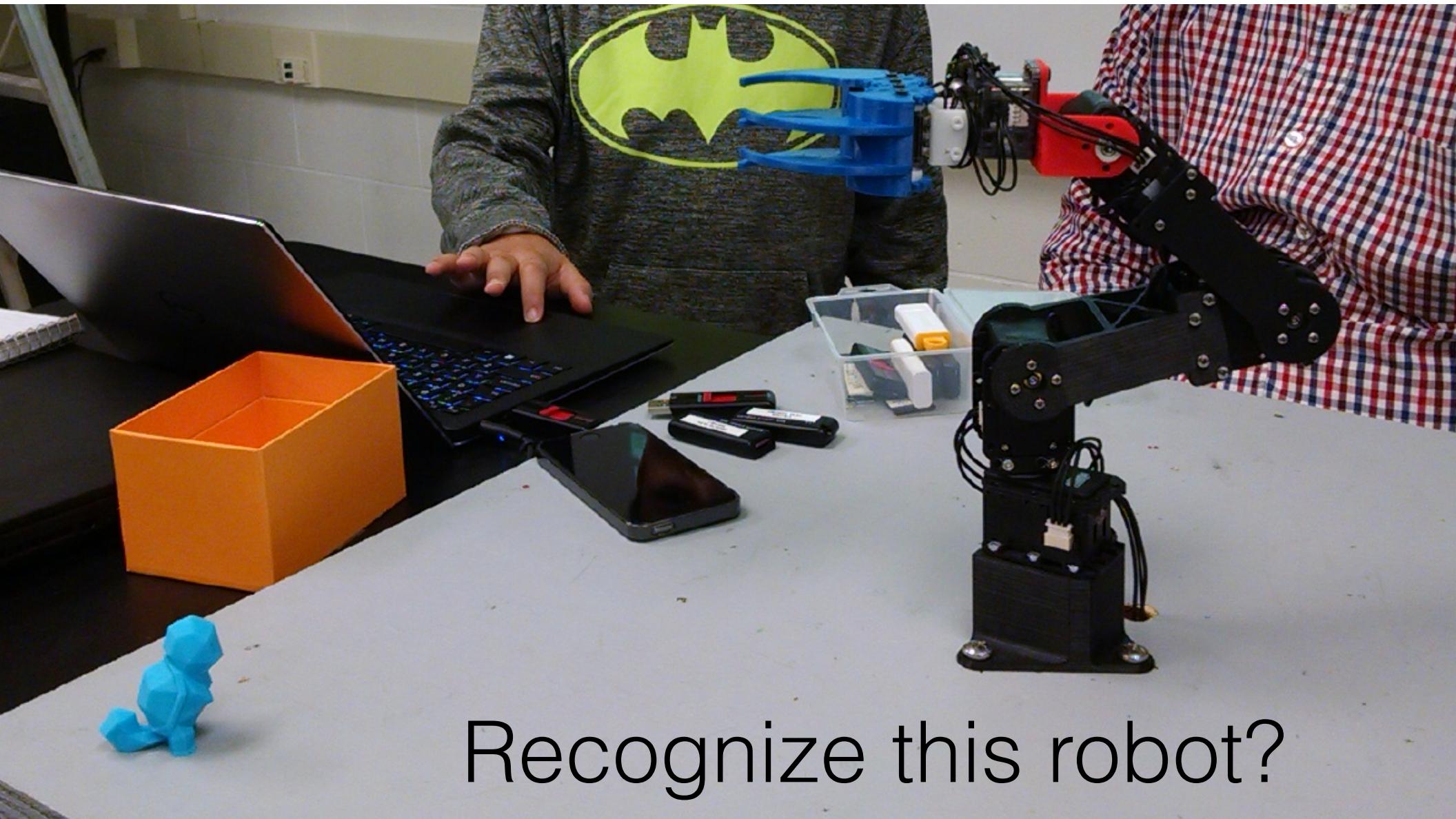
6 DOF position and
orientation of endeffector
UM EECS 398/598 - autorob.github.io



Stanford Manipulator

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

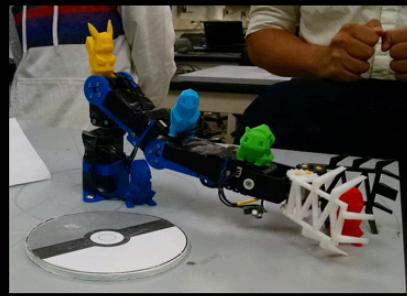
assumes D-H frames
UM EECS 398/598 - autorob.github.io



Recognize this robot?



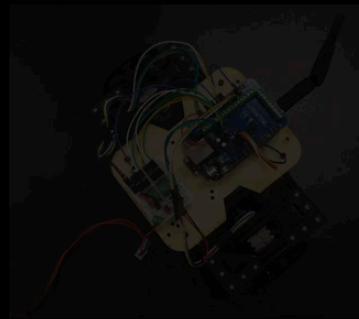
ArmLab



Catch em' All



BotLab

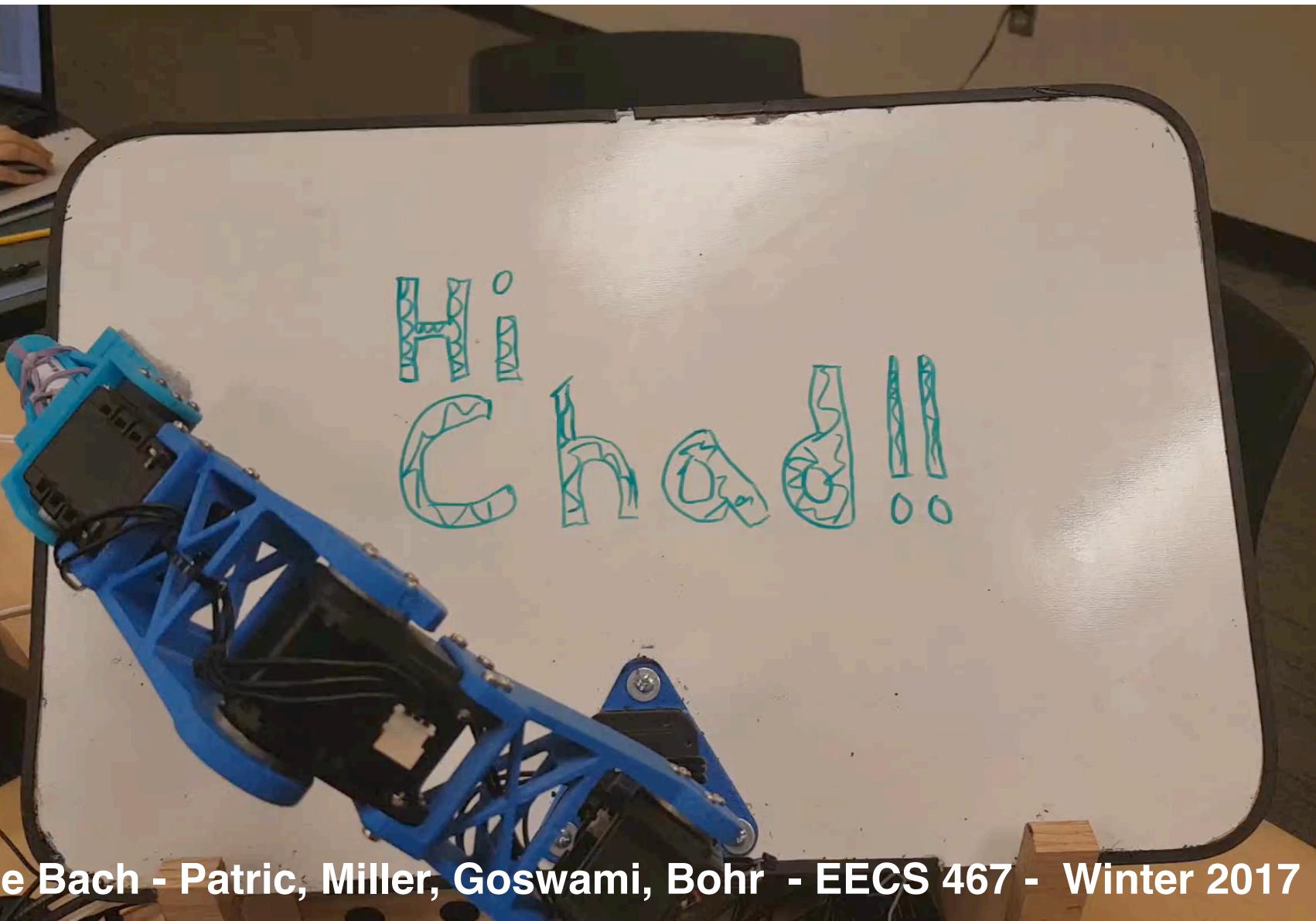


BalanceBot

ROB 550 - Fall 2016



Robot Air Hockey - Attisha, Jiminez, Mulani - EECS 467 - Winter 2017



I'll Be Bach - Patric, Miller, Goswami, Bohr - EECS 467 - Winter 2017



x16

RoBob Ross - Rabideau, Ekins, Rao, Jones - EECS 467 - Winter 2017



Me

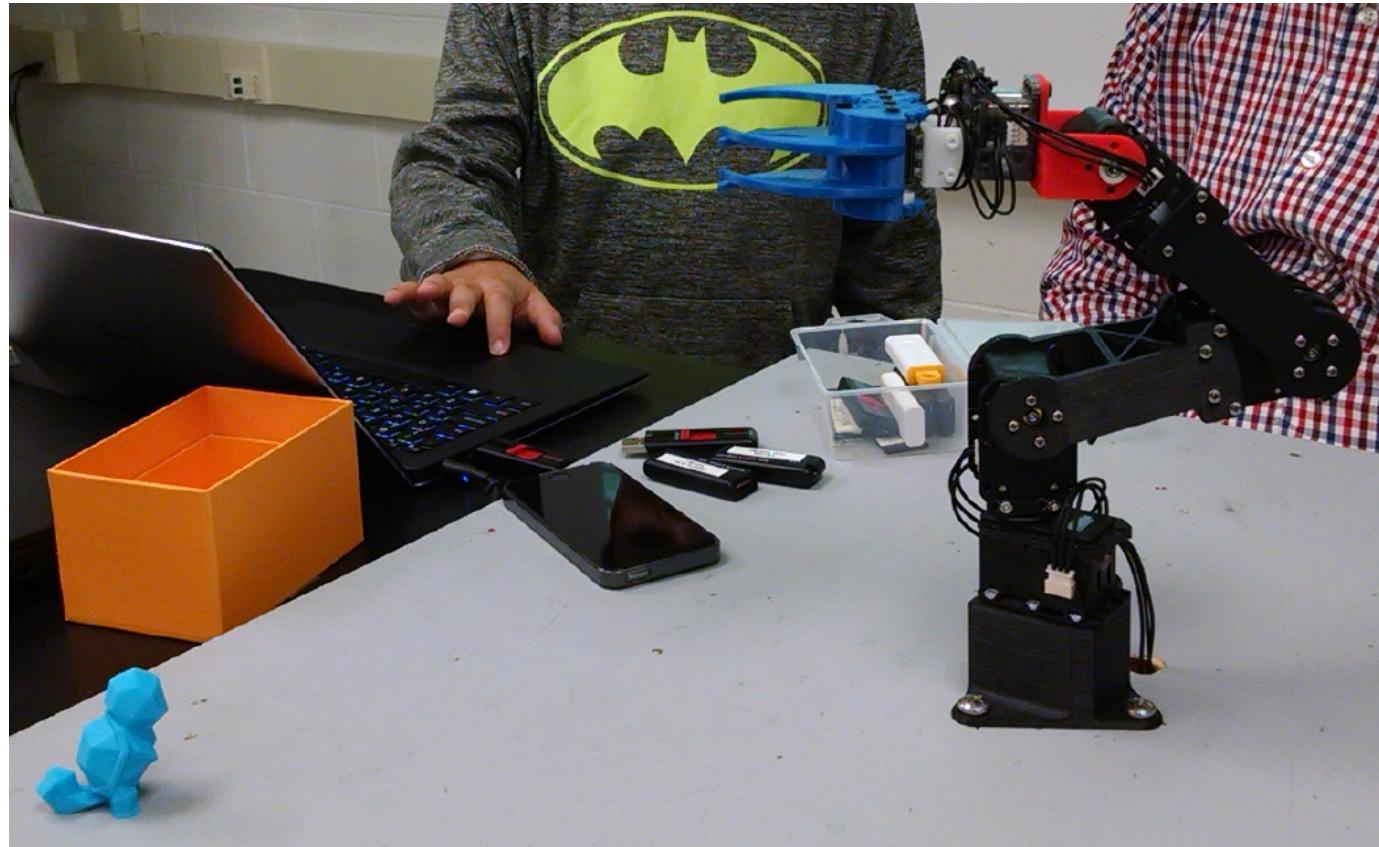


Me as viewed by
RoBob Ross

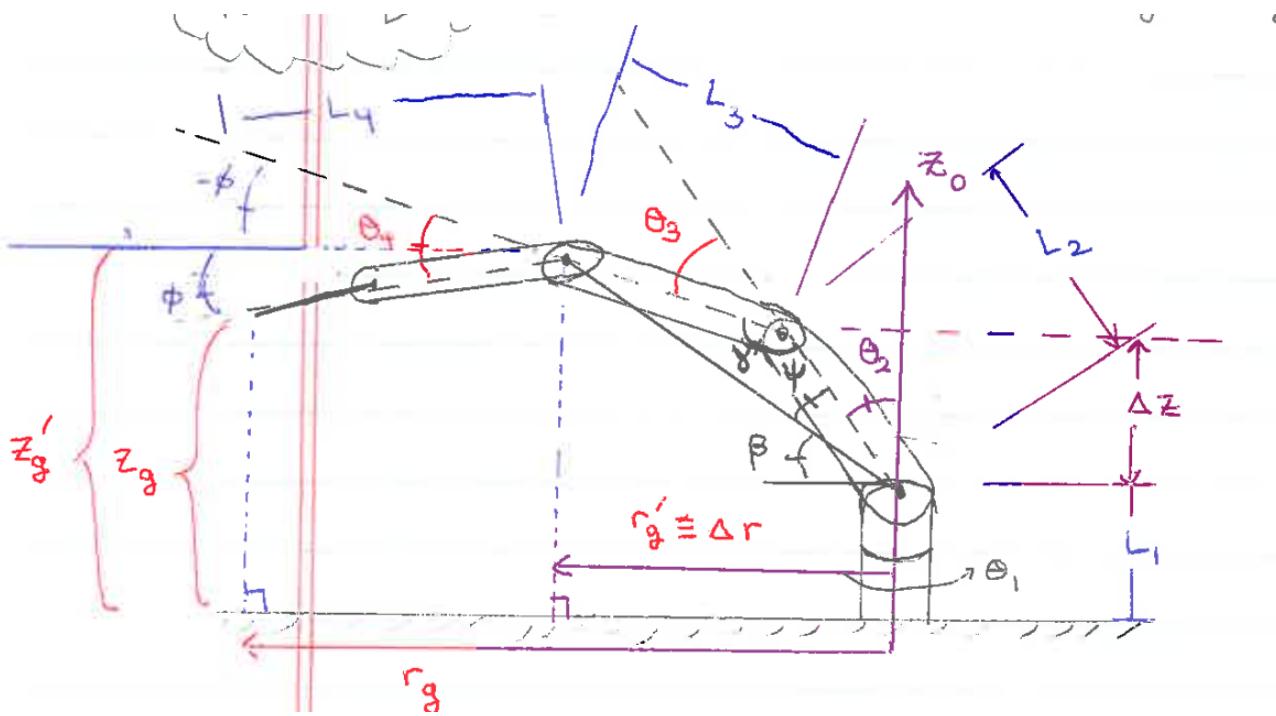


Me as drawn by
RoBob Ross

RexArm (EECS 467 / ROB 550)



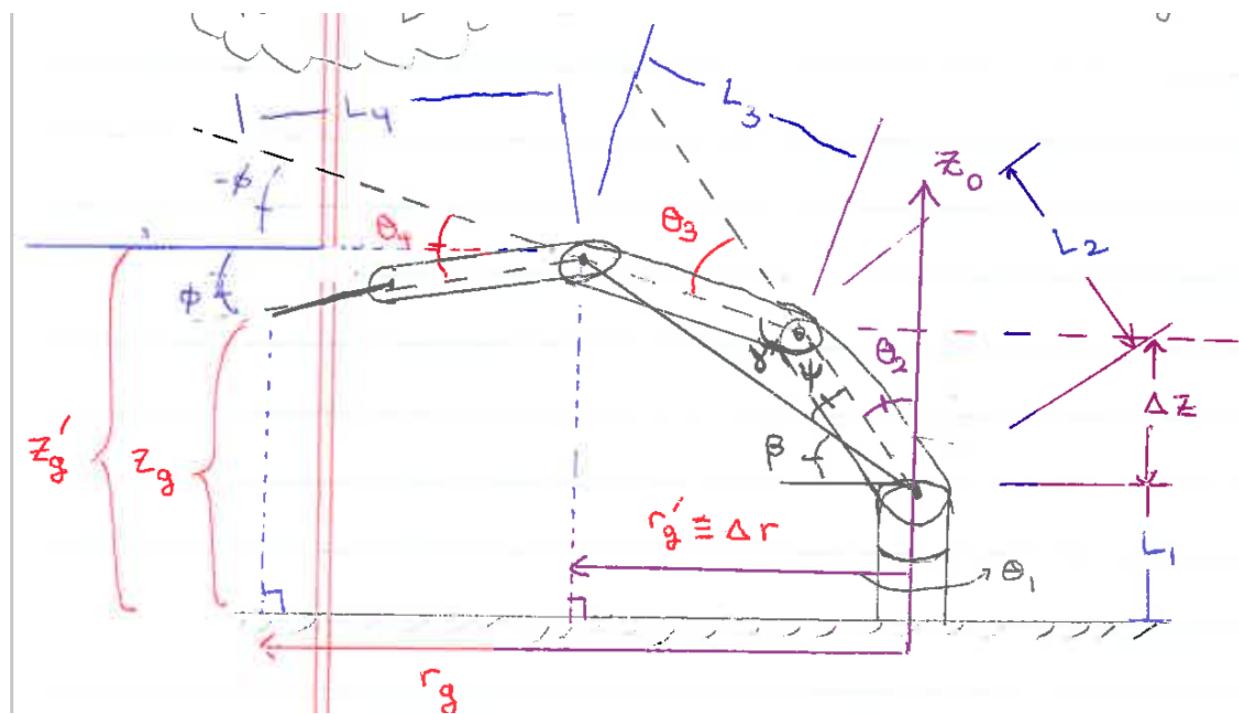
RexArm (EECS 467 / ROB 550)



Find: configuration
 $\mathbf{q} = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4]$
as robot joint angles

Given:

Find: configuration
 $\mathbf{q} = [\theta_1 \theta_2 \theta_3 \theta_4]$
as robot joint angles



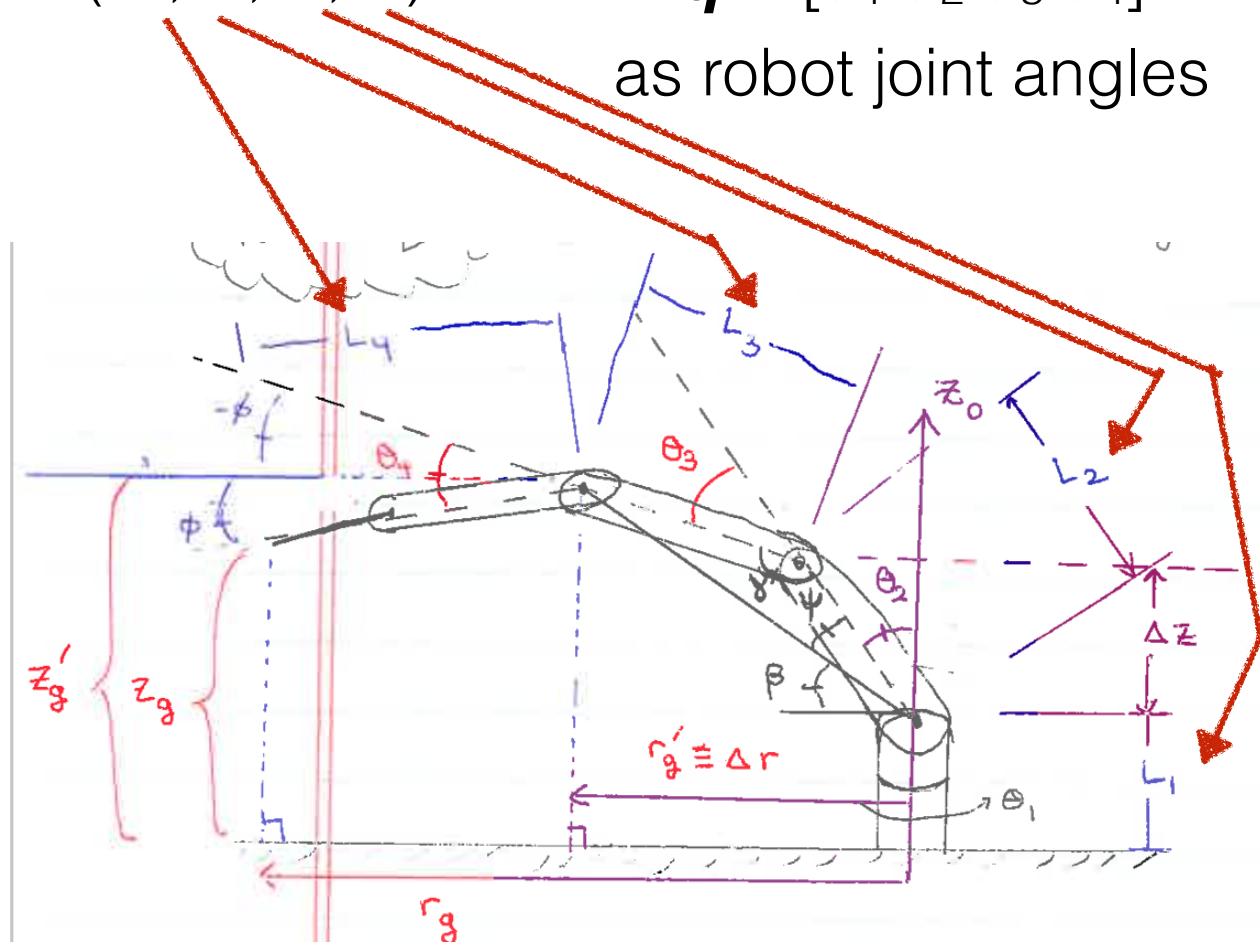
Given:

link lengths (L_4, L_3, L_2, L_1)

Find: configuration

$$\mathbf{q} = [\theta_1 \theta_2 \theta_3 \theta_4]$$

as robot joint angles



Given:

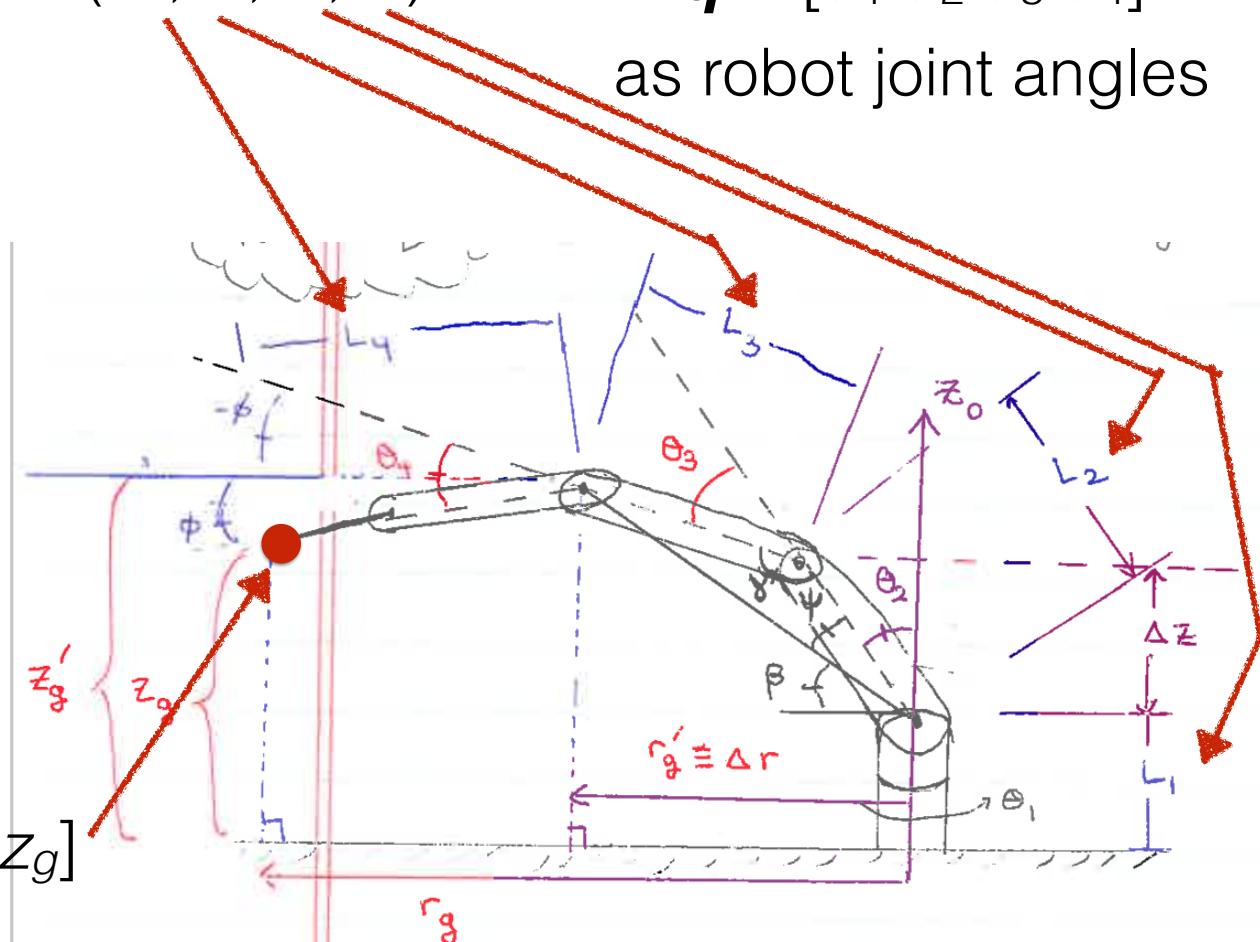
link lengths (L_4, L_3, L_2, L_1)

endeffector position $[x_g \ y_g \ z_g]$
wrt. base frame

Find: configuration

$$\mathbf{q} = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4]$$

as robot joint angles



Given:

link lengths (L_4, L_3, L_2, L_1)

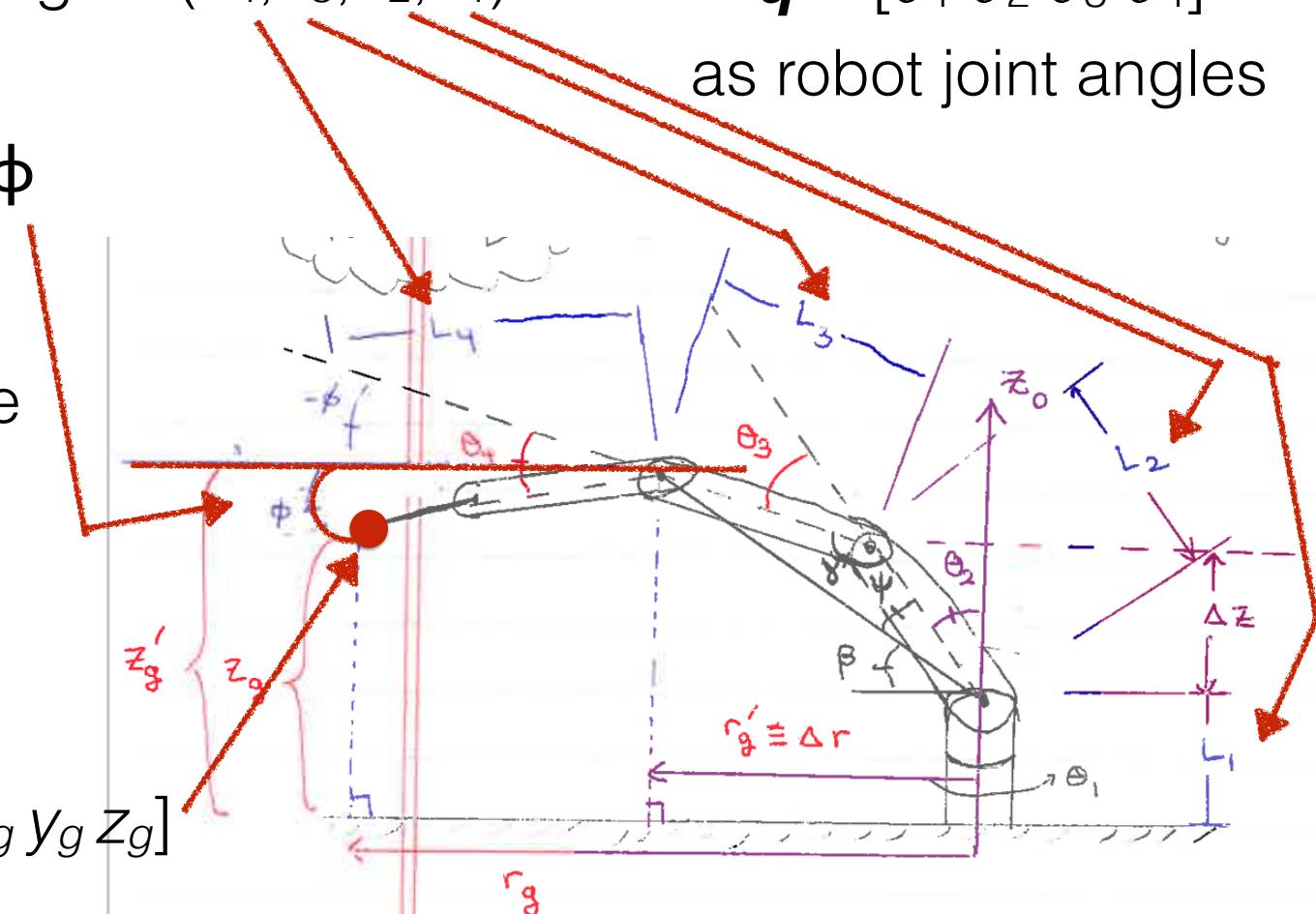
endeffector orientation ϕ
as angle wrt. plane
centered at o_3 and
parallel to ground plane

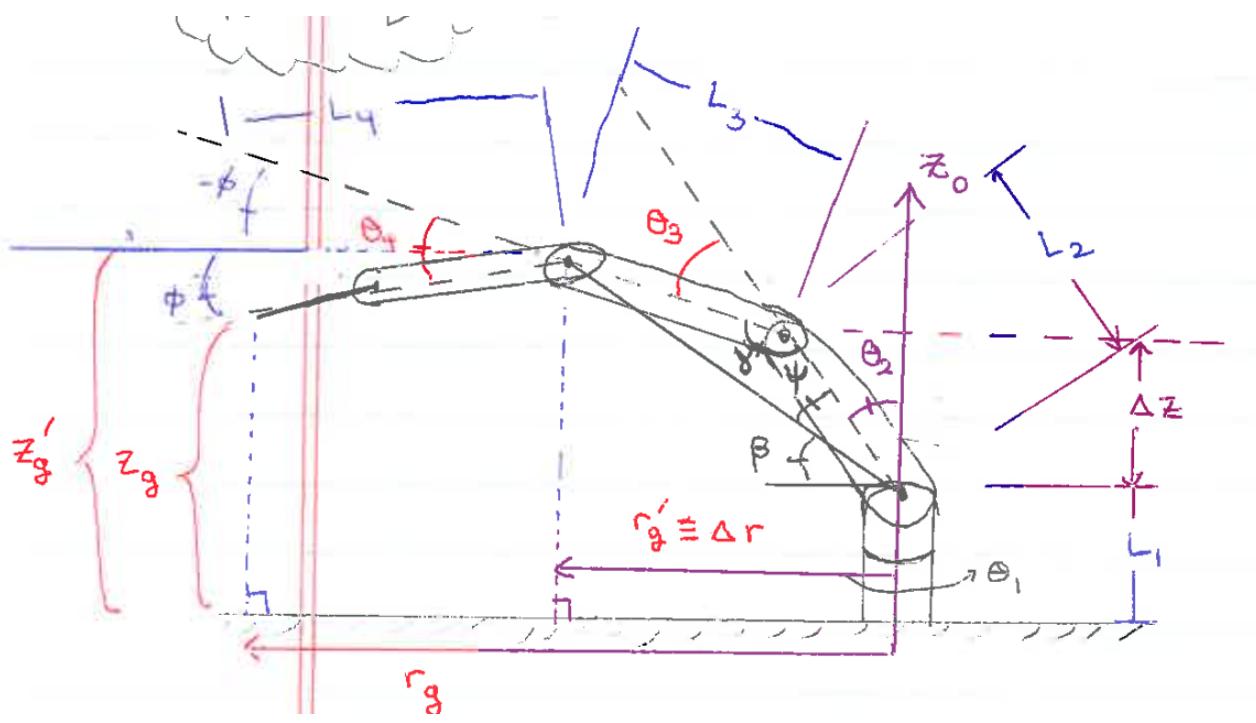
endeffector position $[x_g \ y_g \ z_g]$
wrt. base frame

Find: configuration

$$\mathbf{q} = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4]$$

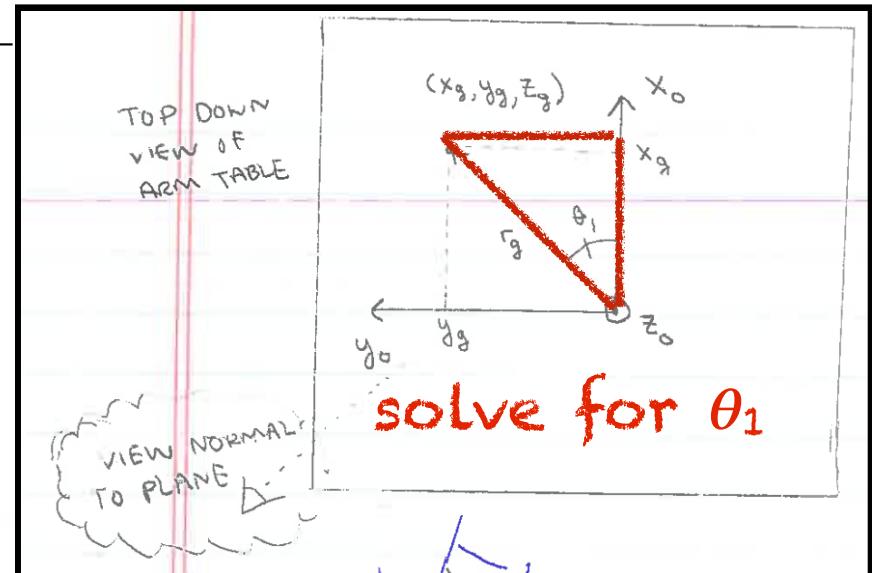
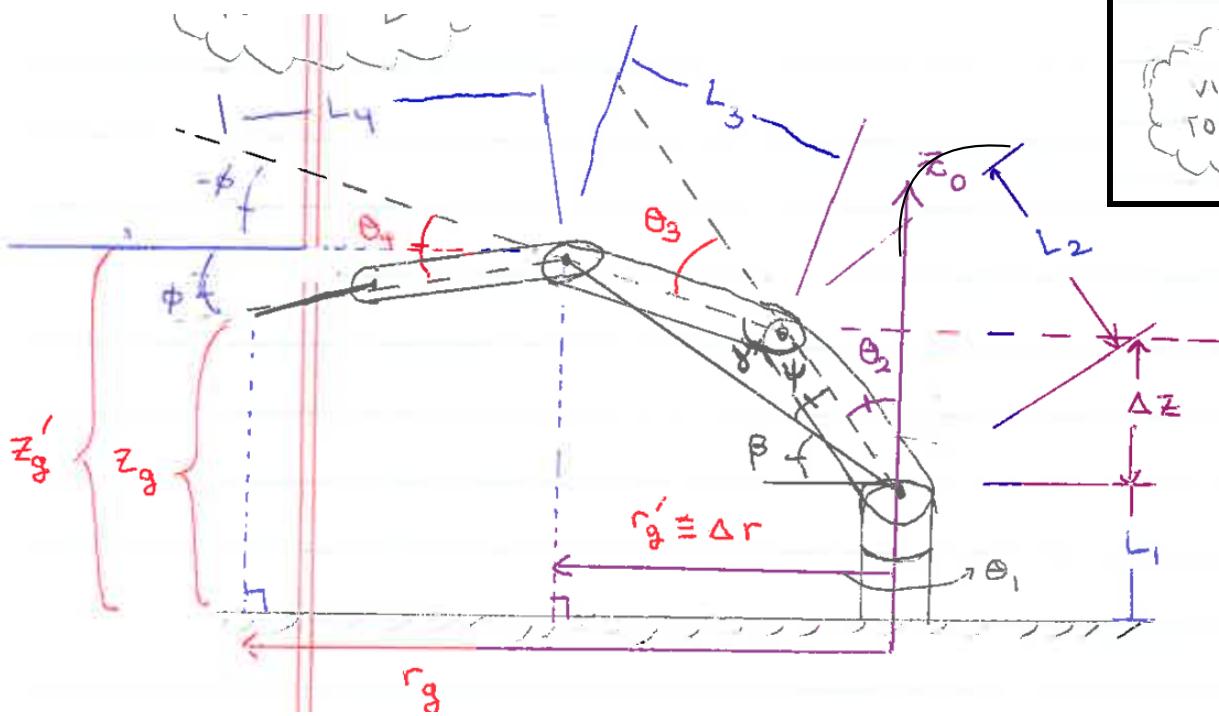
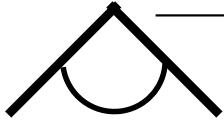
as robot joint angles



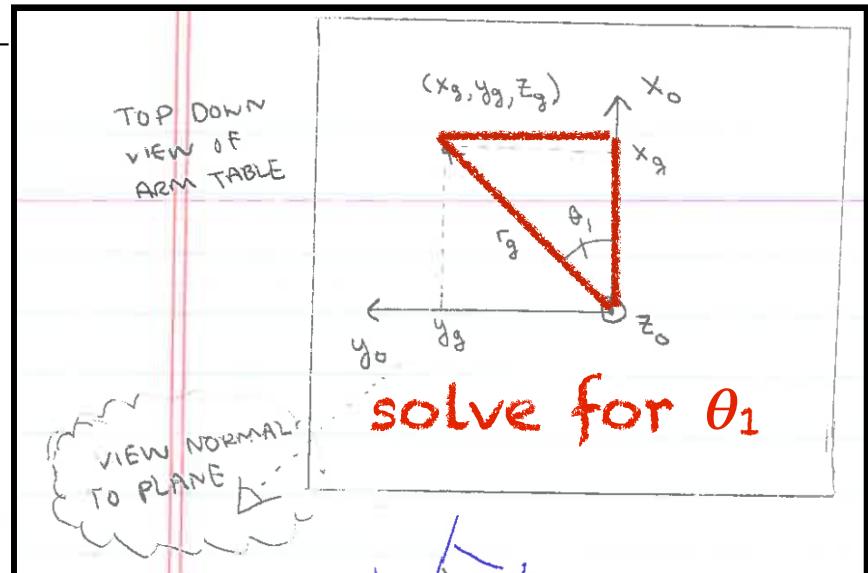
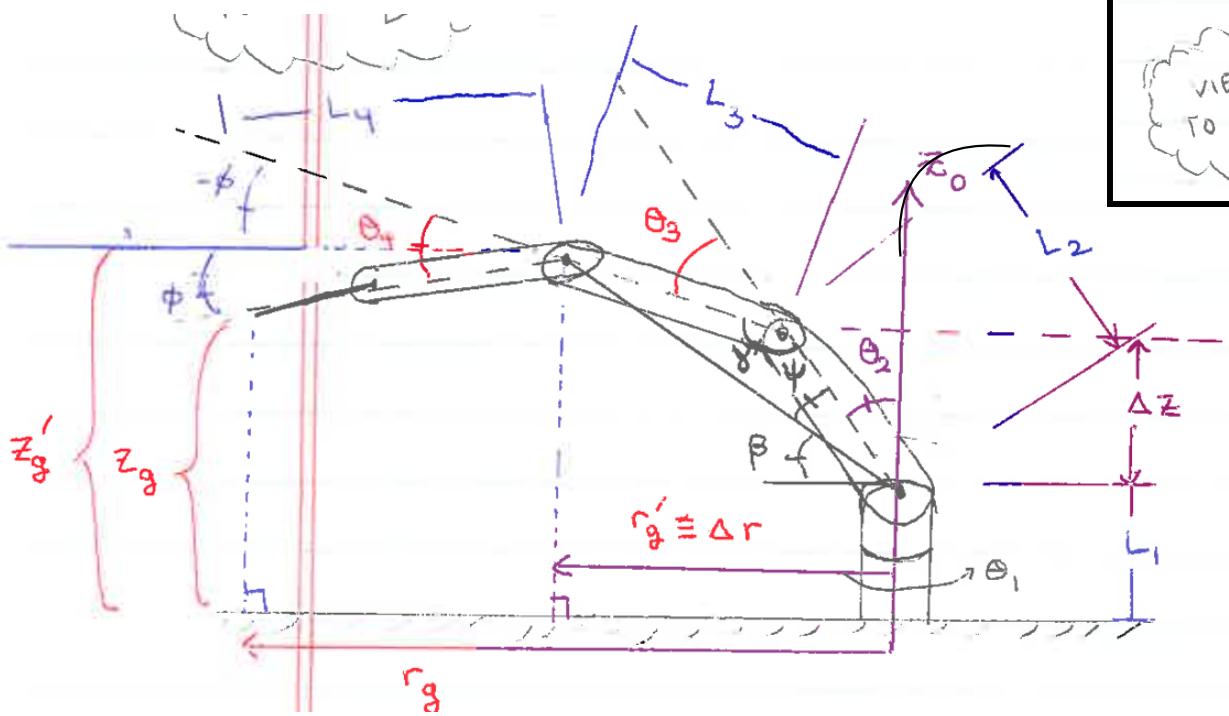
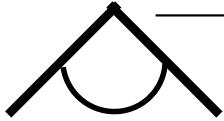


solve for θ_1

overhead view



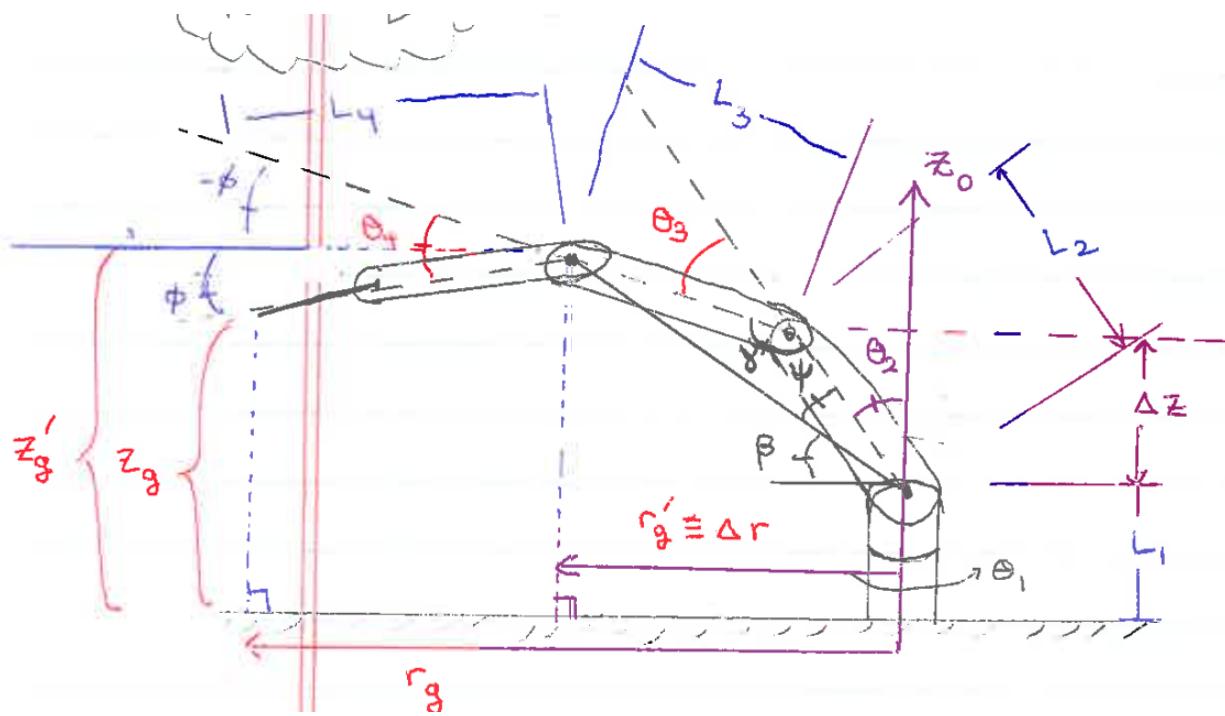
overhead view



$$\theta_1 = \text{atan2}(y_g, x_g)$$

solve for θ_1

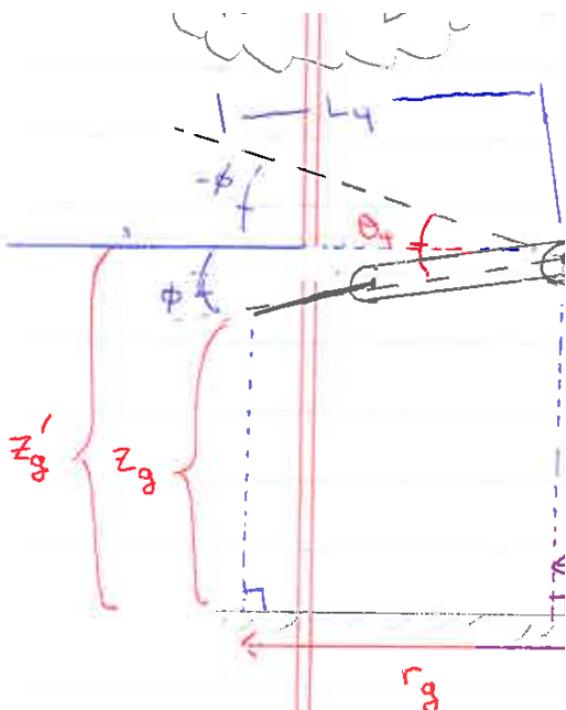
$$\Theta_1 = \text{atan2}(y_g, x_g)$$



solve for θ_3

solve for θ_1

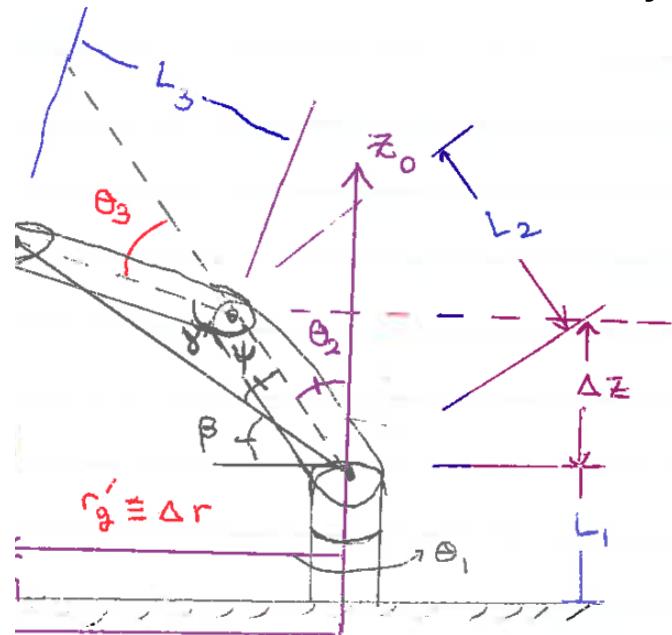
$$\Theta_1 = \text{atan2}(y_g, x_g)$$



Decoupling:

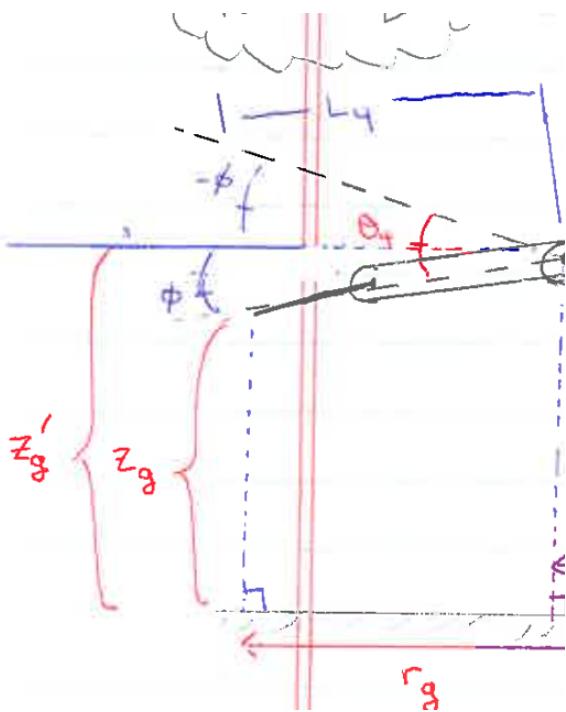
separate endeffector from
rest of the robot at last joint

solve for θ_3



solve for θ_1

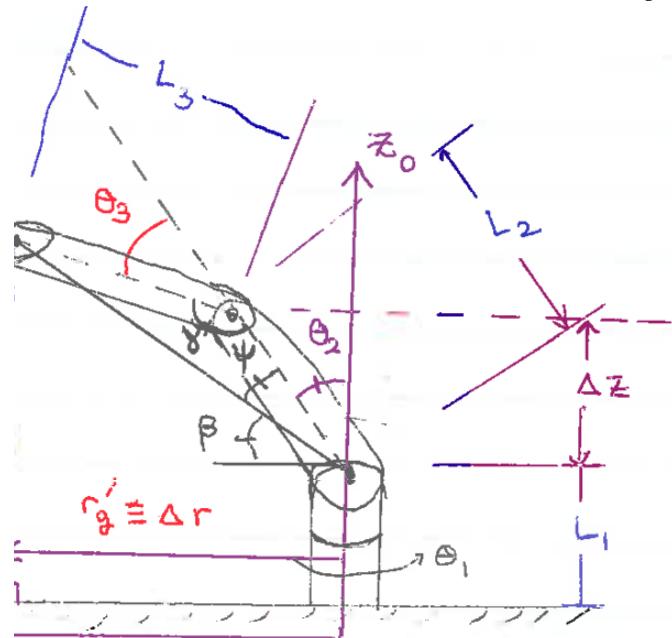
$$\Theta_1 = \text{atan2}(y_g, x_g)$$



Decoupling:

separate endeffector from
rest of the robot at last joint

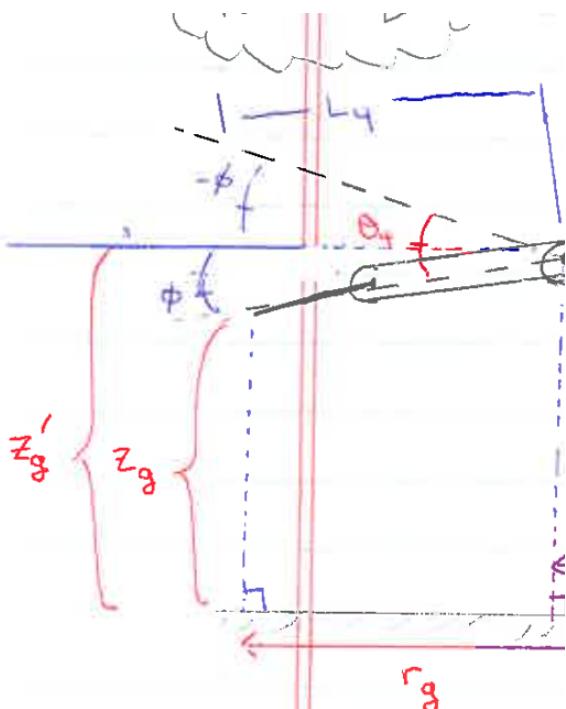
solve for θ_3



and...

solve for θ_1

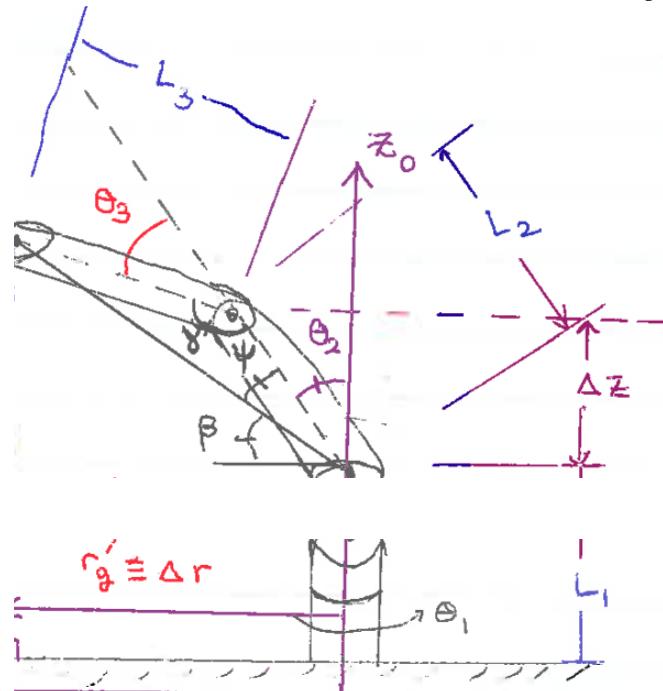
$$\theta_1 = \text{atan}2(y_g, x_g)$$



Decoupling:

separate endeffector from
rest of the robot at last joint

solve for θ_3

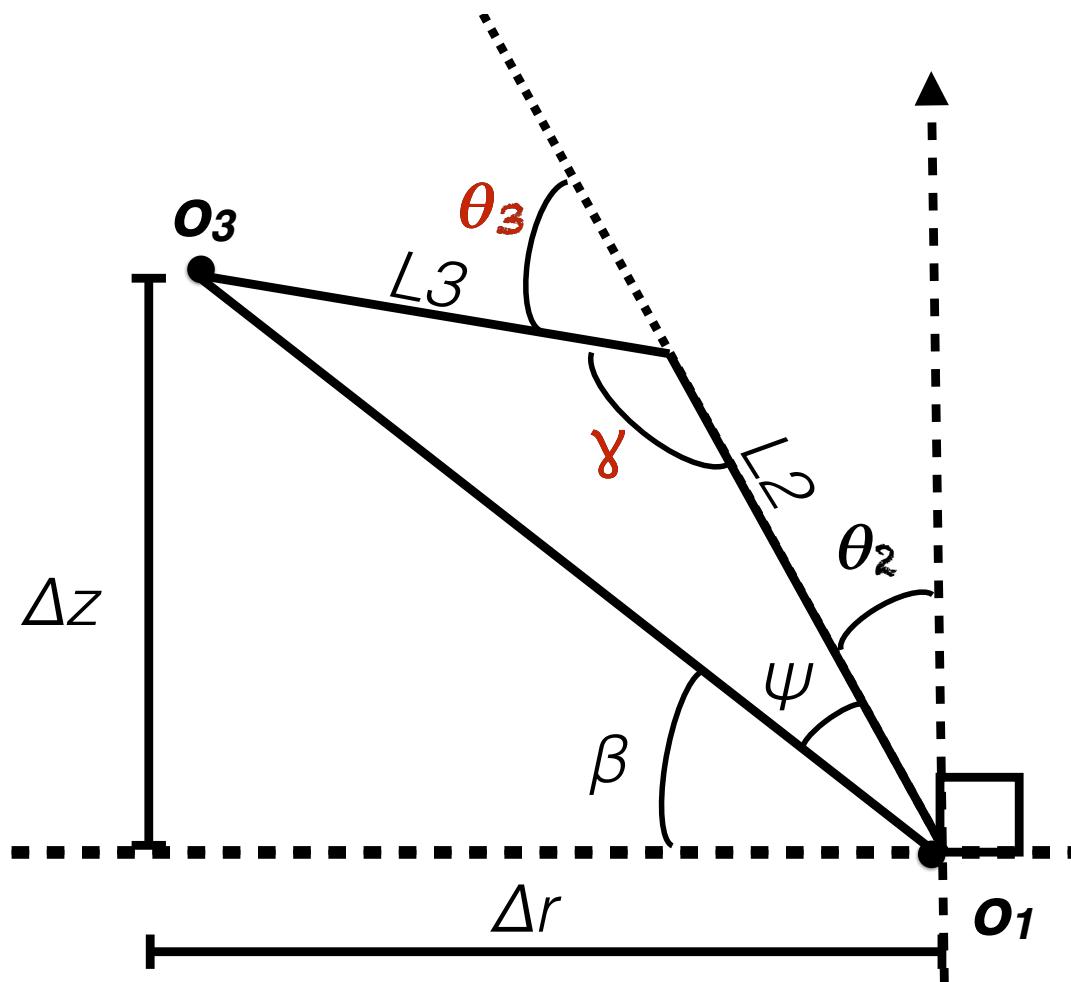
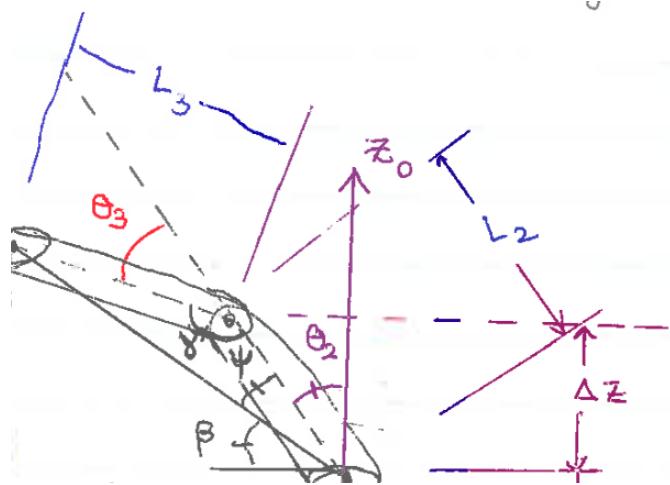


and joint 1 from rest
of robot

solve for θ_1

$$\Theta_1 = \text{atan2}(\gamma_g, x_g)$$

solve for θ_3



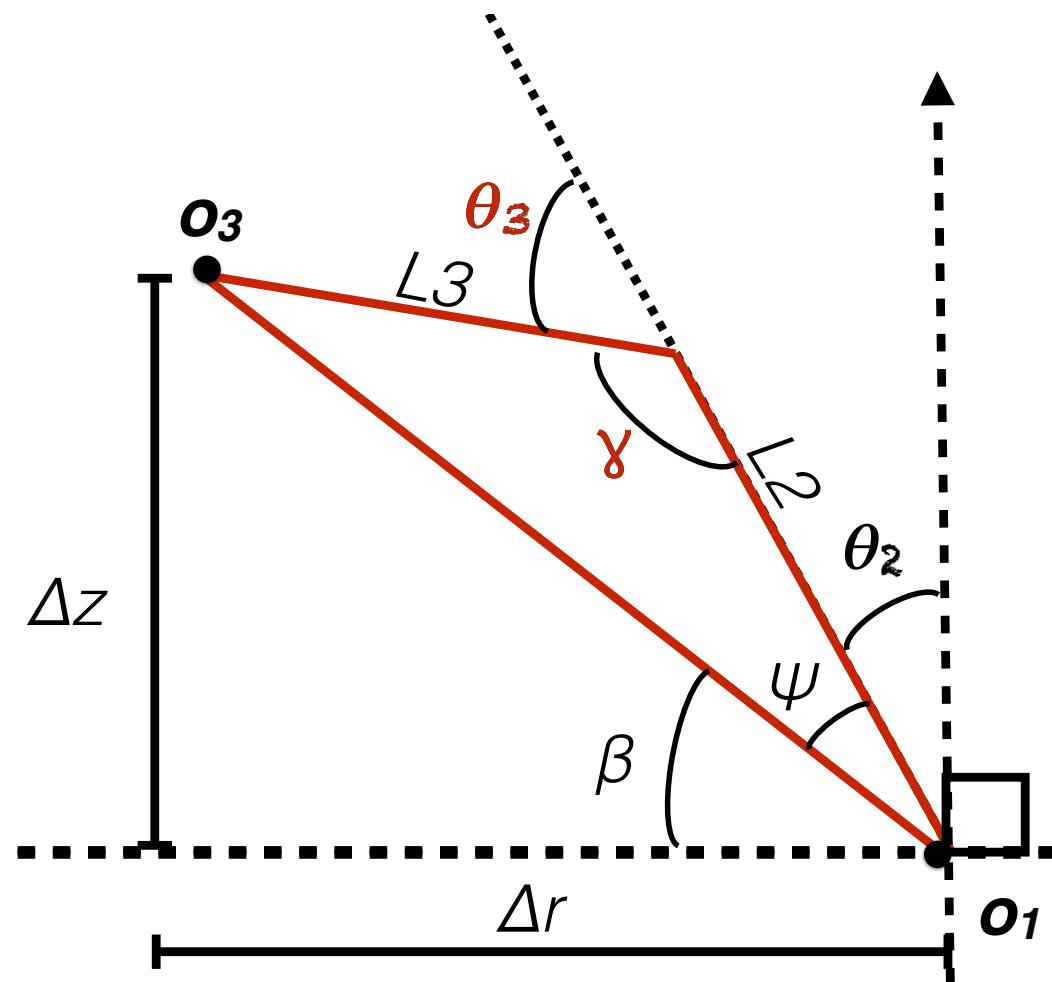
solve for θ_1

$$\Theta_1 = \text{atan2}(\gamma_g, x_g)$$

solve for θ_3

(Law of cosines with supplementary angle γ)

$$\cos \Theta_3 = \frac{\Delta z^2 + \Delta r^2 - L_2^2 - L_3^2}{2 L_2 L_3}$$



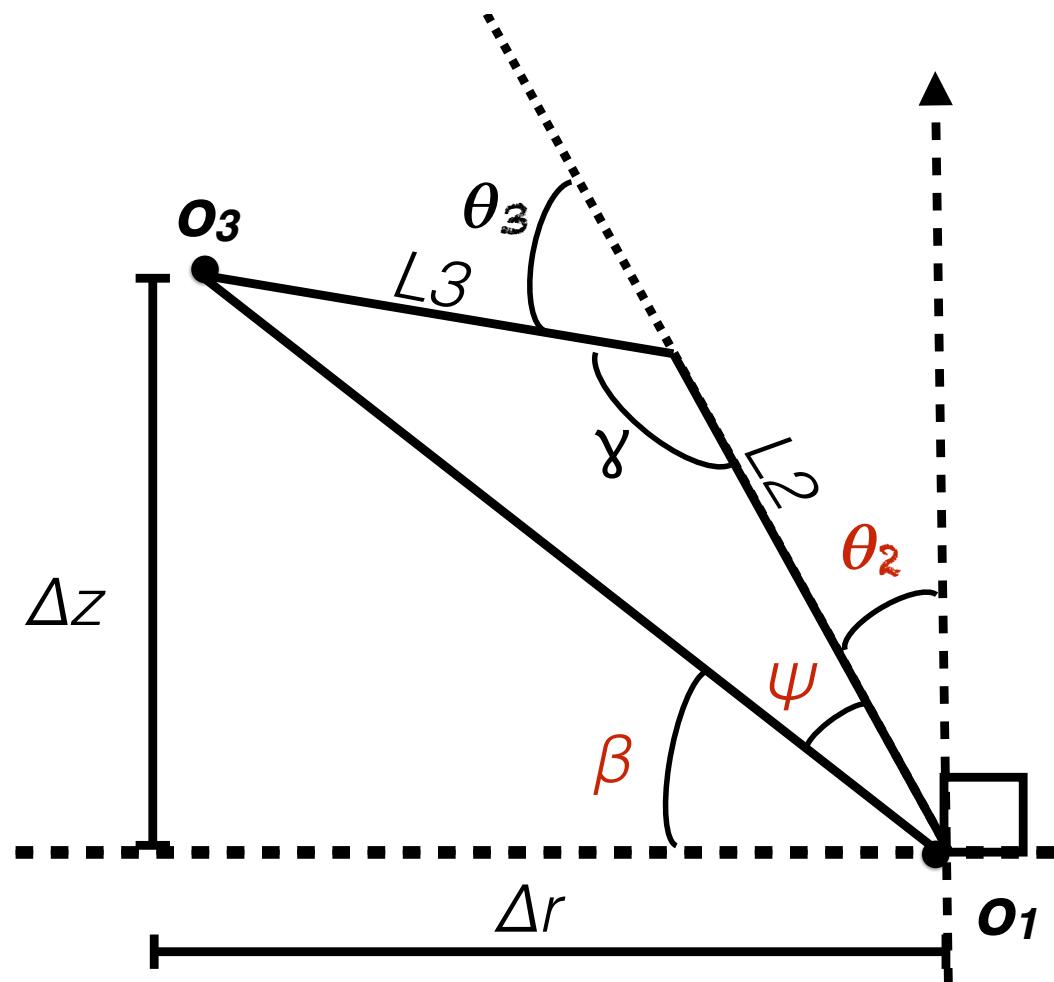
solve for θ_1

$$\Theta_1 = \text{atan2}(\gamma_g, x_g)$$

solve for θ_3

$$\cos \Theta_3 = \frac{\Delta z^2 + \Delta r^2 - L_2^2 - L_3^2}{2 L_2 L_3}$$

solve for θ_2



solve for θ_1

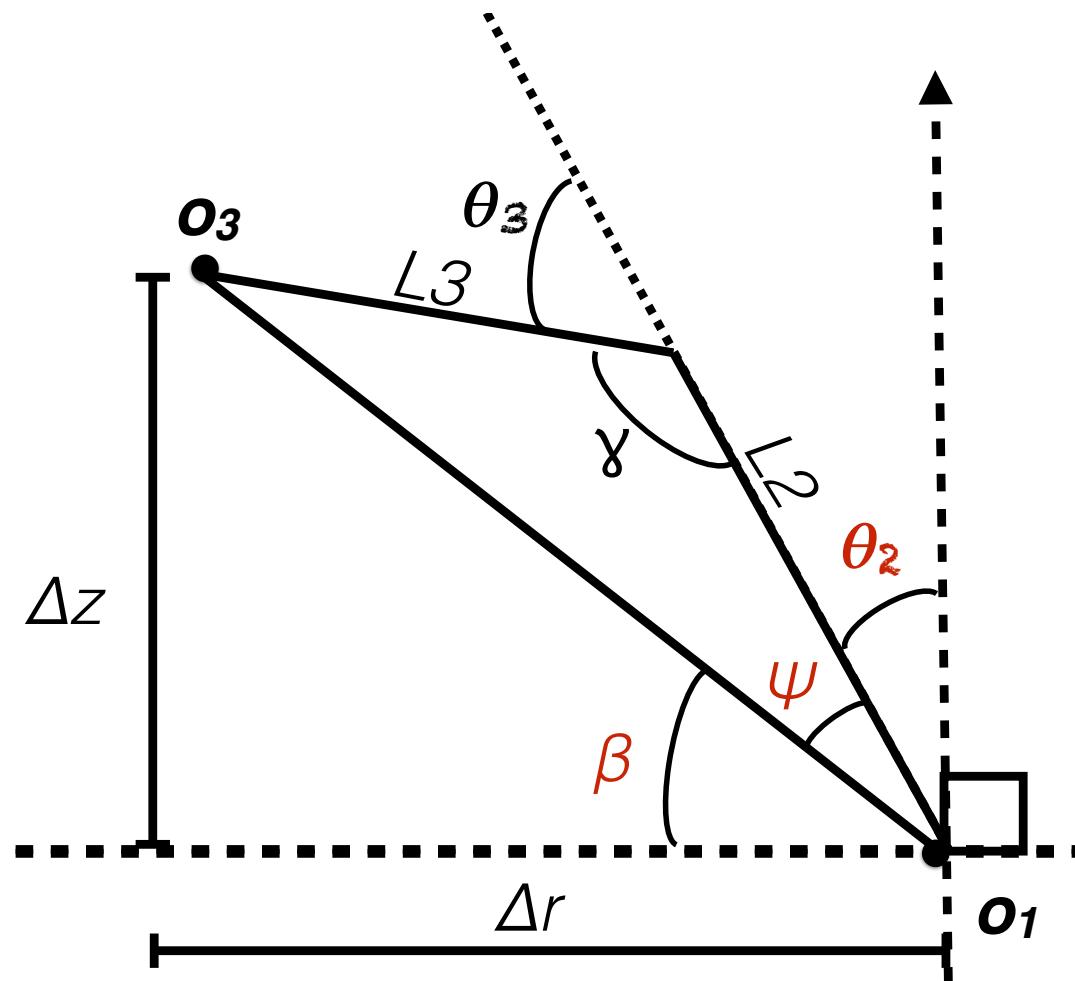
$$\theta_1 = \text{atan2}(\gamma_g, x_g)$$

solve for θ_3

$$\cos \theta_3 = \frac{\Delta z^2 + \Delta r^2 - L_2^2 - L_3^2}{2 L_2 L_3}$$

solve for θ_2

(Law of cosines with angle ψ ,
arctan with angle β)



solve for θ_1

$$\Theta_1 = \text{atan2}(\gamma_g, x_g)$$

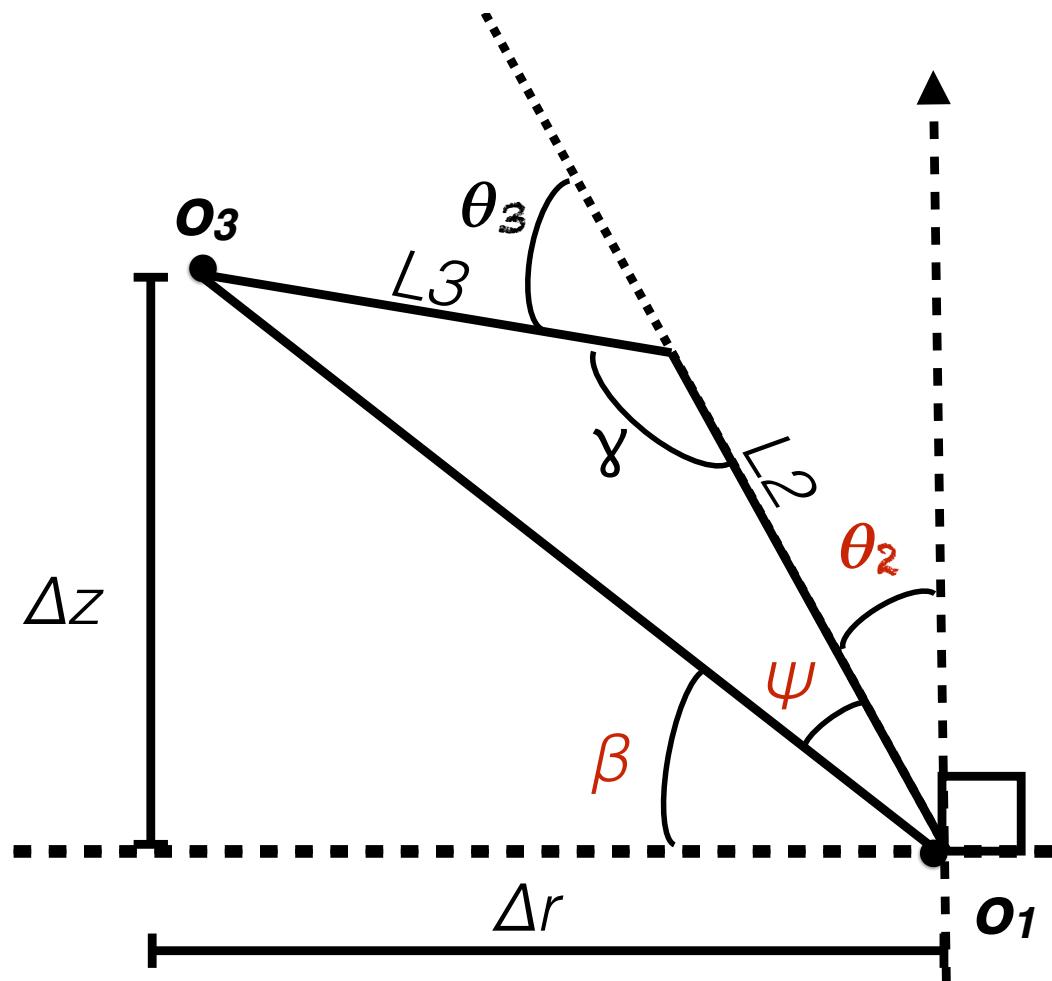
solve for θ_3

$$\cos \Theta_3 = \frac{\Delta z^2 + \Delta r^2 - L_2^2 - L_3^2}{2 L_2 L_3}$$

solve for θ_2

$$\Theta_2 = \begin{cases} \frac{\pi}{2} - \beta - \psi & \text{if } \Theta_3 \geq 0 \quad \text{"Elbow up"} \\ \frac{\pi}{2} - \beta + \psi & \text{if } \Theta_3 < 0 \quad \text{"Elbow-down"} \end{cases}$$

two potential solutions
depending on elbow angle



solve for θ_1

$$\theta_1 = \text{atan2}(y_g, x_g)$$

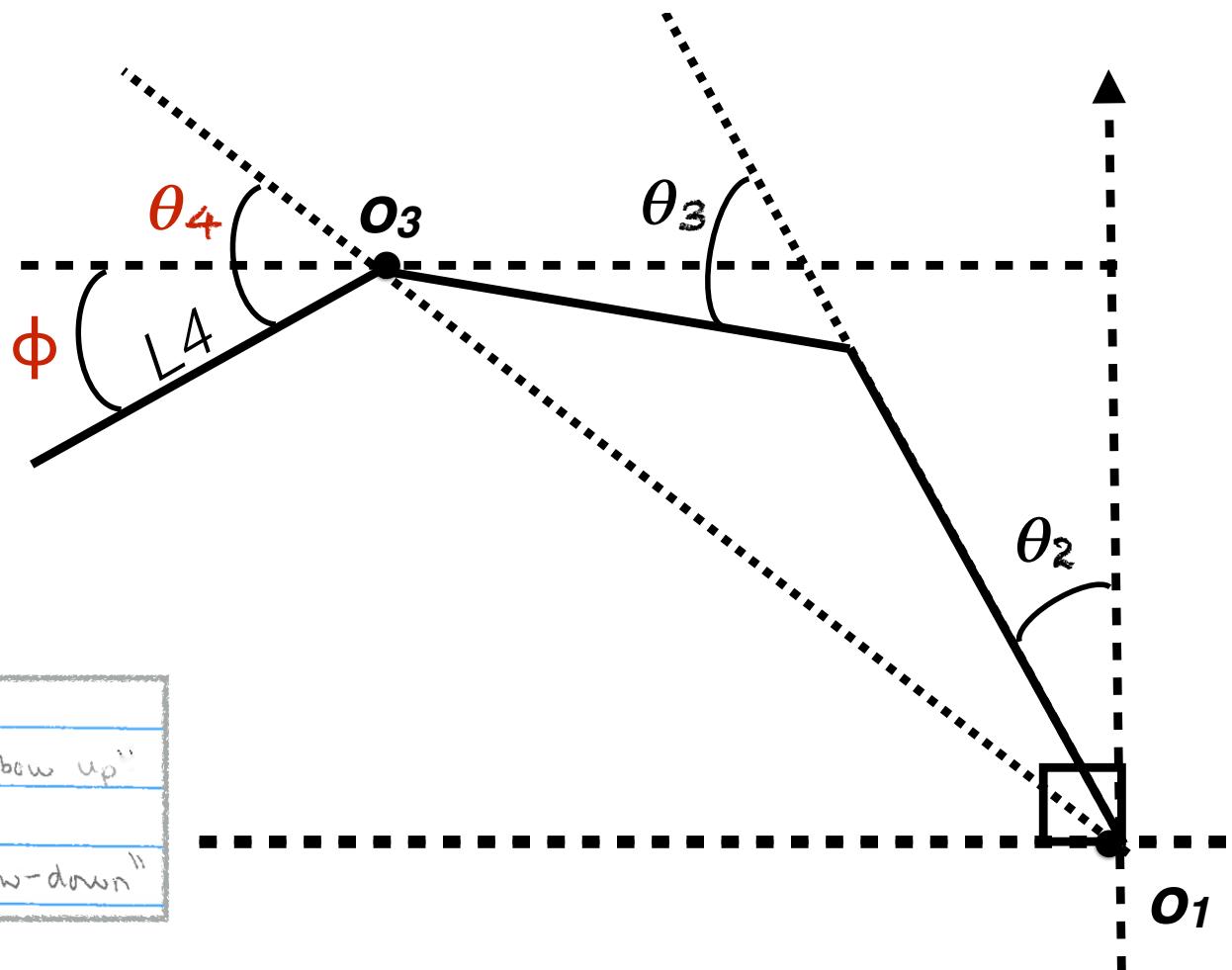
solve for θ_3

$$\cos \theta_3 = \frac{\Delta z^2 + \Delta r^2 - L_2^2 - L_3^2}{2 L_2 L_3}$$

solve for θ_2

$$\theta_2 = \begin{cases} \frac{\pi}{2} - \beta - \psi & \text{if } \theta_3 \geq 0 \quad \text{"Elbow up"} \\ \frac{\pi}{2} - \beta + \psi & \text{if } \theta_3 < 0 \quad \text{"Elbow-down"} \end{cases}$$

solve for θ_4



solve for θ_1

$$\theta_1 = \text{atan2}(y_g, x_g)$$

solve for θ_3

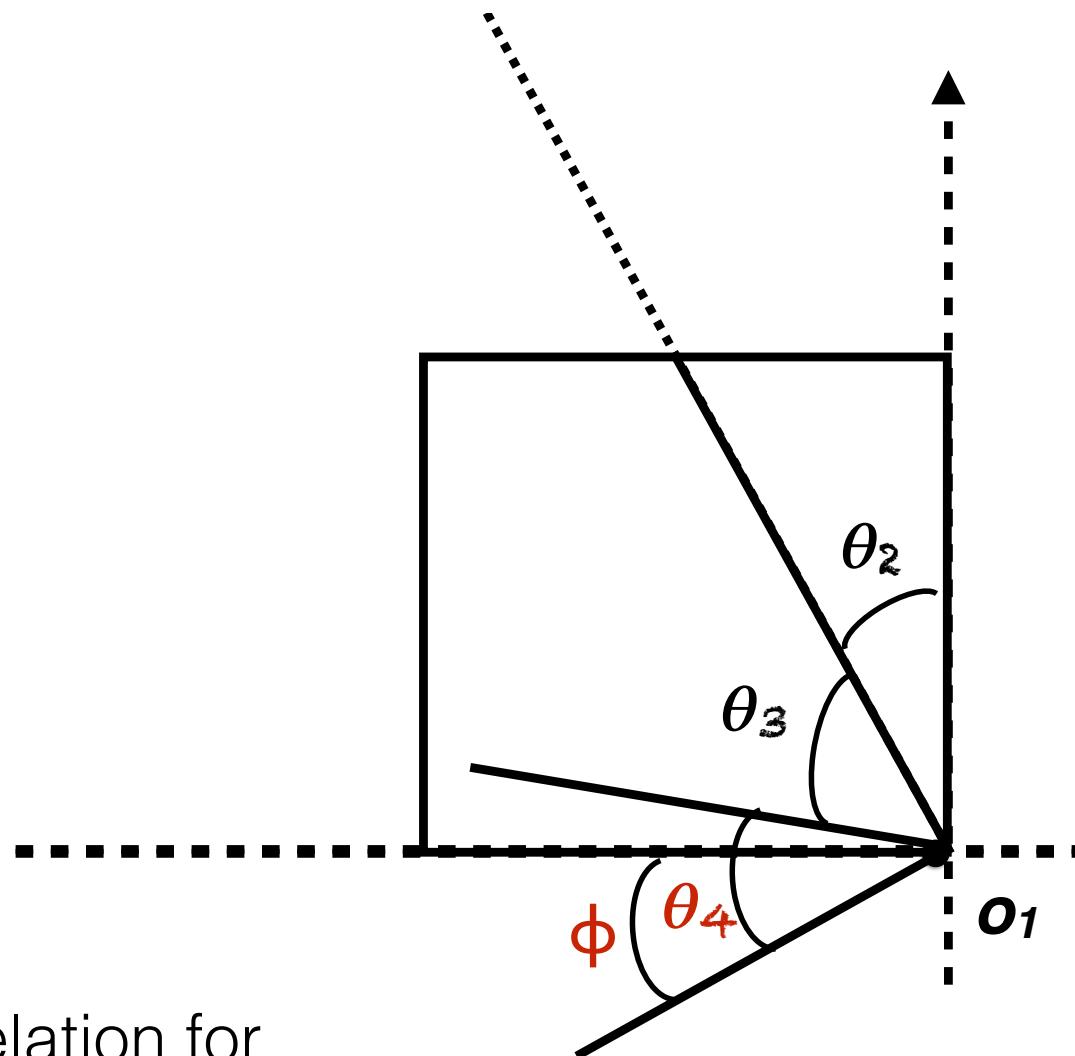
$$\cos \theta_3 = \frac{\Delta z^2 + \Delta r^2 - L_2^2 - L_3^2}{2 L_2 L_3}$$

solve for θ_2

$$\theta_2 = \begin{cases} \frac{\pi}{2} - \beta - \psi & \text{if } \theta_3 \geq 0 \quad \text{"Elbow up"} \\ \frac{\pi}{2} - \beta + \psi & \text{if } \theta_3 < 0 \quad \text{"Elbow-down"} \end{cases}$$

solve for θ_4

(Equivalence relation for
adding angles from \mathbf{z}_0)



solve for θ_1

$$\theta_1 = \text{atan2}(\gamma_g, x_g)$$

solve for θ_3

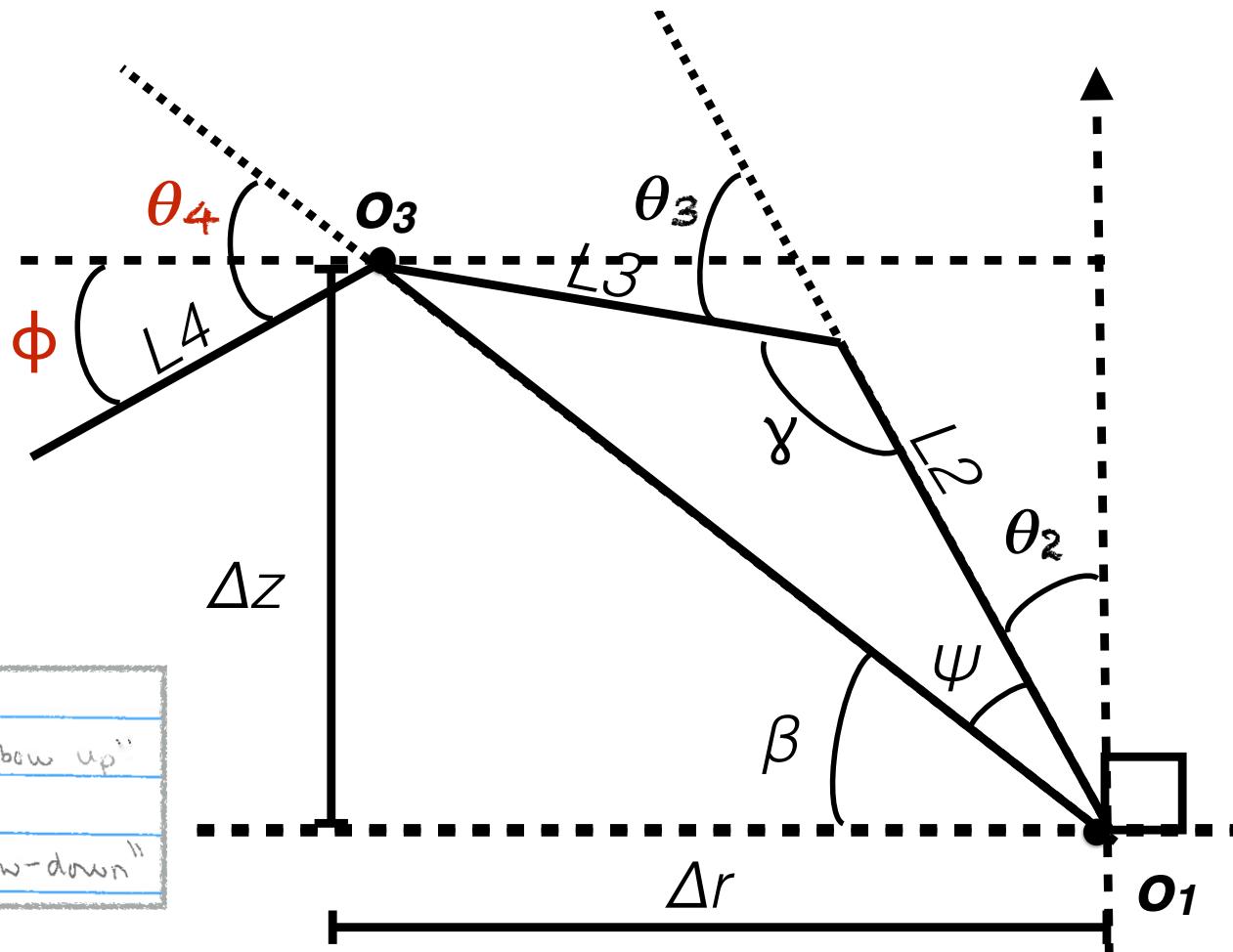
$$\cos \theta_3 = \frac{\Delta z^2 + \Delta r^2 - L_2^2 - L_3^2}{2 L_2 L_3}$$

solve for θ_2

$$\theta_2 = \begin{cases} \frac{\pi}{2} - \beta - \psi & \text{if } \theta_3 \geq 0 \quad \text{"Elbow up"} \\ \frac{\pi}{2} - \beta + \psi & \text{if } \theta_3 < 0 \quad \text{"Elbow-down"} \end{cases}$$

solve for θ_4

$$\theta_4 = \phi - \theta_2 - \theta_3 + \frac{\pi}{2}$$



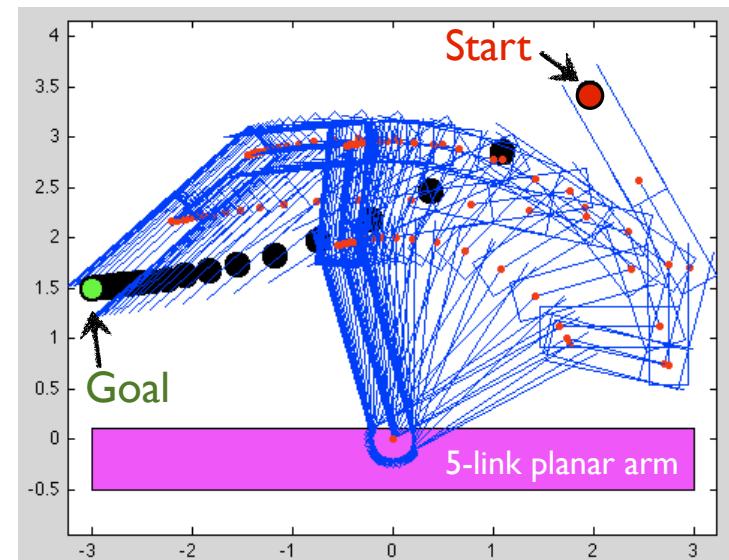
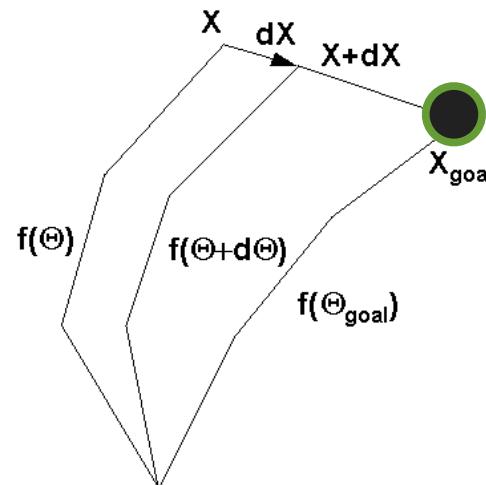
(Addition of angles in arm plane starting from z_0)

Why Closed Form?

- Advantages
 - Speed: IK solution computed in constant time
 - Predictability: consistency in selecting satisfying IK solution
- Disadvantage
 - Generality: general form for arbitrary kinematics difficult to express

Iterative Solutions to IK

- Minimize error between current endeffector and its desired position
- Transform desired endeffector velocity into configuration space
- Repeatedly step to convergence at desired endeffector position



Next Class

- IK as an optimization problem
 - Gradient descent optimization
 - Manipulator Jacobian as the derivative of configuration
- Advanced: IK by Cyclic Coordinate Descent

autorob.github.io

A close-up photograph of a humanoid robot's arm and torso. The robot has a gold-colored metallic finish on its shoulder, elbow, and hand, which is wearing a white glove. Its torso is white with gold accents on the shoulders and a small blue light on the chest. The background is a solid dark blue.

Inverse Kinematics: Manipulator Jacobian

UM EECS 398/598 - autorob.github.io