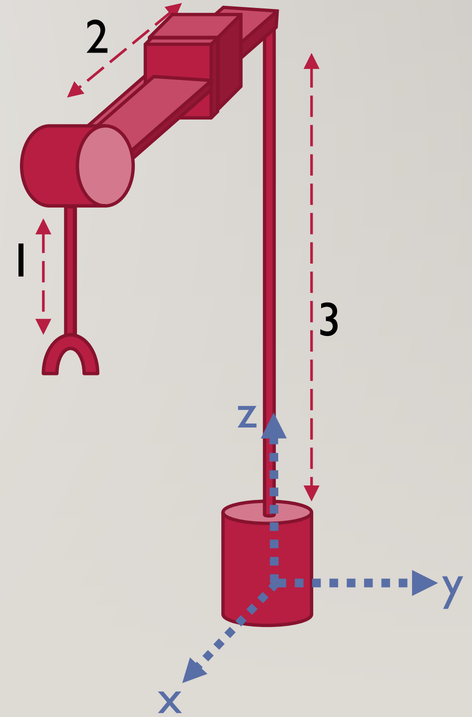
ROBOTICS

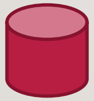
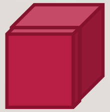
KINEMATICS

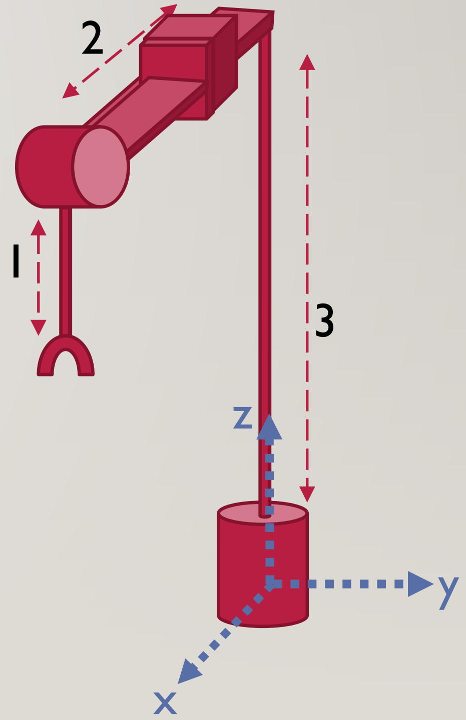
WHAT IS KINEMATICS?

- Robot arms are made up of joints, links and tools
- On a robot arm, we know the joint angles and the lengths of our links
- But where is the end effector?



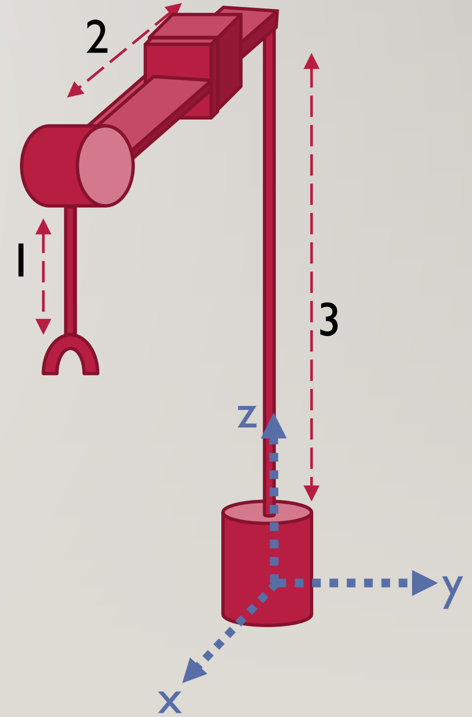
FORWARD KINEMATICS

- The problem of finding the end effector given the joint configurations
- Joint types
  - Revolute 
  - Prismatic 
  - Screw
  - Spherical



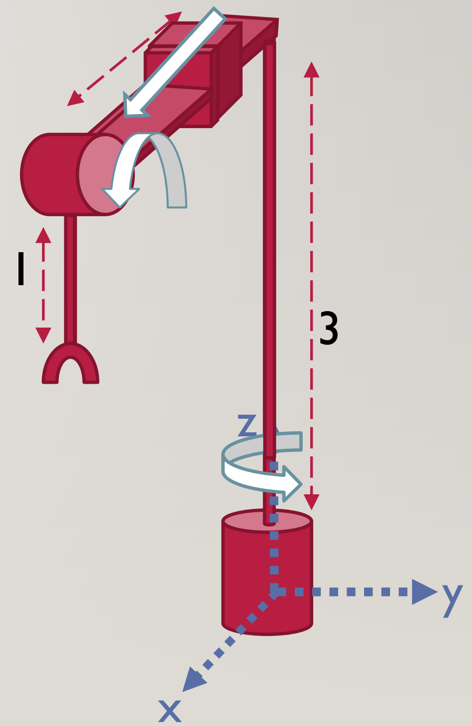
WHAT IS FORWARD KINEMATICS?

- Robot arms are made up of joints, links and tools
- On a robot arm, we know the joint angles and the lengths of our links
- But where is the end effector?
  - Joint 1 is at (0,0,0)
  - Joint 2 is at (0,0,3)
  - Joint 3 is at (2,0,3)
  - Tool point is at (2,0,2)



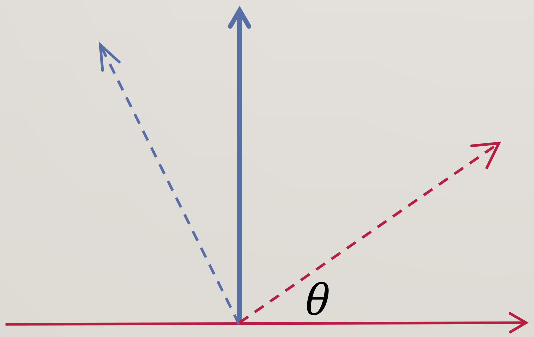
WHAT IS FORWARD KINEMATICS?

- That was straightforward
- But what happens when the joint positions change?
- Now the links no longer align with the axes
    - Trigonometry?
    - Picture a top view of what happens when joint 1 rotates

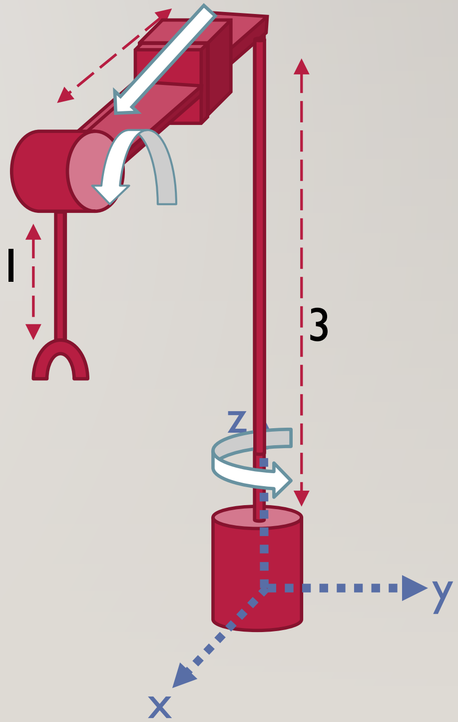


WHAT IS FORWARD KINEMATICS?

- That was straightforward
- But what happens when the joint positions change?
- Now the links no longer align with the axes
  - Trigonometry?
  - Picture a top view of what happens when joint 1 rotates

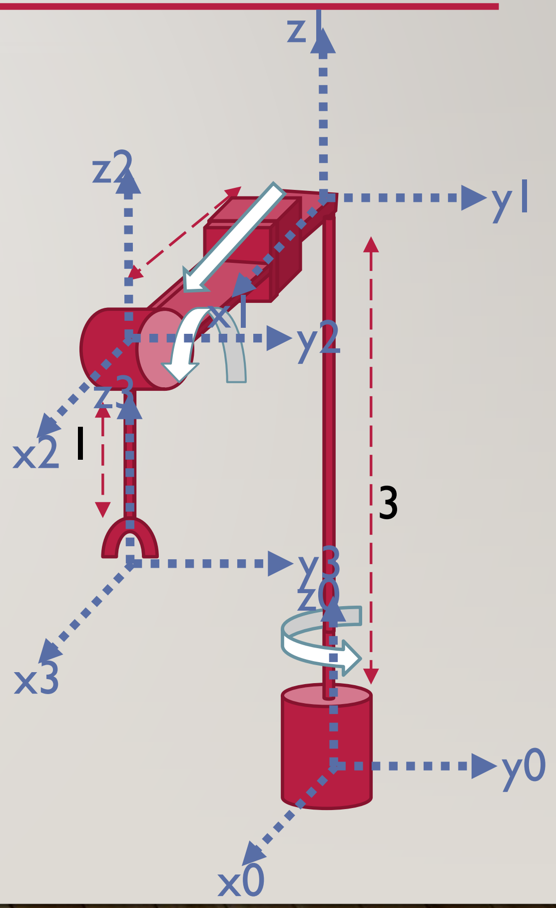


Look familiar?



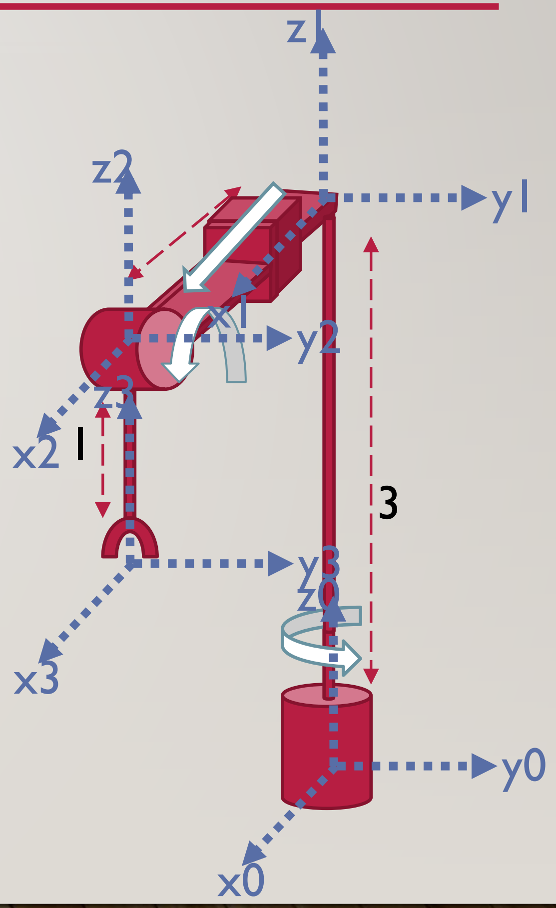
WHAT IS FORWARD KINEMATICS?

- Treat the joints and end effector as coordinate frames



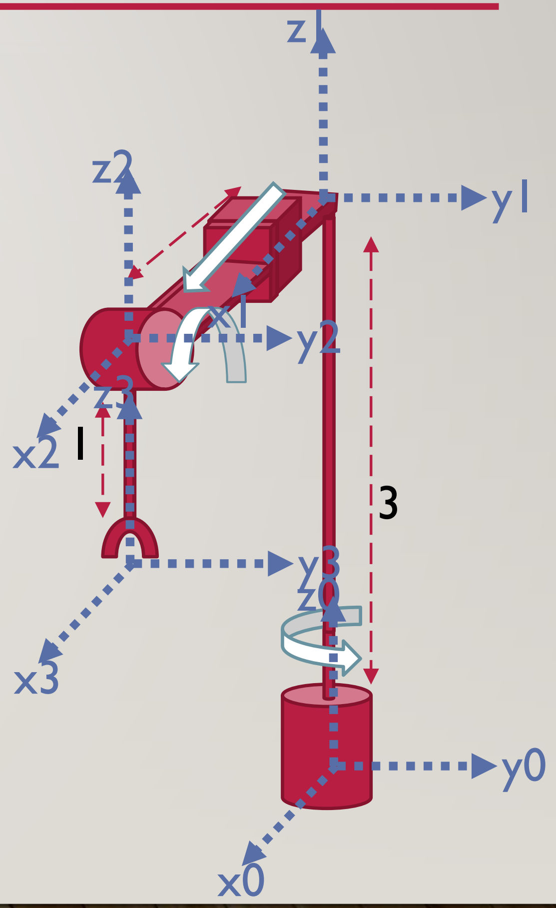
WHAT IS FORWARD KINEMATICS?

- Where is the tool point in reference frame 3?
  - (0,0,0)
- But we only care about reference frame 0 (The world)
- So we find the transformation matrices between the reference frames
- This way, the joint states are parameters in the transformation matrices, and so we can always calculate the position without having to redo our trigonometry



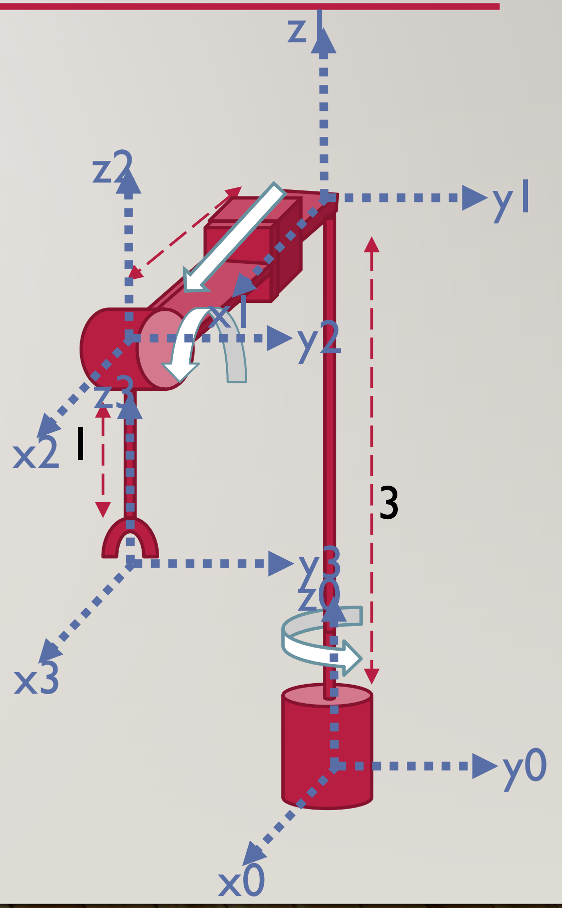
PROBLEM SOLVED?

- Not quite, this is still really hard.
- To get from reference frame 3 to reference frame 2 (T_3^2) we need could potentially rotate around x, rotate around y, rotate around z, translate along x, translate along y, translate along z
- AND, the order is important



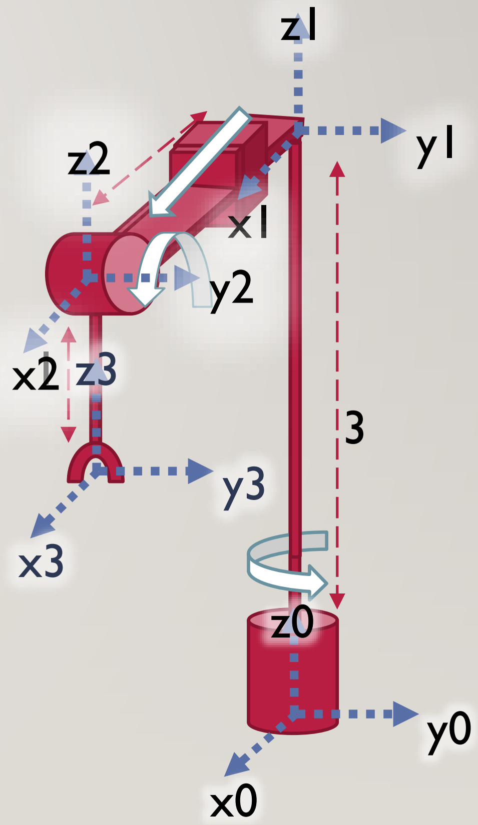
PROBLEM SOLVED?

- So we could have T_3^2=
- $$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) & 0 \\ 0 & \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & q_y \\ 0 & 0 & 1 & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
- And that would just be T_3^2



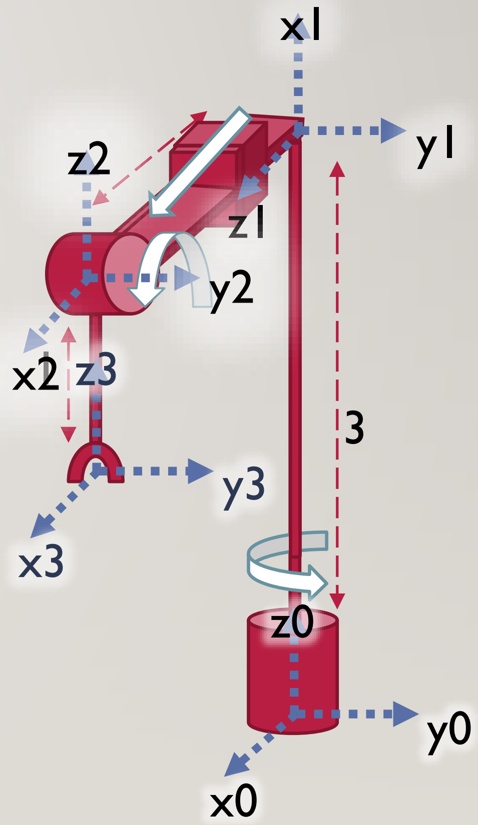
HOW DO WE MAKE IT EASIER?

- The problem is that there is too much freedom, so every transformation requires a lot of thought
- We make use of convention – optional rules that make the thinking a lot easier



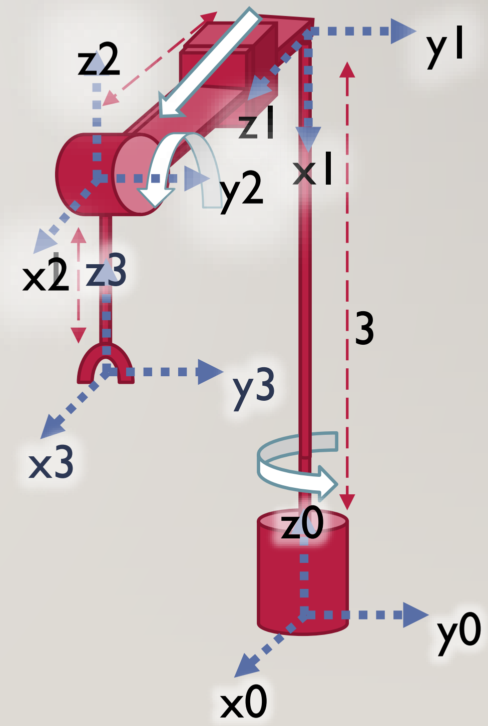
HOW DO WE MAKE IT EASIER?

- Joint effector operates in z
- Right hand axes (Rotation from x to y follows curl of fingers on right hand if thumb is along z)



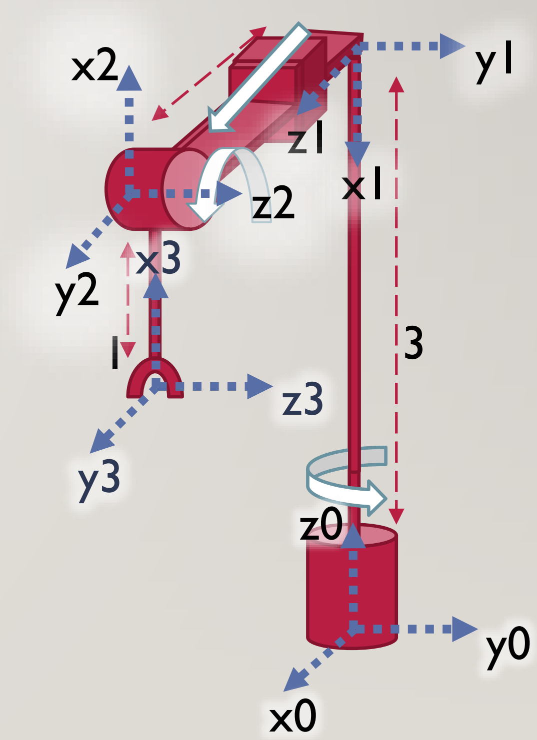
HOW DO WE MAKE IT EASIER?

- Joint effector operates in z
- Right hand axes (Rotation from x to y follows curl of fingers on right hand if thumb is along z)



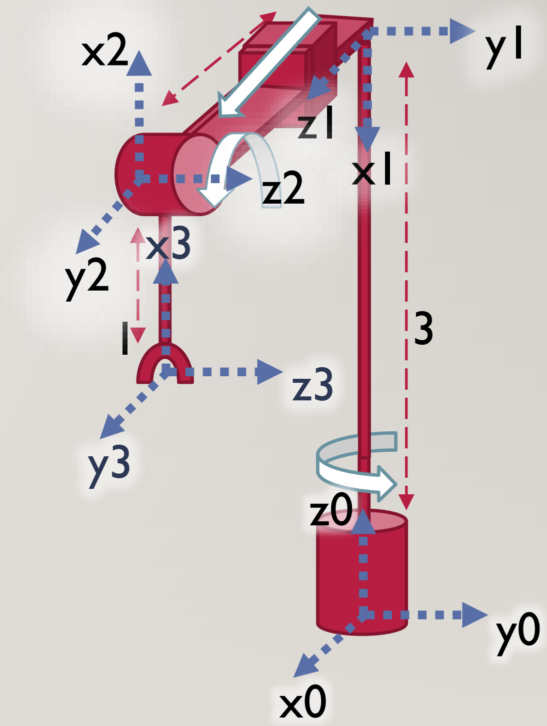
HOW DO WE MAKE IT EASIER?

- Joint effector operates in z
- Right hand axes (Rotation from x to y follows curl of fingers on right hand if thumb is along z)



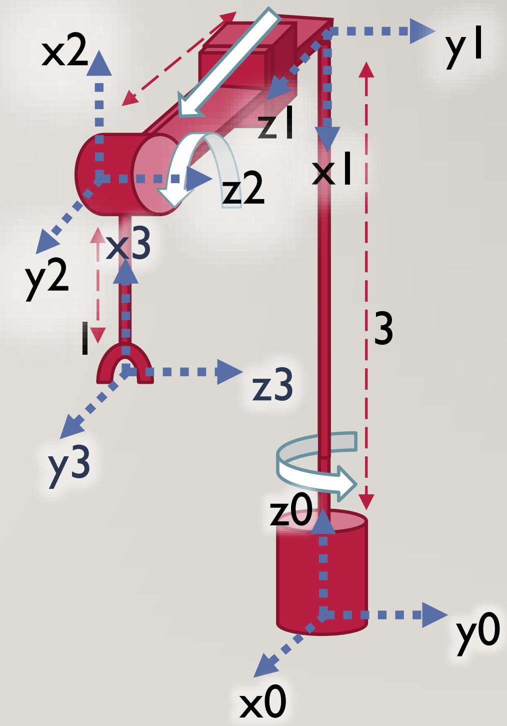
DENAVIT HARTENBERG CONVENTION

- Problem : Lots of rotations and translations to consider
- Intuition : There's only one parameter!
- So we should be able to represent our transformation matrix for joint i as $A_i(q_i)$
- This should be able to represent T_{i-1}^{i}
- Then we should be able to chain these



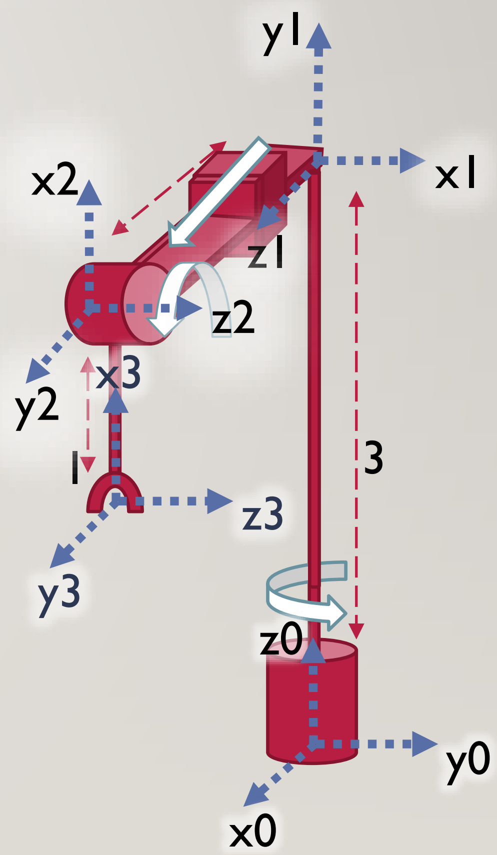
DENAVIT HARTENBERG CONVENTION

- We can reduce the variables we have to play with, and hence make the math easier
- Normally we need 6 variables (a rotation and translation for each axis)
- With DH Convention, we need only 4
- Introduce restrictions around how we choose our axes
- Results in a standard approach, making it easier to communicate



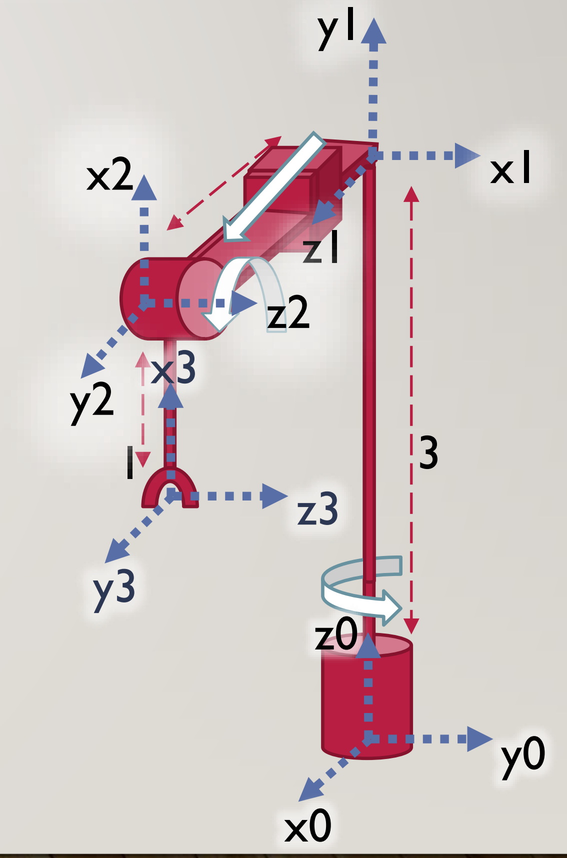
DENAVIT HARTENBERG CONVENTION

- The key is that our joints either rotate in one axis or move along one axis, so we are only operating in one plane
- To remove the y axis, we use the following rules
  - z is the axis of actuation
  - x_i is normal to both z_i and z_{i-1}
  - x_i intersects z_{i-1}



DENAVIT HARTENBERG CONVENTION

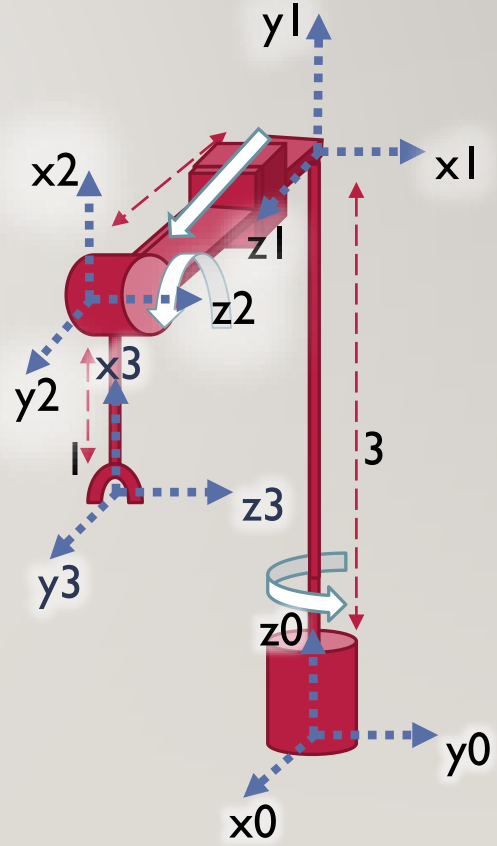
- This allows us to make the whole process algorithmic
- First find the DH parameters
  - a_i is the distance from z_{i-1} to z_i along x_i
  - α_i is the angle from z_{i-1} to z_i around x_i
  - d_i is the displacement from x_{i-1} to x_i along z_{i-1} (Could be a variable if prismatic)
  - θ_i is the angle from x_{i-1} to x_i around z_{i-1} (Could be a variable if revolute)



DENAVIT HARTENBERG CONVENTION

- a_i is the distance from z_{i-1} to z_i along x_i
- α_i is the angle from z_{i-1} to z_i around x_i
- d_i is the displacement from x_{i-1} to x_i along z_{i-1} (Could be a variable if prismatic)
- θ_i is the angle from x_{i-1} to x_i around z_{i-1} (Could be a variable if revolute)

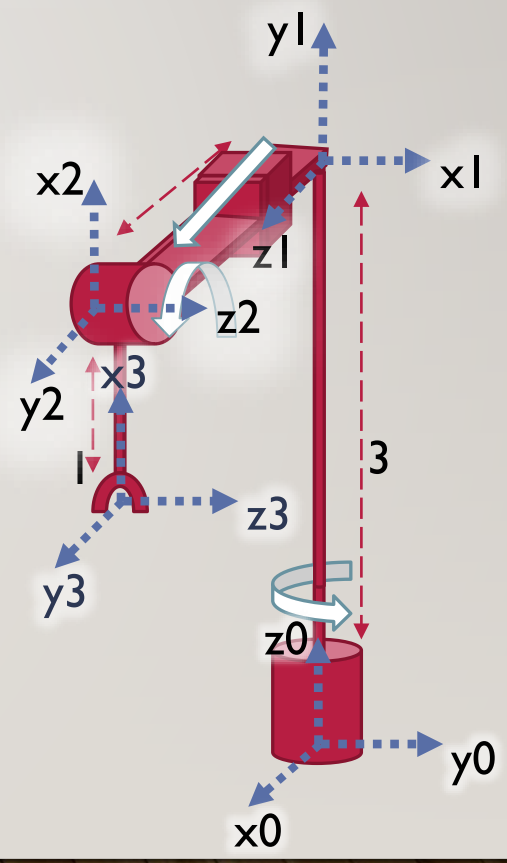
Frame	a	α	d	θ
1				
2				
3				



DENAVIT HARTENBERG CONVENTION

- a_i is the distance from z_{i-1} to z_i along x_i
- α_i is the angle from z_{i-1} to z_i around x_i
- d_i is the displacement from x_{i-1} to x_i along z_{i-1} (Could be a variable if prismatic)
- θ_i is the angle from x_{i-1} to x_i around z_{i-1} (Could be a variable if revolute)

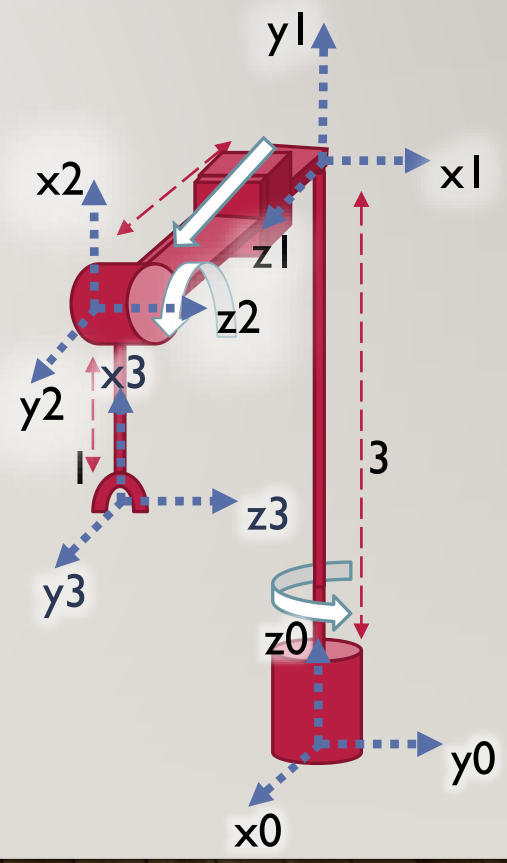
Frame	a	α	d	θ
1	0	90	3	90+q_1
2				
3				



DENAVIT HARTENBERG CONVENTION

- a_i is the distance from z_{i-1} to z_i along x_i
- α_i is the angle from z_{i-1} to z_i around x_i
- d_i is the displacement from x_{i-1} to x_i along z_{i-1} (Could be a variable if prismatic)
- θ_i is the angle from x_{i-1} to x_i around z_{i-1} (Could be a variable if revolute)

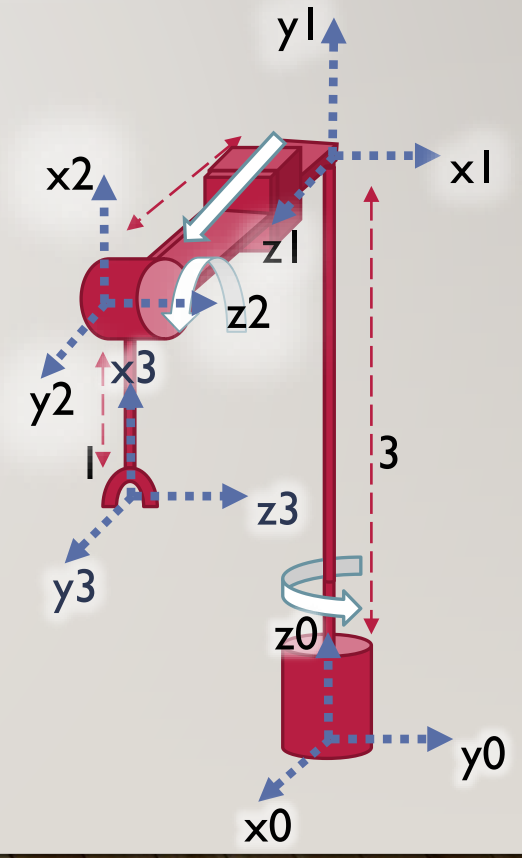
Frame	a	α	d	θ
1	0	90	3	90+q_1
2	0	90	q_2	90
3				



DENAVIT HARTENBERG CONVENTION

- a_i is the distance from z_{i-1} to z_i along x_i
- α_i is the angle from z_{i-1} to z_i around x_i
- d_i is the displacement from x_{i-1} to x_i along z_{i-1} (Could be a variable if prismatic)
- θ_i is the angle from x_{i-1} to x_i around z_{i-1} (Could be a variable if revolute)

Frame	a	α	d	θ
1	0	90	3	90+q_1
2	0	90	q_2	90
3	1	0	0	q_3



WHAT DO I DO WITH THIS INFORMATION?

Frame	a	α	d	θ
1	0	90	3	90+q_1
2	0	90	q_2	90
3	1	0	0	q_3

- This means we can frame the transformation as
  - rotation around x,
  - a shift in x,
  - a shift in z, and
  - a rotation around z, in that order
- Allows us to create a transformation matrix A_i

WHAT DO I DO WITH THIS INFORMATION?

- So $\mathrm{T}_{\mathrm{i}}^{\mathrm{i}-1} = A_i = Rot_z Trans_z Trans_x Rot_x$
- $A_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- $\mathrm{T}_{\mathrm{i}}^{\mathrm{i}-1} = A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- The really cool part about this is it’s consistent