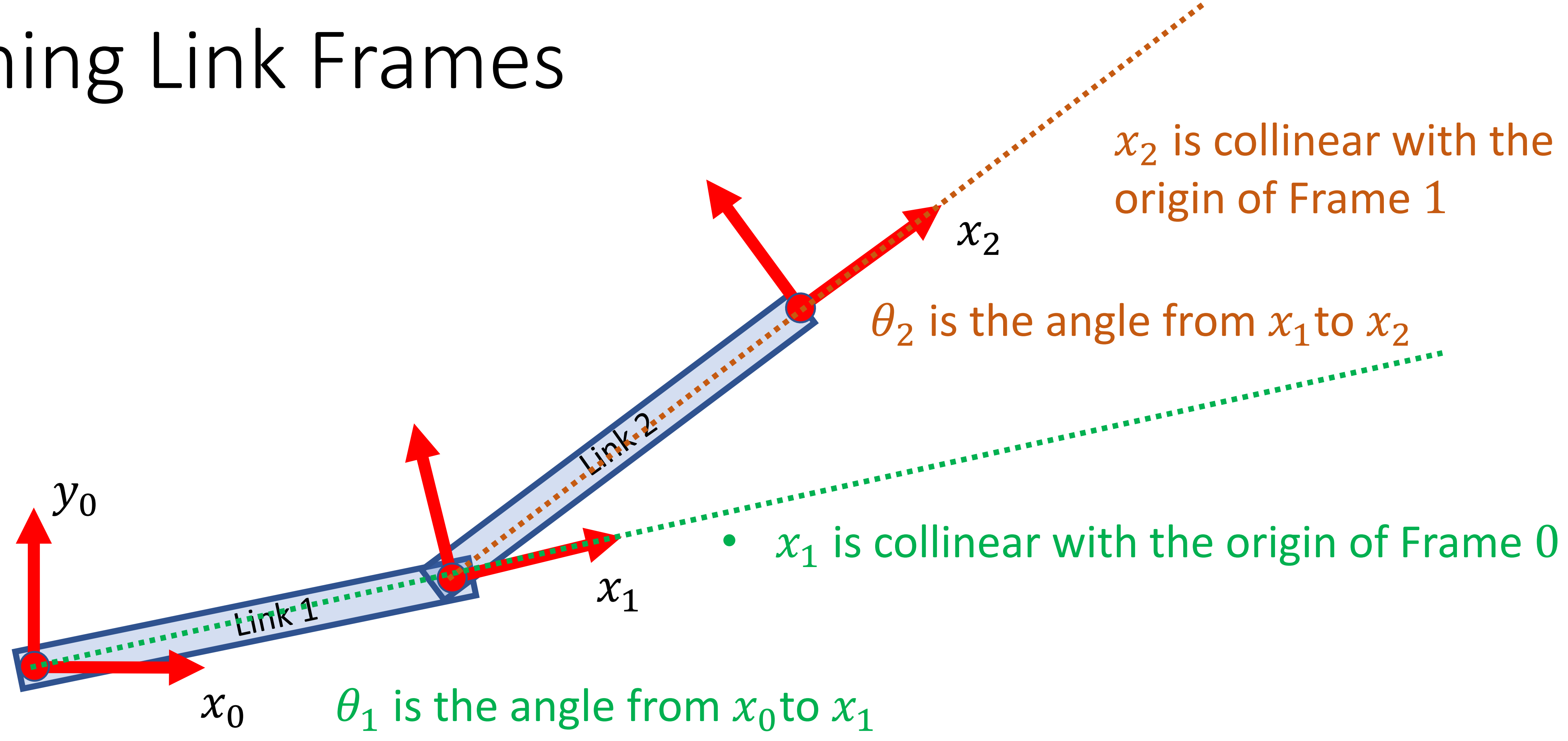
CS 3630

Inverse Kinematics:

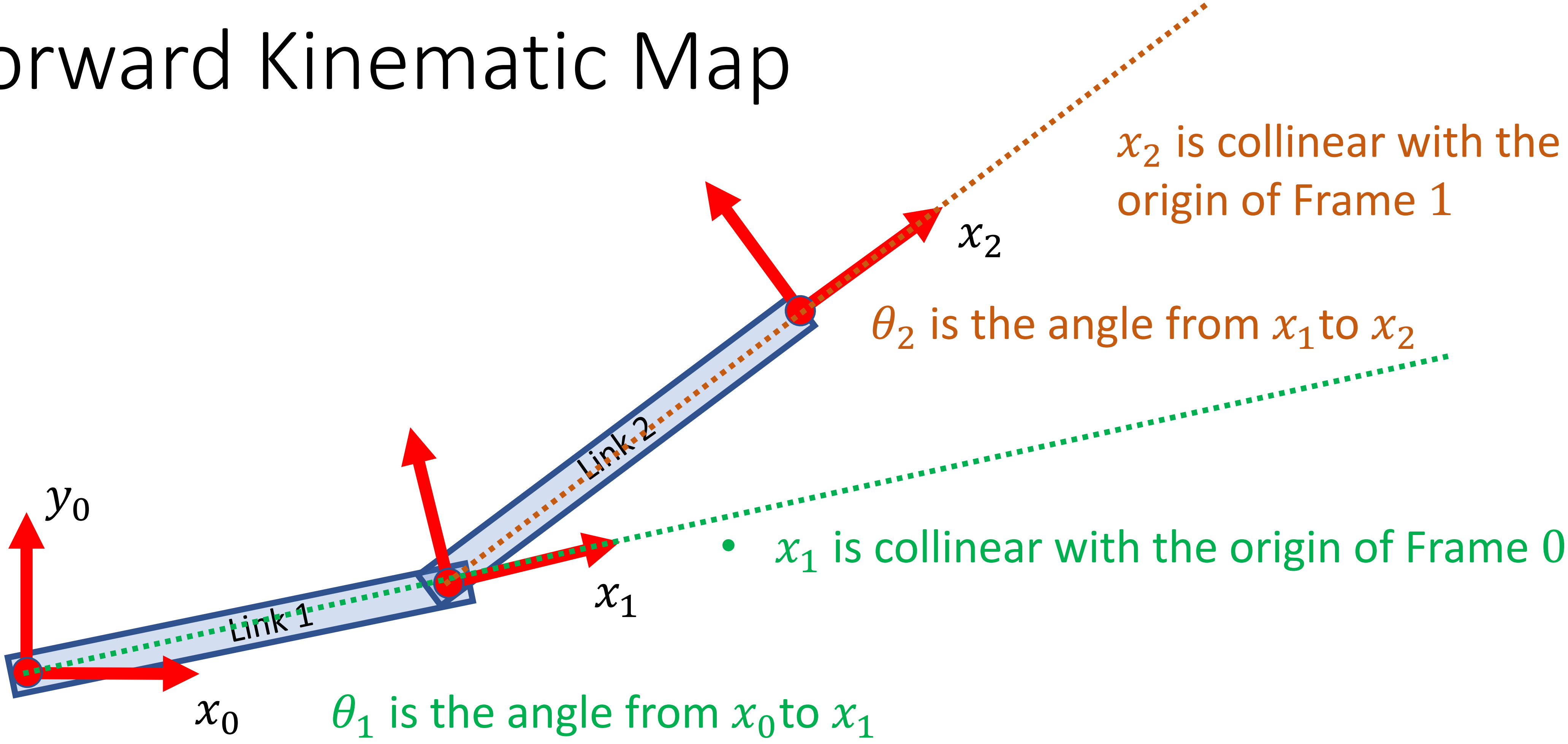
Planar Arms

Assigning Link Frames



- Frame n is the end-effector frame. It can be attached to link n in any manner that is convenient.
- In this case, $n = 2$, so Frame 2 is the end-effector frame.

The Forward Kinematic Map

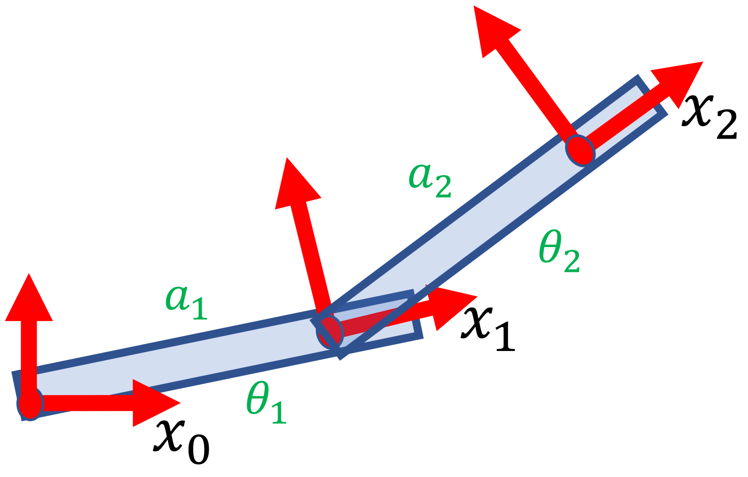


Once we have coordinate frames for each link:

- Determine T_i^{i-1} for adjacent links as a function of q_i
- The forward kinematic map is given by: $T_n^0(q_1 \dots q_n) = T_1^0(q_1) \dots T_n^{n-1}(q_n)$

The Forward Kinematic Map

- The forward kinematic map gives the position and orientation of the end-effector frame as a function of the joint variables:



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

- For the two-link planar arm, we have

$$T_2^0 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & a_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & a_1\sin\theta_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & a_2\sin\theta_2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & a_1\cos\theta_1 + a_2\cos(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & a_1\sin\theta_1 + a_2\sin(\theta_1+\theta_2) \\ 0 & 0 & 1 \end{bmatrix}$$

Inverse Kinematics

The General Inverse Kinematics Problem:

Given the forward kinematic map: $T_n^0 = F(q_1, \dots, q_n)$

Solve for $q_1, \dots, q_n$ to achieve a desired T^d

i.e., find $q_1^d, \dots, q_n^d$ such that $F(q_1^d, \dots, q_n^d) = T^d$

Why is this difficult?

- In general, $F(q_1, \dots, q_n)$ will be nonlinear. Solving nonlinear equations is hard.
- Further, for a general $F(q_1, \dots, q_n)$ we don't know
  - Does a solution to $F(q_1, \dots, q_n) = \mathrm{T}^{\mathrm{d}}$ exist?
  - If a solution exists, is it unique?

The Inverse Kinematic Solution

For the two-link arm, typically the goal is to place the end-effector at a desired location.

- Denote the coordinates of the origin of frame 2 by $o_{2,x}, o_{2,y}$.
- Solve for θ_1 and θ_2 such that

$$o_{2,x} = a_1 \cos\theta_1 + a_2 \cos(\theta_1 + \theta_2)$$
$$o_{2,y} = a_1 \sin\theta_1 + a_2 \sin(\theta_1 + \theta_2)$$

- Recall that a_1 and a_2 are constants defined by the mechanical structure of the arm.
- This is a nonlinear set of equations in θ_1 and θ_2 --- and nonlinear equations can be very difficult to solve!

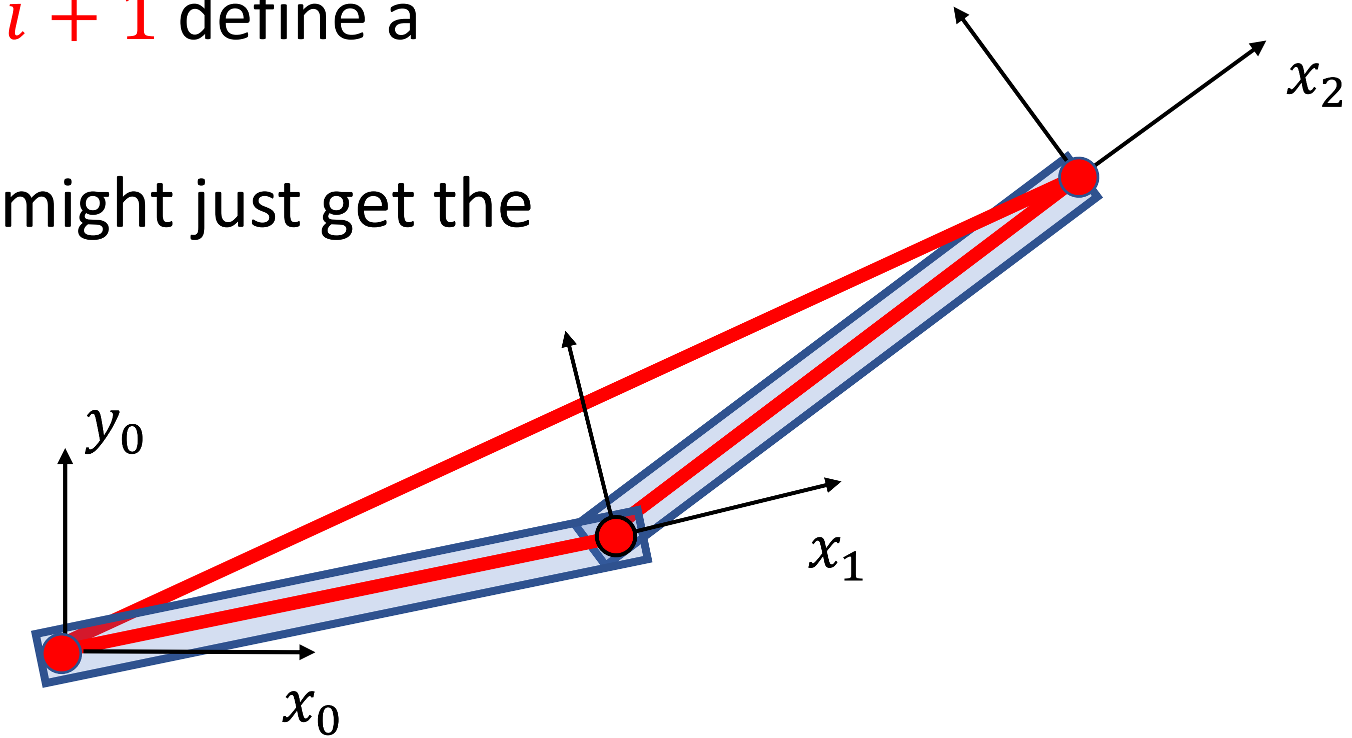
$$T_2^0 = \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & a_1\cos\theta_1 + a_2\cos(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & a_1\sin\theta_1 + a_2\sin(\theta_1+\theta_2) \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} * & * & o_{2,x} \\ * & * & o_{2,y} \\ 0 & 0 & 1 \end{bmatrix}$$

NOTE:

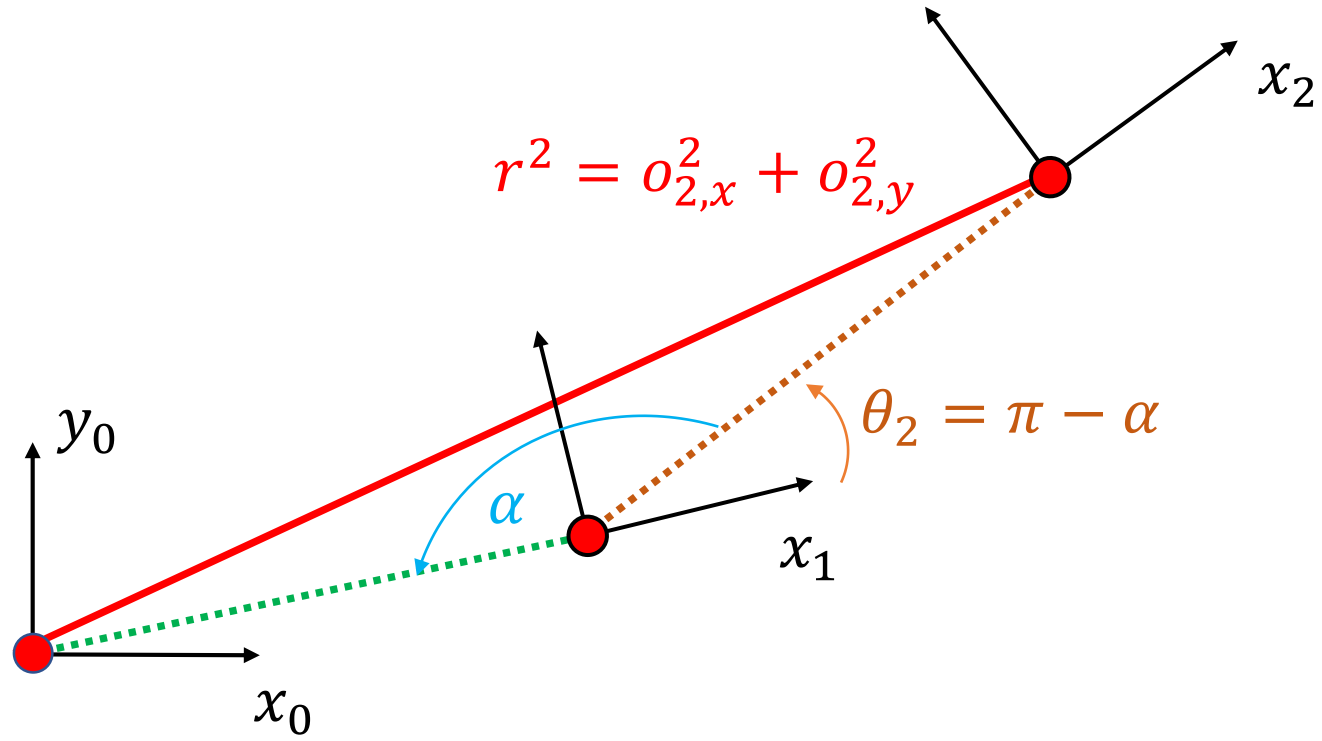
- We don’t care about the orientation for this problem.
- In fact, we can’t choose the orientation if we also choose $o_{2,x}, o_{2,y}$.

Geometric Methods (closed-form solns)

- For some manipulators, it is possible to use fairly simple trigonometry to solve the inverse kinematics problem.
- Any two adjacent links are coplanar (any two intersecting lines are coplanar).
- The origins of frames $i-1$, i, and $i+1$ define a triangle.
- Simple, trigonometry in the plane might just get the job done!



Solving for θ_2



Denote the coordinates of the origin of frame 2 by $o_{2,x}, o_{2,y}$.

The Law of Cosines:

$$r^2 = a_1^2 + a_2^2 - 2a_1a_2\cos\alpha$$

Define:

$$D \stackrel{\text{def}}{=} \frac{a_1^2 + a_2^2 - r^2}{2a_1a_2} = \cos\alpha$$

Then $\sin\alpha = \pm\sqrt{1 - D^2}$

Finally,

$$\alpha = \tan^{-1}\frac{\pm\sqrt{1 - D^2}}{D}$$

- With this set of equations, we can solve for θ_2 using simple solutions to closed-form equations.
- ***We never need to solve a nonlinear system!***

What about existence and uniqueness?

Does a solution always exist for $\alpha = \tan^{-1}\frac{\pm\sqrt{1-D^2}}{D}$?

No solution exists if $D^2 > 1$:

$$D^2 = \left(\frac{a_1^2 + a_2^2 - r^2}{2a_1a_2}\right)^2 \leq 1$$

$$a_1^2 + a_2^2 - r^2 \leq \pm 2a_1a_2$$

$$a_1^2 \pm 2a_1a_2 + a_2^2 \leq r^2$$

$$(a_1 \pm a_2)^2 \leq r^2$$

$$|\,\boldsymbol{a_1} \pm \boldsymbol{a_2}| \leq \boldsymbol{r}$$

In this case, a solution exists!

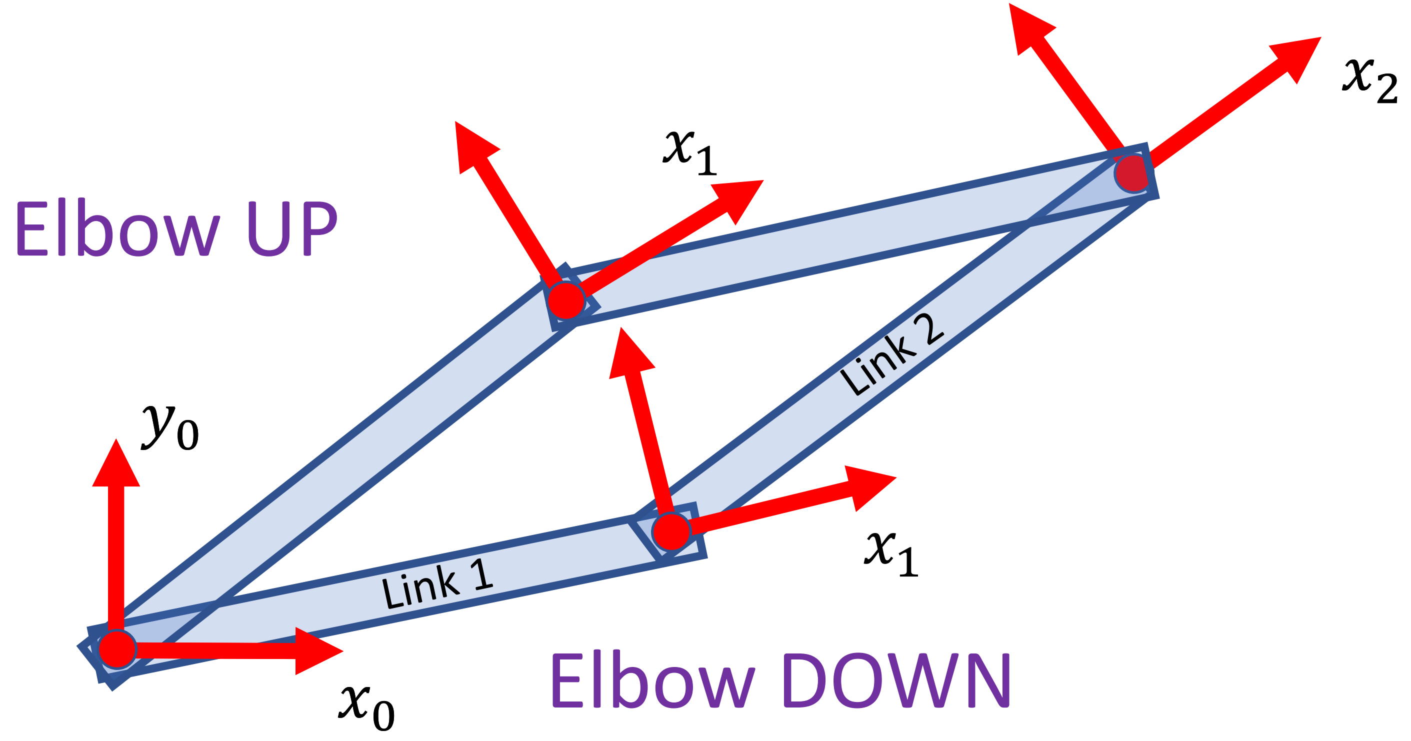
What about existence and uniqueness?

Is the solution unique for $\alpha = \tan^{-1} \frac{\pm\sqrt{1-D^2}}{D}$?

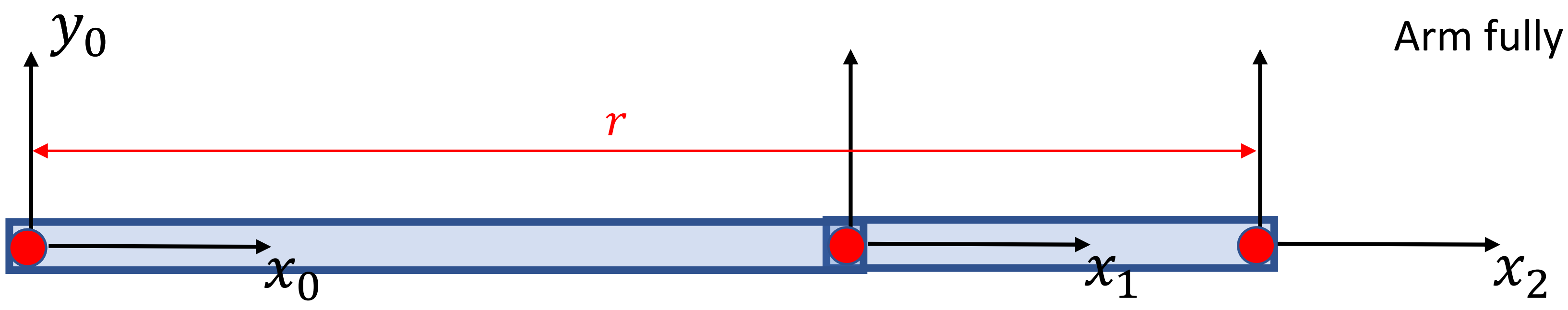
Clearly, the solution is not unique, since we may choose either square root!

The second solution uses $\alpha = \tan^{-1} \frac{-\sqrt{1-D^2}}{D}$ which results in an "elbow UP" configuration.

NOTE: when $| a_1 \pm a_2 |= r$ the two solutions "collapse" into a single solution.

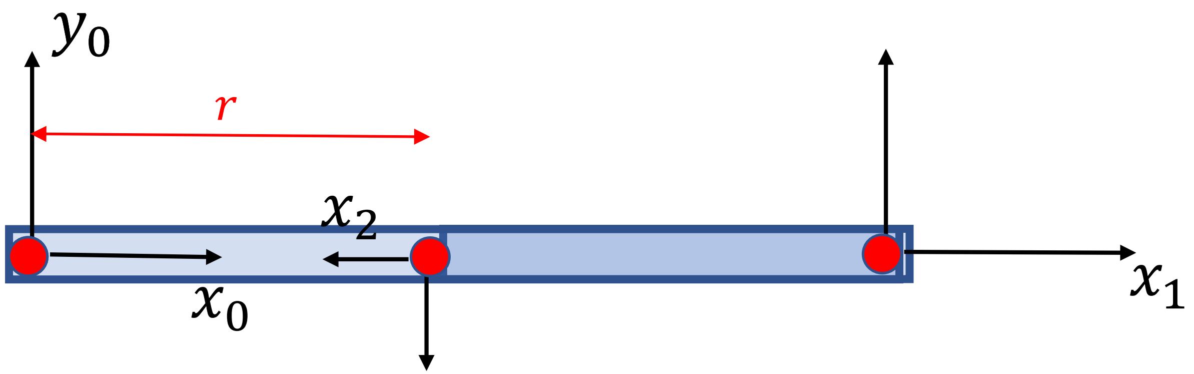


Degenerate Solutions



Arm fully extended

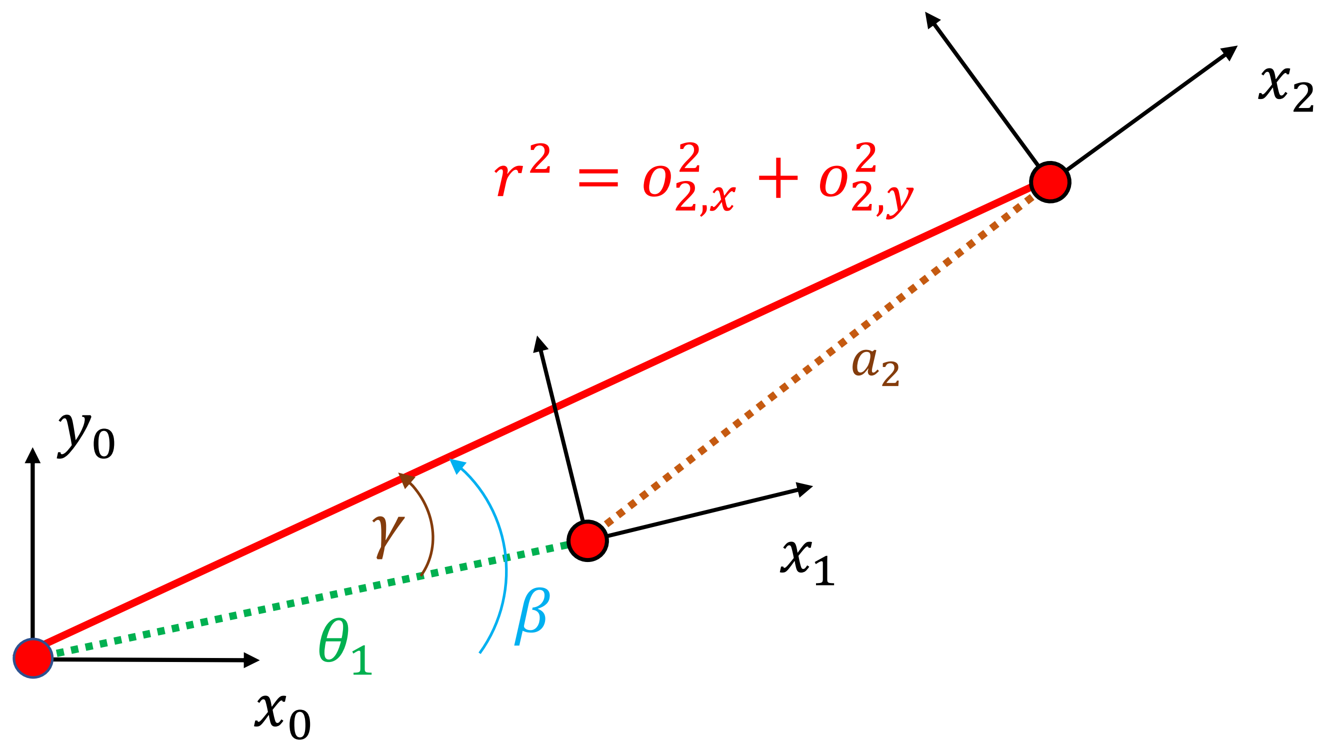
$$a_1 + a_2 = r$$



Arm “folds back” on itself

$$a_1 - a_2 = r$$

Solving for θ_1



Denote the coordinates of the origin of frame 2 by $o_{2,x}, o_{2,y}$.

$$\theta_1 = \beta - \gamma$$

$$\beta = \tan^{-1} \frac{o_{2,y}}{o_{2,x}}$$

The Law of Cosines, this time for γ:

$$a_2^2 = a_1^2 + r^2 - 2a_1 r \cos \gamma$$

- With this set of equations, we can solve for θ_1 using simple solutions to closed-form equations.
- ***We never need to solve a nonlinear system!***

<u>Elbow up is left as an exercise for you!</u>

Position and Orientation

Suppose we wish to position the end effector frame at a specific position, ***and*** with a specific orientation.

- ***We can parameterize the end effector frame by*** $(\boldsymbol{X_e}, \boldsymbol{Y_e}, \boldsymbol{\phi})$
  - $(\boldsymbol{X_e}, \boldsymbol{Y_e})$ ***give the position of the origin of the frame***
  - $\boldsymbol{\phi}$ ***gives the orientation of the frame:***

$$T_2^0 = \begin{bmatrix} \cos\phi & -\sin\phi & X_e \\ \sin\phi & \cos\phi & Y_e \\ 0 & 0 & 1 \end{bmatrix}$$

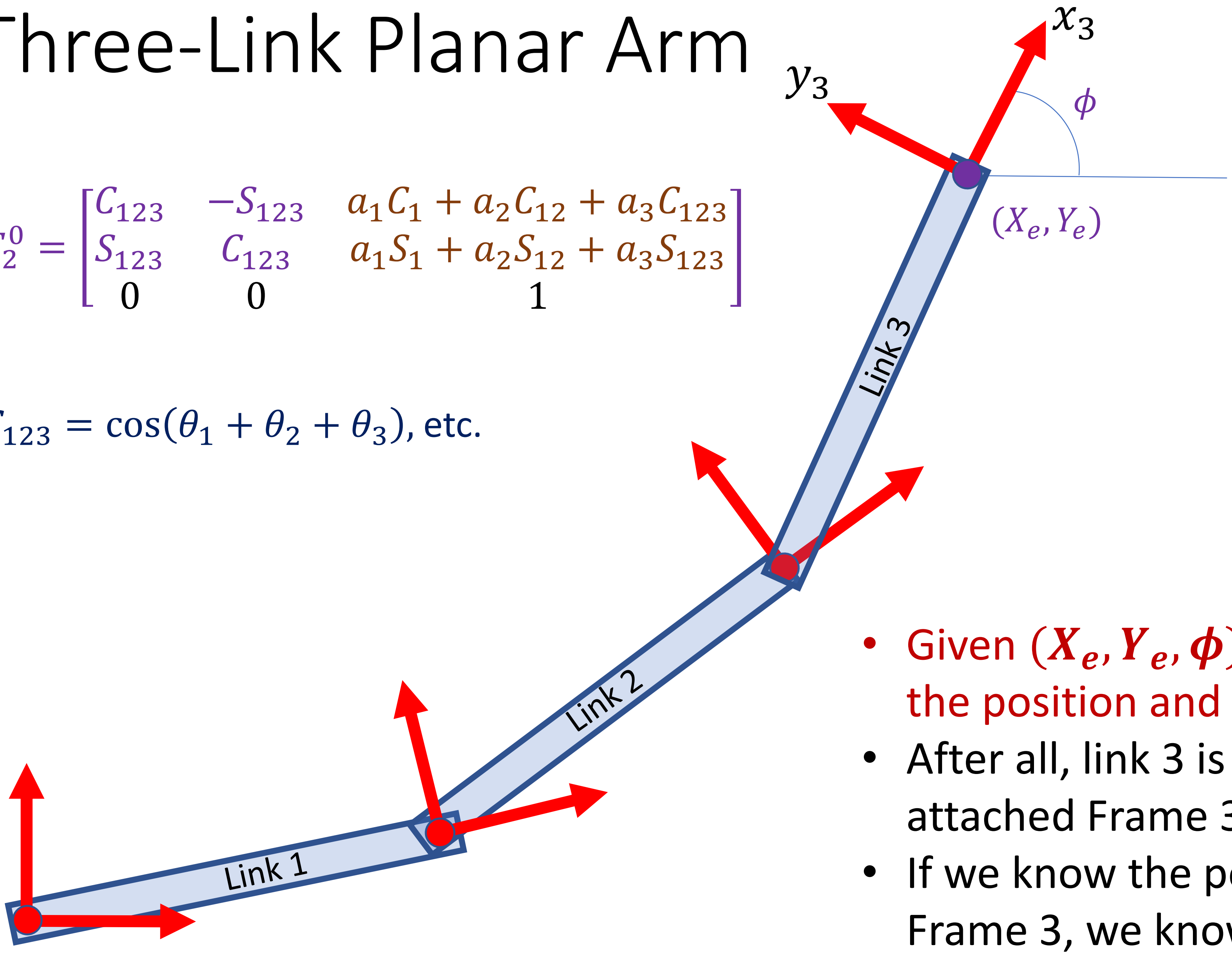
- We can't do this with a two-link arm.
  - Intuitively, we have three inputs (θ_1 and θ_2) and three outputs.
  - Our solution to the two-link arm shows that once we choose θ_1 and θ_2 the orientation of the end-effector frame is fully determined.

➢ ***Add another link!***

Three-Link Planar Arm

$$T_2^0 = \begin{bmatrix} C_{123} & -S_{123} & a_1C_1 + a_2C_{12} + a_3C_{123} \\ S_{123} & C_{123} & a_1S_1 + a_2S_{12} + a_3S_{123} \\ 0 & 0 & 1 \end{bmatrix}$$

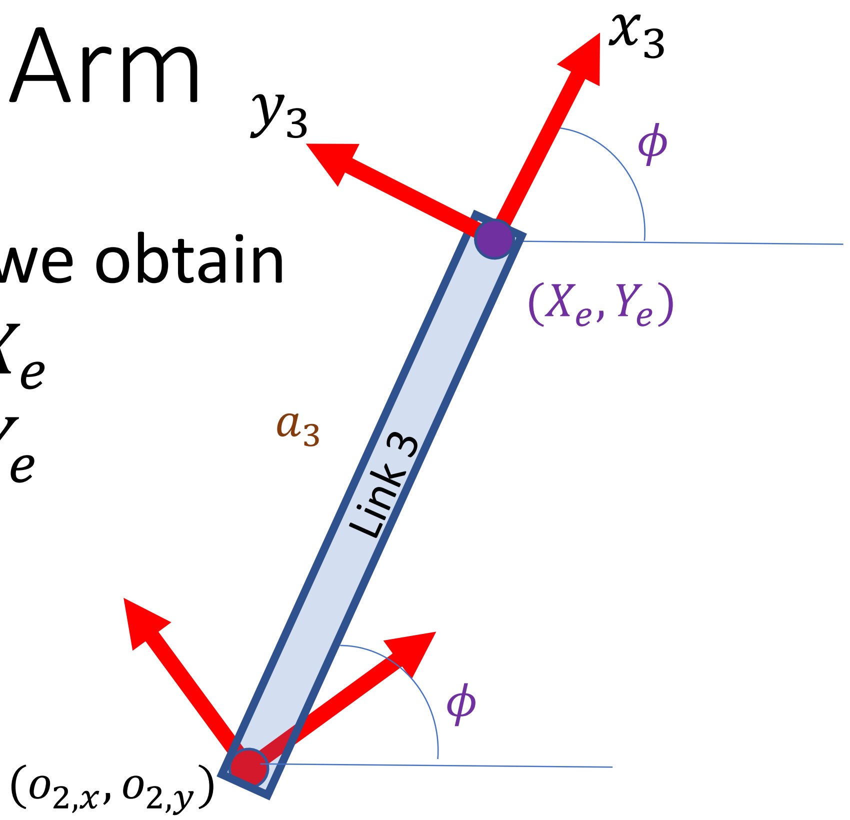
$C_{123} = \cos(\theta_1 + \theta_2 + \theta_3)$, etc.



$$T_3^0 = \begin{bmatrix} \cos\phi & -\sin\phi & X_e \\ \sin\phi & \cos\phi & Y_e \\ 0 & 0 & 1 \end{bmatrix}$$

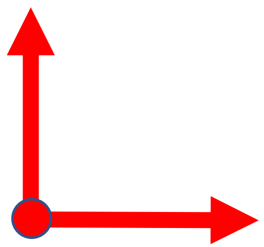
- Given $(\boldsymbol{X_e}, \boldsymbol{Y_e}, \boldsymbol{\phi})$, we know everything about the position and orientation of link 3!
- After all, link 3 is a rigid body, with a rigidly attached Frame 3.
- If we know the position and orientation of Frame 3, we know everything about Link 3!

Three-Link Planar Arm



Using simple trigonometry we obtain

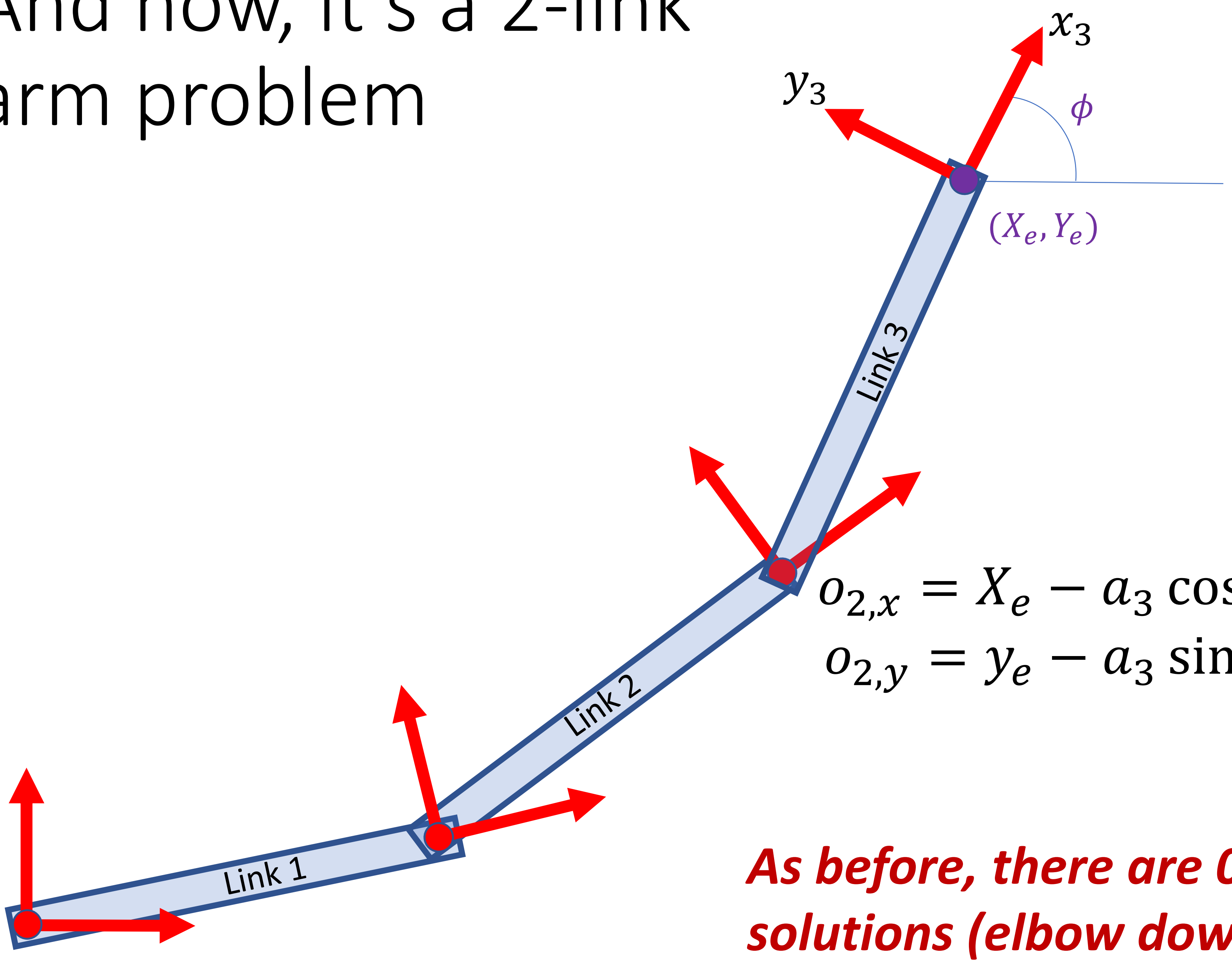
$$o_{2,x} + a_3 \cos\phi = X_e$$
$$o_{2,y} + a_3 \sin\phi = Y_e$$



And from this we immediately conclude

$$o_{2,x} = X_e - a_3 \cos\phi$$
$$o_{2,y} = y_e - a_3 \sin\phi$$

And now, it's a 2-link arm problem



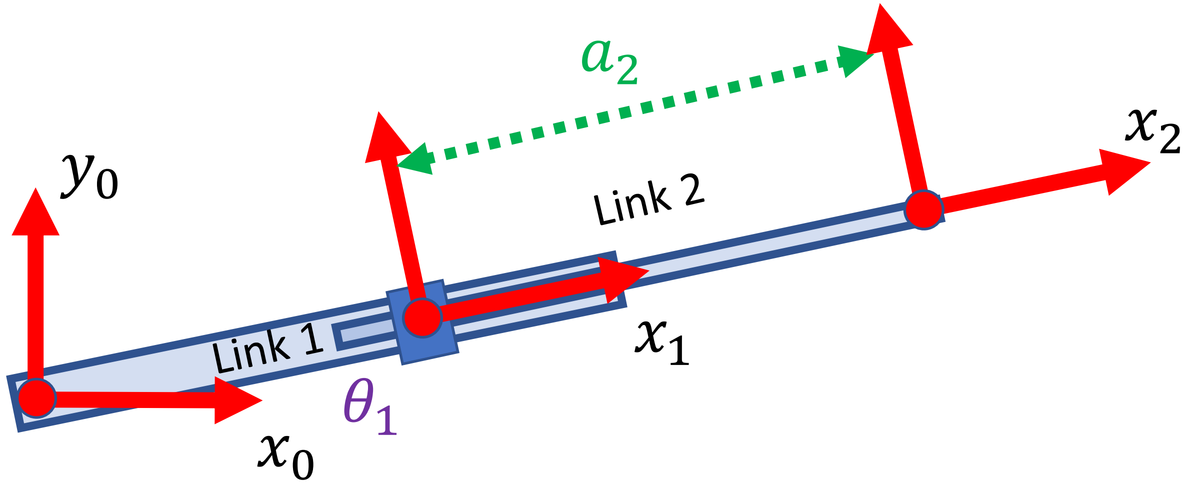
$$o_{2,x} = X_e - a_3 \cos\phi$$
$$o_{2,y} = y_e - a_3 \sin\phi$$

As before, there are 0, 1, or two solutions (elbow down is shown).

Prismatic Joint no link offset

Joint 2 is prismatic.

- Define Frames 0 and 1 as before: x_1 is collinear with the origin of Frame 0.
- Define Frame 2 such that x_2 is collinear with x_1



$\theta_2 = 0$ since x_1 and x_2 axes are parallel

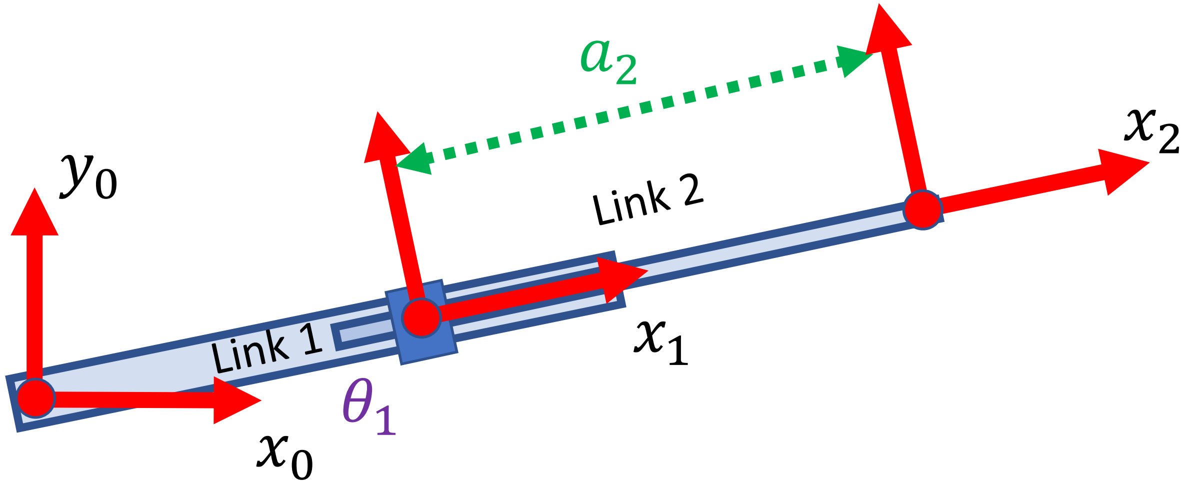
$$T_2^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & a_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} 1 & 0 & a_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Prismatic Joints

Joint 2 is prismatic.

- Define Frames 0 and 1 as before: x_1 is collinear with the origin of Frame 0.
- Define Frame 2 such that x_2 is collinear with x_1



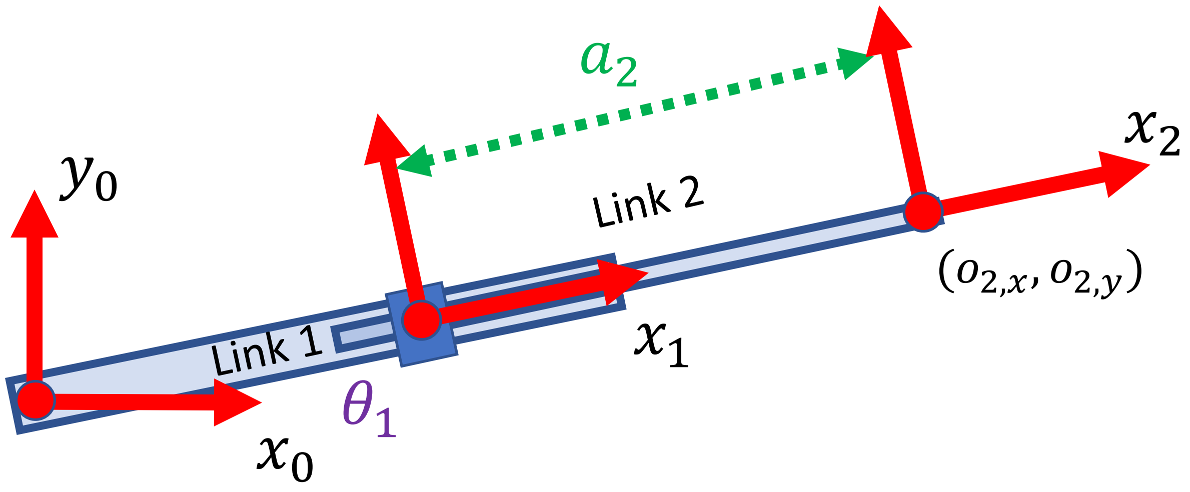
$$T_2^0 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & a_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & a_1\sin\theta_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & a_i \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & (a_1+a_2)\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & (a_1+a_2)\sin\theta_1 \\ 0 & 0 & 1 \end{bmatrix}$$

Inverse Kinematic Solution

As before, let $(o_{2,x}, o_{2,y})$ denote the coordinates of the origin of Frame 2.

Since $\theta_2 = 0$, the orientation of the end-effector frame is completely determined by θ_1:

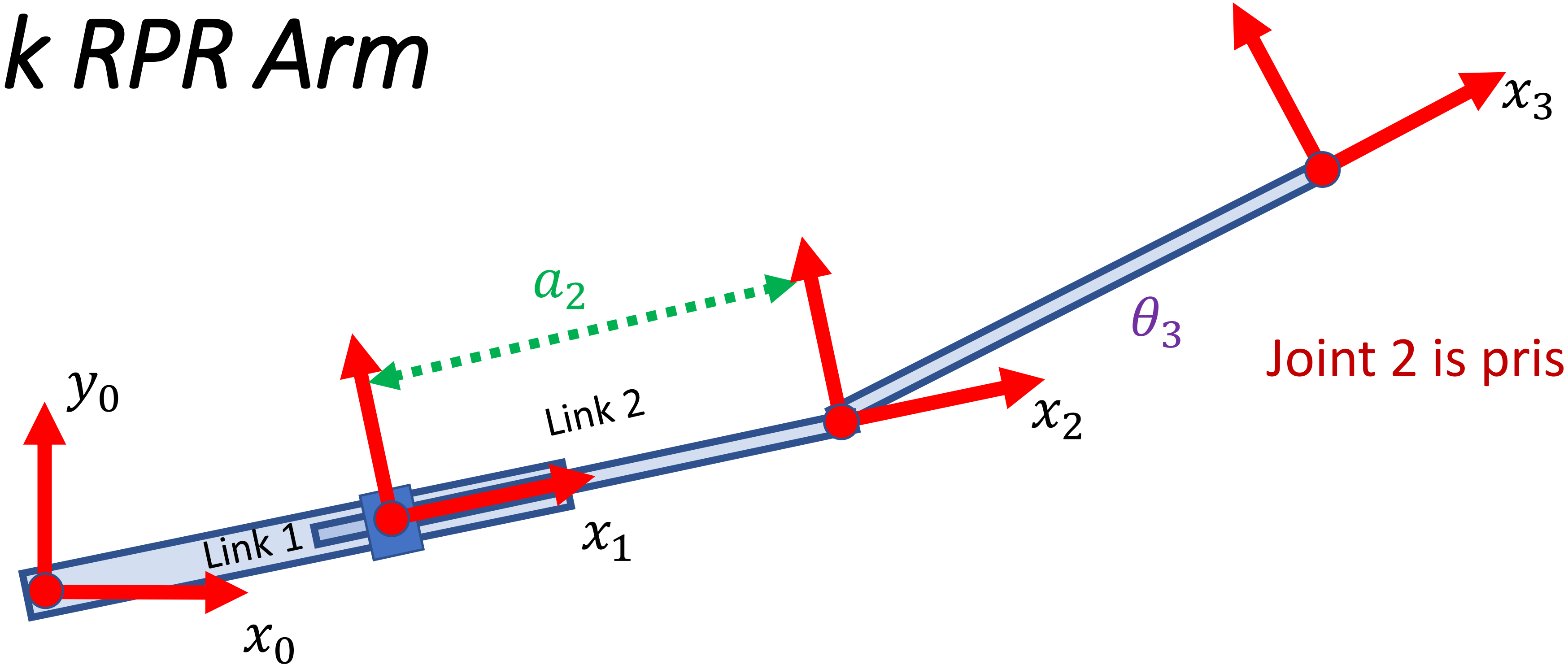
$$\theta_1 = \tan^{-1}\frac{o_{2,y}}{o_{2,x}}$$



Once we have a solution for θ_1, we can directly solve the forward kinematic equations for a_2:

$$(a_1+a_2)\cos\theta_1 = o_{2,x} \Rightarrow a_2 = \frac{o_{2,x} - a_1\cos\theta_1}{\cos\theta_1}$$

Three-link RPR Arm

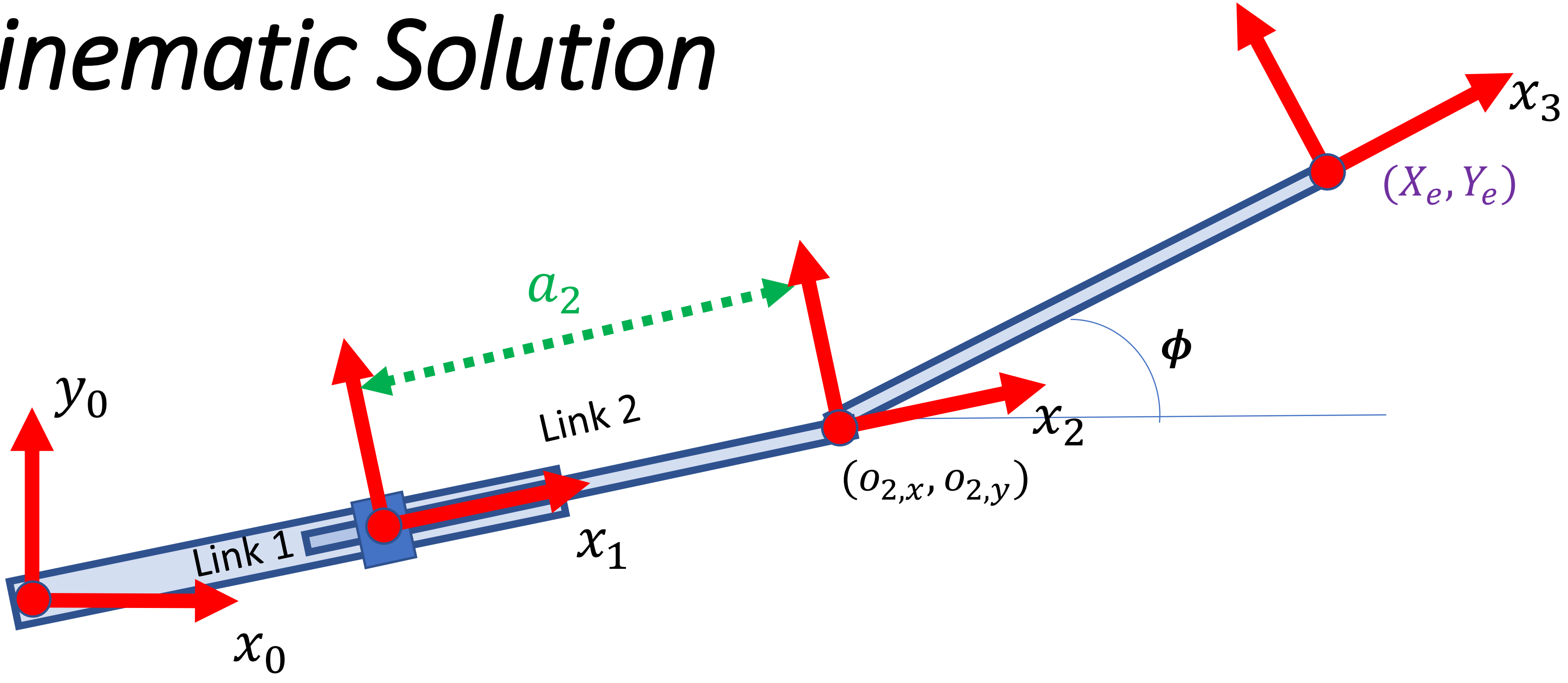


Joint 2 is prismatic.

$$T_3^0 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & a_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & a_1\sin\theta_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & a_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & a_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & a_3\sin\theta_3 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta_1+\theta_3) & -\sin(\theta_1+\theta_3) & (a_1+a_2)\cos\theta_1 + a_3\cos(\theta_1+\theta_3) \\ \sin(\theta_1+\theta_3) & \cos(\theta_1+\theta_3) & (a_1+a_2)\sin\theta_1 + a_3\sin(\theta_1+\theta_3) \\ 0 & 0 & 1 \end{bmatrix}$$

Inverse Kinematic Solution



As with the three-link RRR arm,

- ***Parameterize the end effector frame by*** $(\boldsymbol{X_e}, \boldsymbol{Y_e}, \boldsymbol{\phi})$
- ***Use*** $\boldsymbol{\phi}$ ***and*** $\boldsymbol{a_3}$ ***to solve for*** $(\boldsymbol{o_{2,x}}, \boldsymbol{o_{2,y}})$
- ***Solve for*** $\boldsymbol{\theta_1}$ ***and*** $\boldsymbol{a_2}$ ***using the two-link RP solution given above.***
- $\boldsymbol{\theta_3 = \phi - \theta_1}$

Other Kinds of Robots

So far, we've looked only at simple planar arms with revolute joints.

Life becomes more complicated if

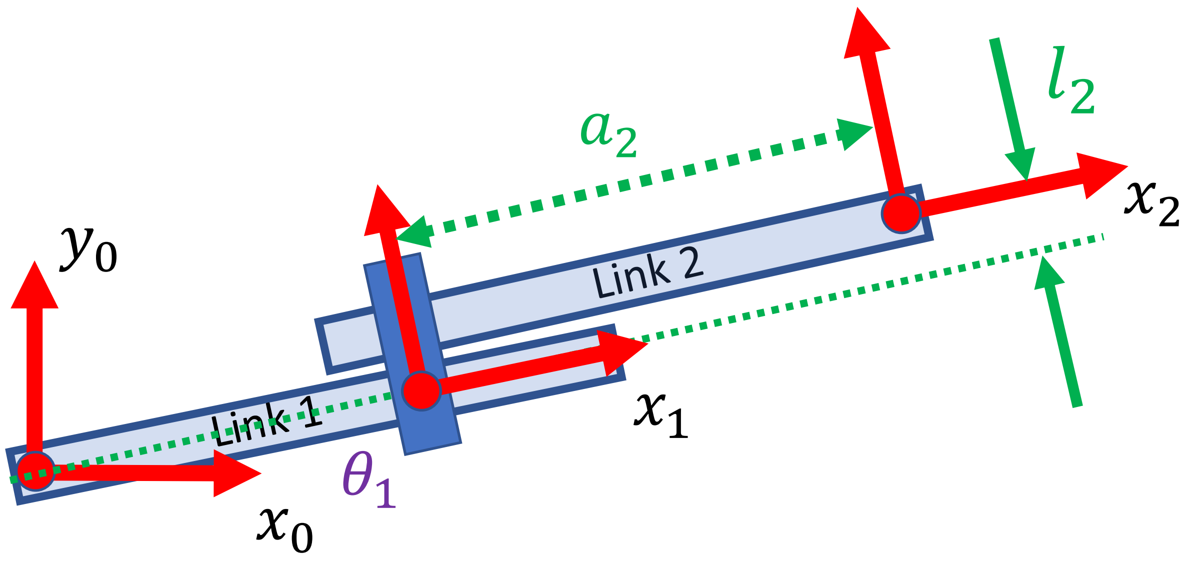
- We have prismatic joints with "link offsets"
- Robots are not planar (e.g., anthropomorphic arms)
- Robots have more joints than end-effector degrees of freedom (e.g., if a planar arm has four joint). Such robots are said to be *redundant*.

We'll need to find other ways to solve the inverse kinematics for such robots.

Let's see an example…

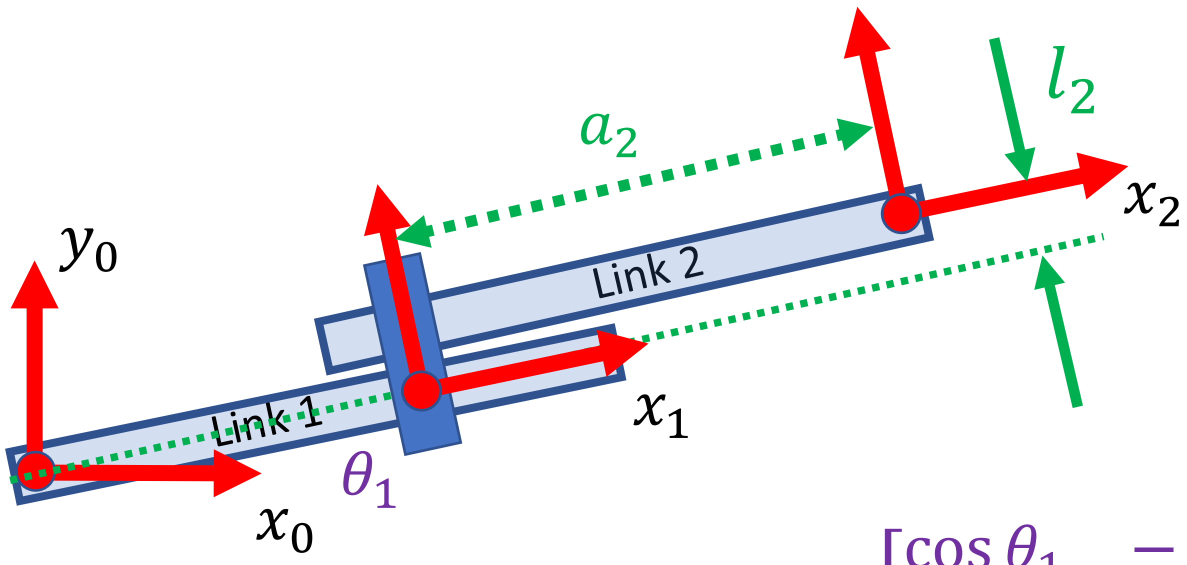
Prismatic Joint with link offset

Joint 2 is prismatic.



$$T_2^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & a_2 \\ 0 & 1 & l_2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a_2 \\ 0 & 1 & l_2 \\ 0 & 0 & 1 \end{bmatrix}$$

Prismatic Joint with link offset



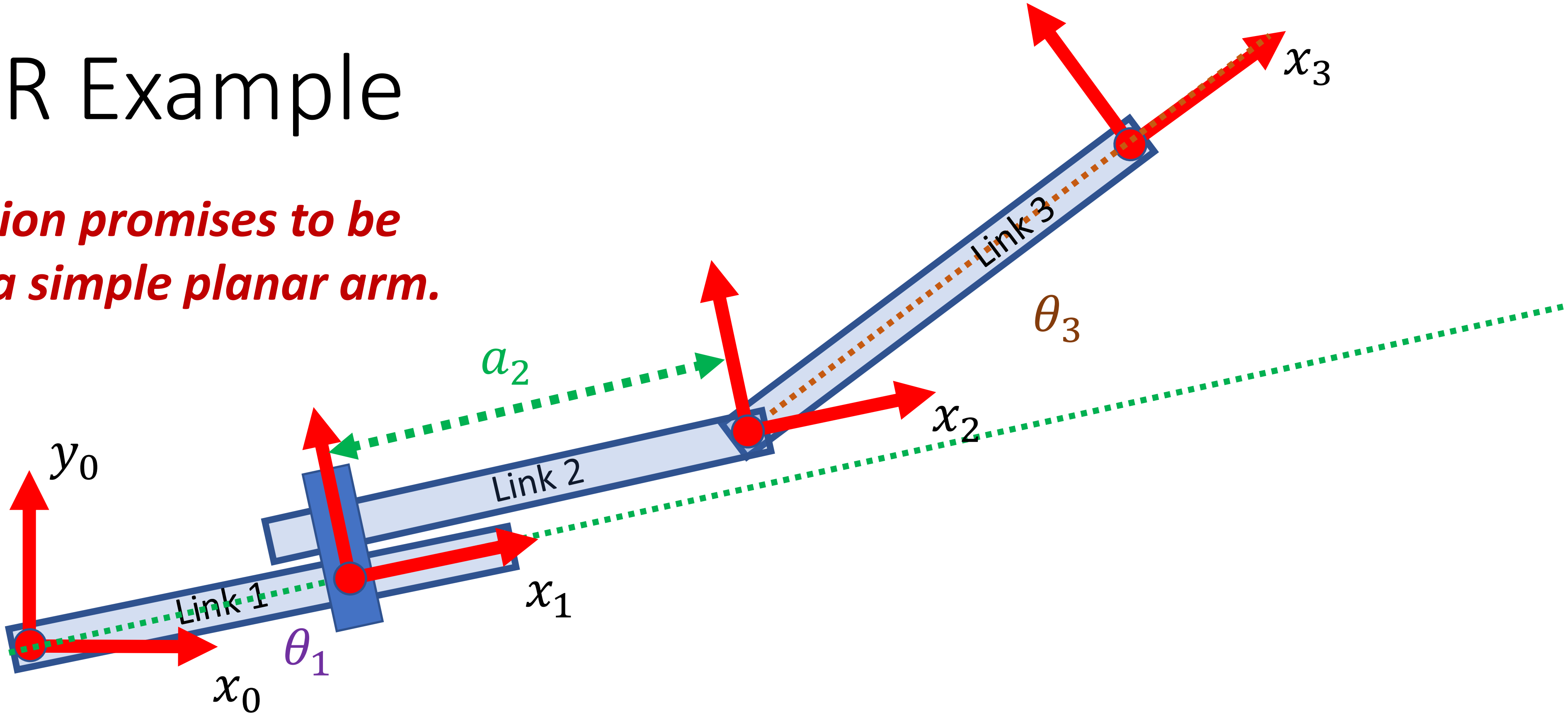
$$T_2^1 = \begin{bmatrix} 1 & 0 & a_2 \\ 0 & 1 & l_2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_2^0 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & a_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & a_1\sin\theta_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & a_2 \\ 0 & 1 & l_2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & (a_1+a_2)\cos\theta_1 - l_2\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 & (a_1+a_2)\sin\theta_1 + l_2\cos\theta_1 \\ 0 & 0 & 1 \end{bmatrix}$$

Three-link RPR Example

The geometric IKS solution promises to be painful, and this is still a simple planar arm.

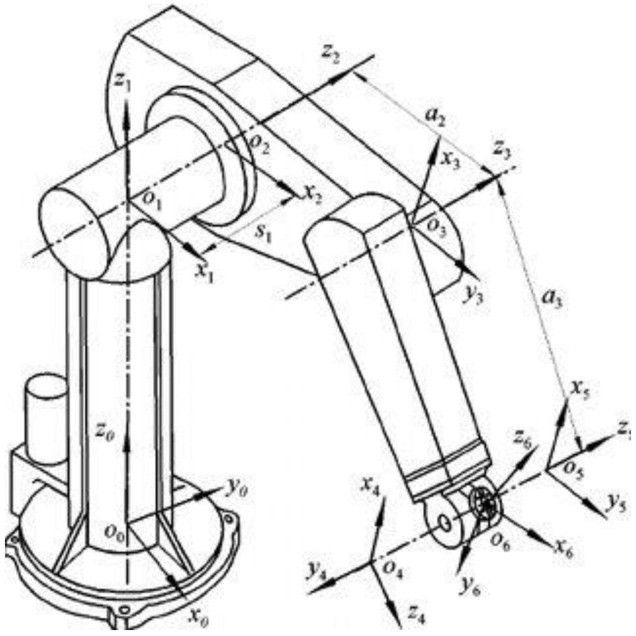


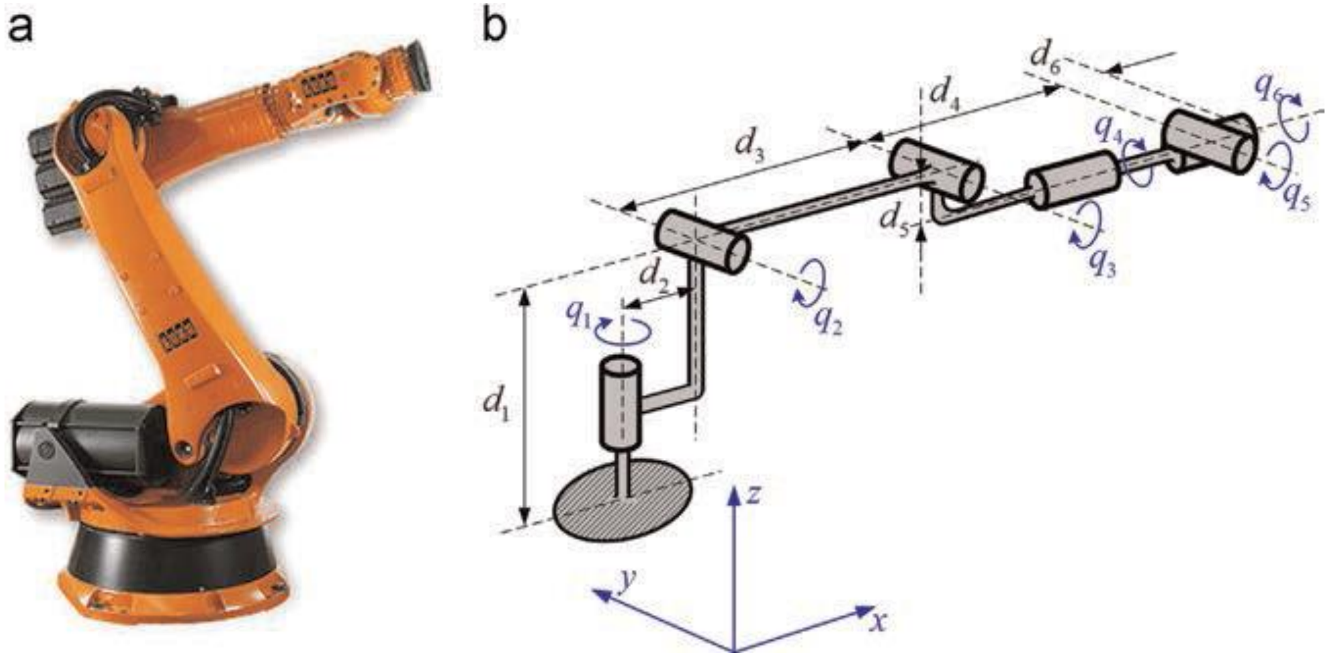
$$T_3^0 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & (a_1{+}a_2)\cos\theta_1 - l_2\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 & (a_1{+}a_2)\sin\theta_1 + l_2\cos\theta_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & a_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & a_3\sin\theta_3 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_{13} & -S_{13} & (a_1{+}a_2)\,C_1 - l_2 S_1 + a_3 C_{13} \\ S_{13} & C_{13} & (a_1{+}a_2)\,S_1 + l_2 C_1 + a_3 S_{13} \\ 0 & 0 & 1 \end{bmatrix}$$

More General Robot Arms

Imagine finding the geometric IKS solutions for more complicated arms in 3D work spaces…





Numerical Methods for Inverse Kinematics

- Denote the forward kinematic map by $F(q)$.
- $F(q)$ is a vector-valued function, each component gives one coordinate of the end-effector pose. If the end effector has m degrees of freedom and the robot has n joints, then

$$F(q_1, \ldots, q_n) = \begin{bmatrix} f_1(q) \\ \vdots \\ f_m(q) \end{bmatrix}$$

- For the two-link arm, we would have

$$F(\theta_1, \theta_2) = \begin{bmatrix} x(\theta_1, \theta_2) \\ y(\theta_1, \theta_2) \end{bmatrix}$$

- Denote by x^d the desired value of the end-effector pose.
- For the two-link arm we would have

$$x^d = \begin{bmatrix} o_{2,x}^d \\ \\ o_{2,y}^d \end{bmatrix}$$

- ***The inverse kinematic solution is the vector $\mathbf{q}^d$ such that $F(\mathbf{q}^d) = x^d$.***

Iterative Methods

Because we cannot solve $\boldsymbol{F}(\boldsymbol{q}^{d}) = \boldsymbol{x}^{d}$ for $\boldsymbol{q}^{d}$ (nonlinear equation with no closed-form solution), we take an iterative approach.

- Generate a sequence of values $\boldsymbol{q}^{0}, \boldsymbol{q}^{1}, \boldsymbol{q}^{2} \ldots$ until we find some $\boldsymbol{q}^{N}$ such that $\boldsymbol{F}(\boldsymbol{q}^{N})$ is sufficiently close to $\boldsymbol{x}^{d}$, i.e.,

$$\left\|\boldsymbol{F}(\boldsymbol{q}^{N}) - \boldsymbol{x}^{d}\right\| < \boldsymbol{\epsilon}$$

- In general, at each iteration, we compute $\boldsymbol{q}^{i+1}$ by

$$\boldsymbol{q}^{i+1} = \boldsymbol{q}^{i} + \boldsymbol{\delta q}$$

- The only trick is to determine a good value for $\boldsymbol{\delta q}$ at each iteration.

Gradient Descent

- In general, for gradient descent methods, the goal is to minimize the value of some function $L(q)$ by choosing the updates according to $\boldsymbol{\delta q} = -\nabla \mathbf{L}(\mathbf{q}^{\mathbf{i}})$ where the gradient is w.r.t. $\boldsymbol{q}$.

- To solve the inverse kinematics problem, we wish to minimize the error between $\boldsymbol{F(q)}$ and $\boldsymbol{x^d}$.
- Let's define $L(q)$ in terms of the squared error:

$$L(q) = \frac{1}{2}\left(F(q) - x^d\right)^T \left(F(q) - x^d\right)$$

- For this problem, we define the iteration as

$$\boldsymbol{q^{i+1} = q^i - \alpha_i} \nabla \mathbf{L}(\mathbf{q}^{\mathbf{i}})$$

- We can use the scalar $\boldsymbol{\alpha_i}$ to determine the magnitude of the step size.

- But what exactly is $\nabla \mathbf{L}(\mathbf{q}^{\mathbf{i}})$

Calculating the Error Gradient $(n, m = 2)$

$$L(q) = \frac{1}{2}\left[(f_1(q) - x_1^d) \quad (f_2(q) - x_2^d)\right]\begin{bmatrix} f_1(q) - x_1^d \\ f_2(q) - x_2^d \end{bmatrix}$$

$$= \frac{1}{2}(f_1(q) - x_1^d)^2 + \frac{1}{2}\left(f_2(q) - x_2^d\right)^2$$

$$\nabla L(q) = \begin{bmatrix} \dfrac{\partial L}{\partial q_1} \\ \dfrac{\partial L}{\partial q_2} \end{bmatrix}$$

$$\frac{\partial}{\partial q_1} L(q) = \frac{\partial}{\partial q_1} \; \frac{1}{2}\left\{\left(f_1(q) - x_1^d\right)^2 + \left(f_2(q) - x_2^d\right)^2\right\}$$

$$\frac{\partial}{\partial q_2} L(q) = \frac{\partial}{\partial q_2} \; \frac{1}{2}\left\{\left(f_1(q) - x_1^d\right)^2 + \left(f_2(q) - x_2^d\right)^2\right\}$$

$$\boxed{= \frac{\partial f_1}{\partial q_1}(f_1(q) - x_1^d) + \frac{\partial f_2}{\partial q_1}(f_2(q) - x_2^d)}$$

$$\boxed{= \frac{\partial f_1}{\partial q_2}(f_1(q) - x_1^d) + \frac{\partial f_2}{\partial q_2}(f_2(q) - x_2^d)}$$

Stack these to obtain the gradient.

Calculating the Gradient (cont)

$$\nabla L(q) = \begin{bmatrix} \frac{\partial L}{\partial q_1} \\ \frac{\partial L}{\partial q_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial q_1}(f_1(q) - x_1^d) + \frac{\partial f_2}{\partial q_1}(f_2(q) - x_2^d) \\ \frac{\partial f_1}{\partial q_2}(f_1(q) - x_1^d) + \frac{\partial f_2}{\partial q_2}(f_2(q) - x_2^d) \end{bmatrix}$$

We can write this as a matrix equation

$$= \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_2}{\partial q_1} \\ \frac{\partial f_1}{\partial q_2} & \frac{\partial f_2}{\partial q_2} \end{bmatrix} \begin{bmatrix} f_1(q) - x_1^d \\ f_2(q) - x_2^d \end{bmatrix}$$

And we recognize that $\begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_2}{\partial q_1} \\ \frac{\partial f_1}{\partial q_2} & \frac{\partial f_2}{\partial q_2} \end{bmatrix} = J^T(q)$, the arm Jacobian,

and that $\begin{bmatrix} f_1(q) - x_1^d \\ f_2(q) - x_2^d \end{bmatrix} = (f(q) - x^d)$!

$$= \boldsymbol{J^T(q)}\,(\boldsymbol{f(q) - x^d})$$

And our gradient descent update becomes

$$\boldsymbol{q^{i+1}} = q^i - \alpha_i \nabla \mathrm{L}(\mathrm{q^i}) = \boldsymbol{q^i} + \boldsymbol{\alpha_i}\, \boldsymbol{J^T}(\boldsymbol{q^i})(\boldsymbol{x^d} - \boldsymbol{f}(\boldsymbol{q^i}))$$

Inverse Jacobian Method

Let's take a look at the Taylor series expansion for the forward kinematic map around the i-th iteratate:

$$F(q^i + \delta q^i) = F(q^i) + J(q^i)\delta q^i + h.o.t.$$

Here, *h.o.t.* refers to higher order terms (these go to zero quickly as $\delta q^i \to 0$).

If we neglect the higher order terms, the ideal choice for δq^i would be

$$\delta q^i = q^d - q^i$$

in which case we would step to the goal configuration in a single step!

In this case,

$$\boxed{F(q^i + \delta q^i)} - F(q^i) \approx J(q^i)\delta q^i$$

or

$$\boxed{x^d} - F(q^i) \approx J(q^i)\delta q^i$$

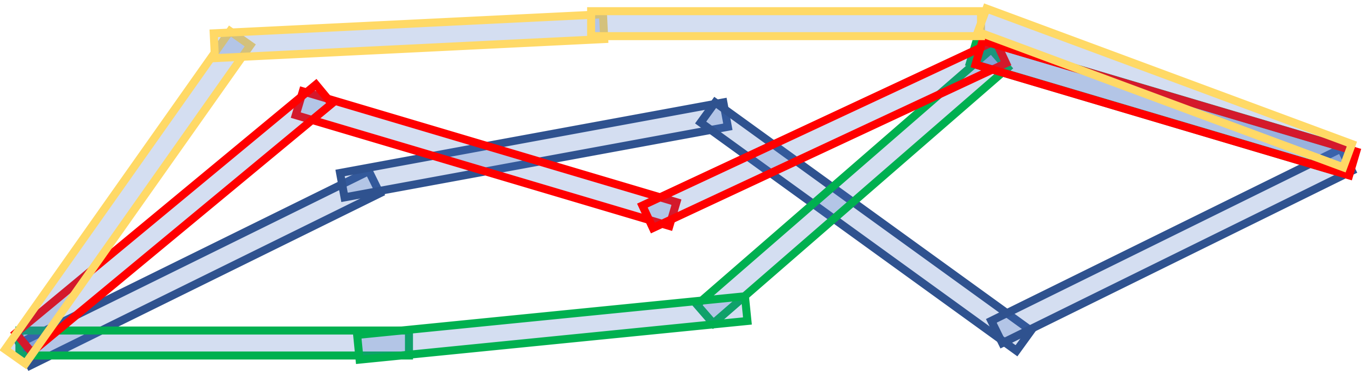
and this leads to the update

$$\boldsymbol{\delta q^i = J^{-1}(q^i)(x^d - F(q^i))}$$

Remember – we don't know q^d.
Happily, q^d does not appear on the r.h.s!

What about redundant arms…

- For the two-link arm, we can **position** the end-effector origin anywhere in the arm's workspace: two inputs (θ_1, θ_2) and two "outputs" (X_e, Y_e).
- For the three-link arm, we can position the end-effector origin anywhere in the arm's workspace, ***<u>and</u>*** we can choose the orientation of the frame: three inputs $(\theta_1, \theta_2, \theta_3)$ and three "outputs" (X_e, Y_e, ϕ).
- Suppose we had a four-link arm?
  - Infinitely may ways to achieve a desired end-effector configuration (X_e, Y_e, ϕ).



The case for $n > m$

In this case, there are "extra" joints:

$$F(q^i + \delta q^i) - F(q^i) \approx J(q^i)\delta q^i$$

If we write this out in detail, we see

$$\begin{bmatrix} f_1(q^i + \delta q^i) \\ \vdots \\ f_m(q^i + \delta q^i) \end{bmatrix} - \begin{bmatrix} f_1(q^i) \\ \vdots \\ f_m(q^i) \end{bmatrix} \approx \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_n} \\ & \ddots & \\ \frac{\partial f_m}{\partial q_1} & \cdots & \frac{\partial f_m}{\partial q_n} \end{bmatrix} \begin{bmatrix} \delta q_1^i \\ \vdots \\ \delta q_n^i \end{bmatrix}$$

Since J is not square, we can't invert it.

- Suppose $rank(J) = m$.
- Then $JJ^T \in \mathbb{R}^{m \times m}$ and $rank\,(JJ^T) = m$
- Then $(JJ^T)^{-1}$ exists

What can we do with this…

Pseudoinverses

- Define $J^+ = J^T(JJ^T)^{-1}$
- Suppose we let $\delta q^i = J^+\{F(q^i + \delta q^i) - F(q^i)\}$

  Then
$$J\delta q^i = JJ^+\{F(q^i + \delta q^i) - F(q^i)\} = F(q^i + \delta q^i) - F(q^i)$$

- In other words,
$$J^+\{F(q^i + \delta q^i) - F(q^i)\} = \delta q^i$$

  is a solution to the equation
$$F(q^i + \delta q^i) - F(q^i) = J(q^i)\delta q^i$$

- We can use this to define our update law:
$$\boldsymbol{\delta q^i = J^+\{F(q^i + \delta q^i) - F(q^i)\}}$$

- J^+ **is called a pseudoinverse.**