Lab 3: DH Frames Guide

A guide to help you implement Forward Kinematics with DH frames

Updated: Spring 2019

What are Forward Kinematics?

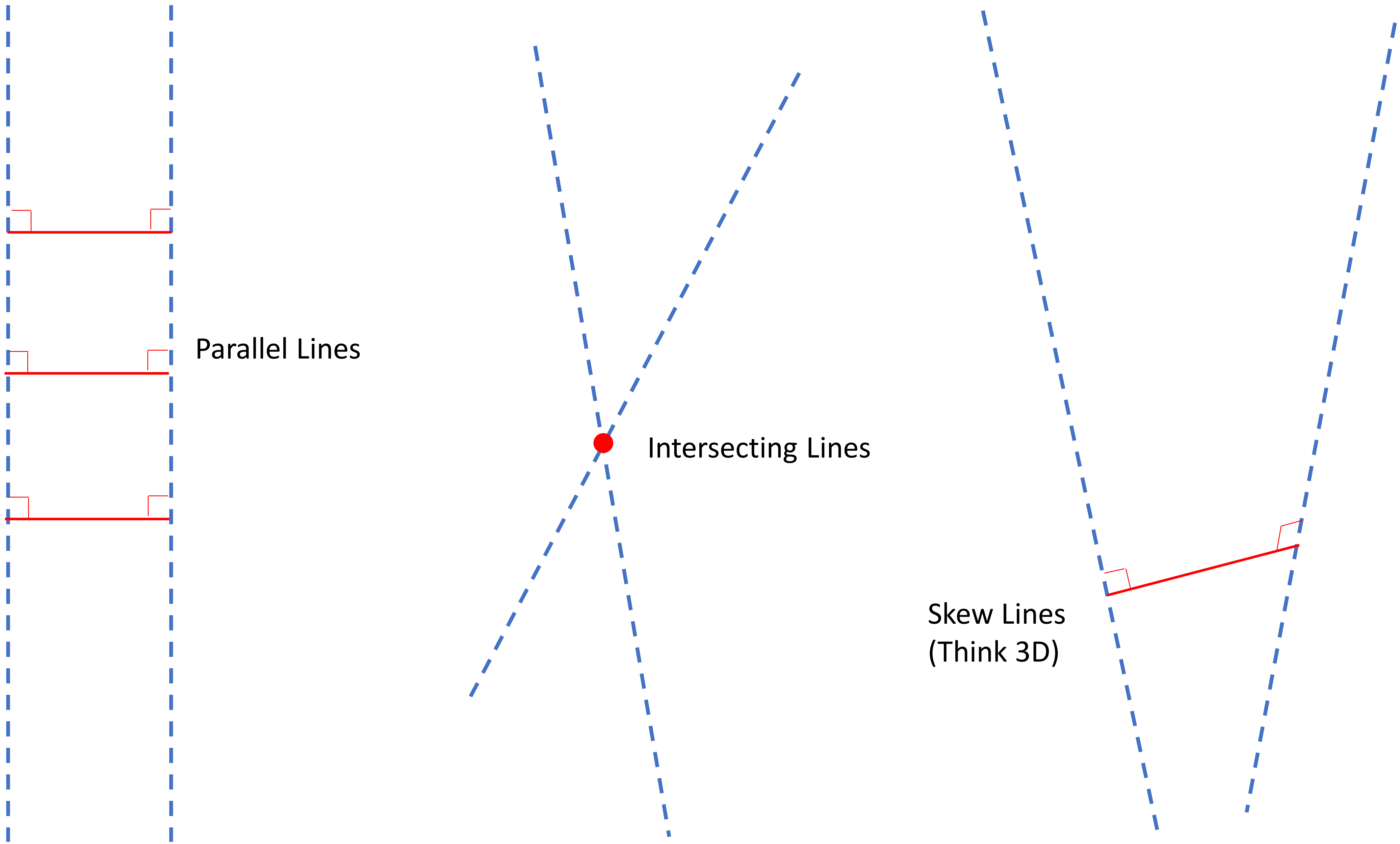
- Forward Kinematics convert from joint angles to an end effector position.
- We are formulating the T_{0n} matrix (ie the homogeneous transform) of Frame n in Frame 0.
- $T_{0n} = \begin{bmatrix} R_{0n} & d_{0n} \\ 0 & 1 \end{bmatrix}$
- What we really care about is the translation vector, d_{0n}, in T_{0n}.
- Thus given the joint angles, we can find the end effector position.
- You might find this video helpful:
  - https://www.youtube.com/watch?v=rA9tm0gTln8

What are DH Frames?

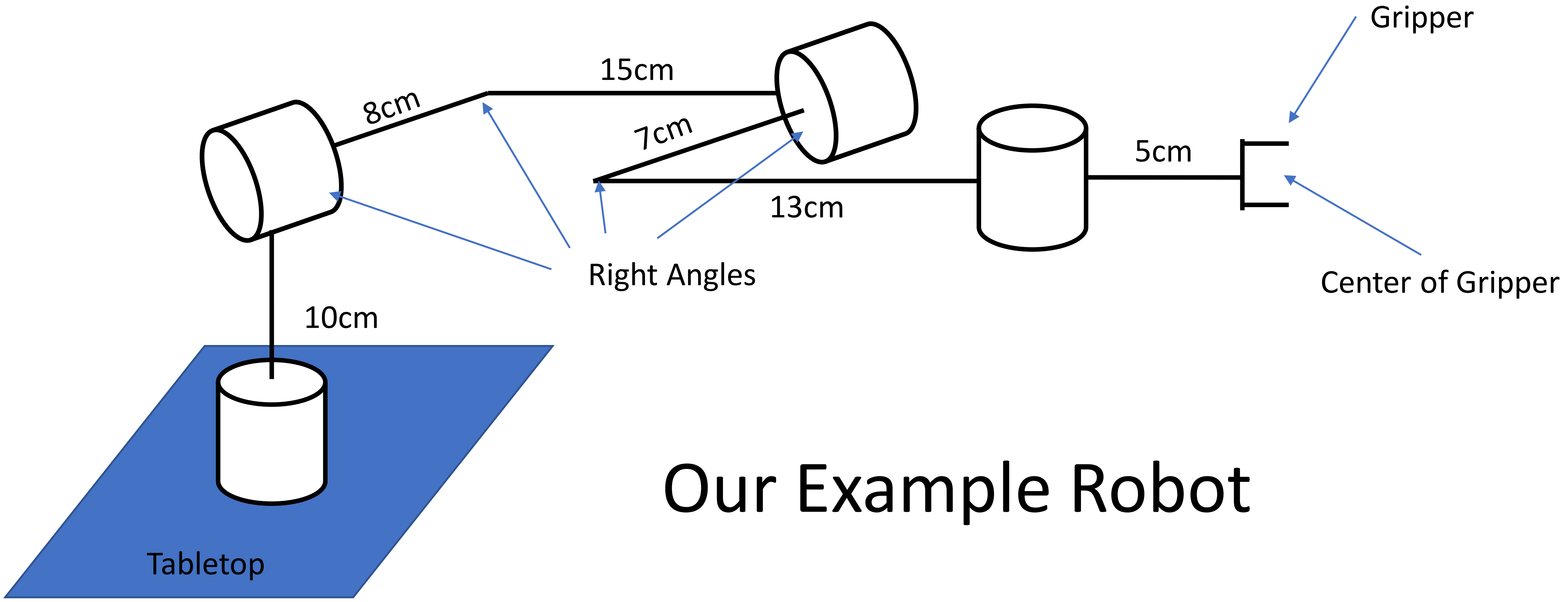
- DH Frames are a way of assigning frames to the joints of robot manipulator
- Their restrictions allow for a minimal representation of each joint's position and orientation and development of the forward kinematics
- Spong's book explains the theory in detail (and I think clearly), but for our purposes all you need to be able to do is find the solution to the forward kinematics
- The physical understanding of a_i, α_i, d_i, and θ_i requires some theory, so I will avoid this if possible (See Last Slide).

What is a common normal?

- The term common normal is used frequently in this document
- Given two lines in space, the common normal is the line segment that is orthogonal to both of them
- If the lines intersect, it is trivial – the “line” has length zero and occurs at the point of intersection
- If the lines are parallel, there are an infinite number of common normals
- If the lines are skew (not parallel, but never intersect), then there is a unique common normal


Parallel Lines
Intersecting Lines
Skew Lines
(Think 3D)

Let's start with this as our robot. Each cylinder represents a revolute joint. Thus, this is an RRRR robot. The numbers represent the length of the links and offsets. This is drawn in bad isometric form, so interpret each angle as a right angle.



Our Example Robot

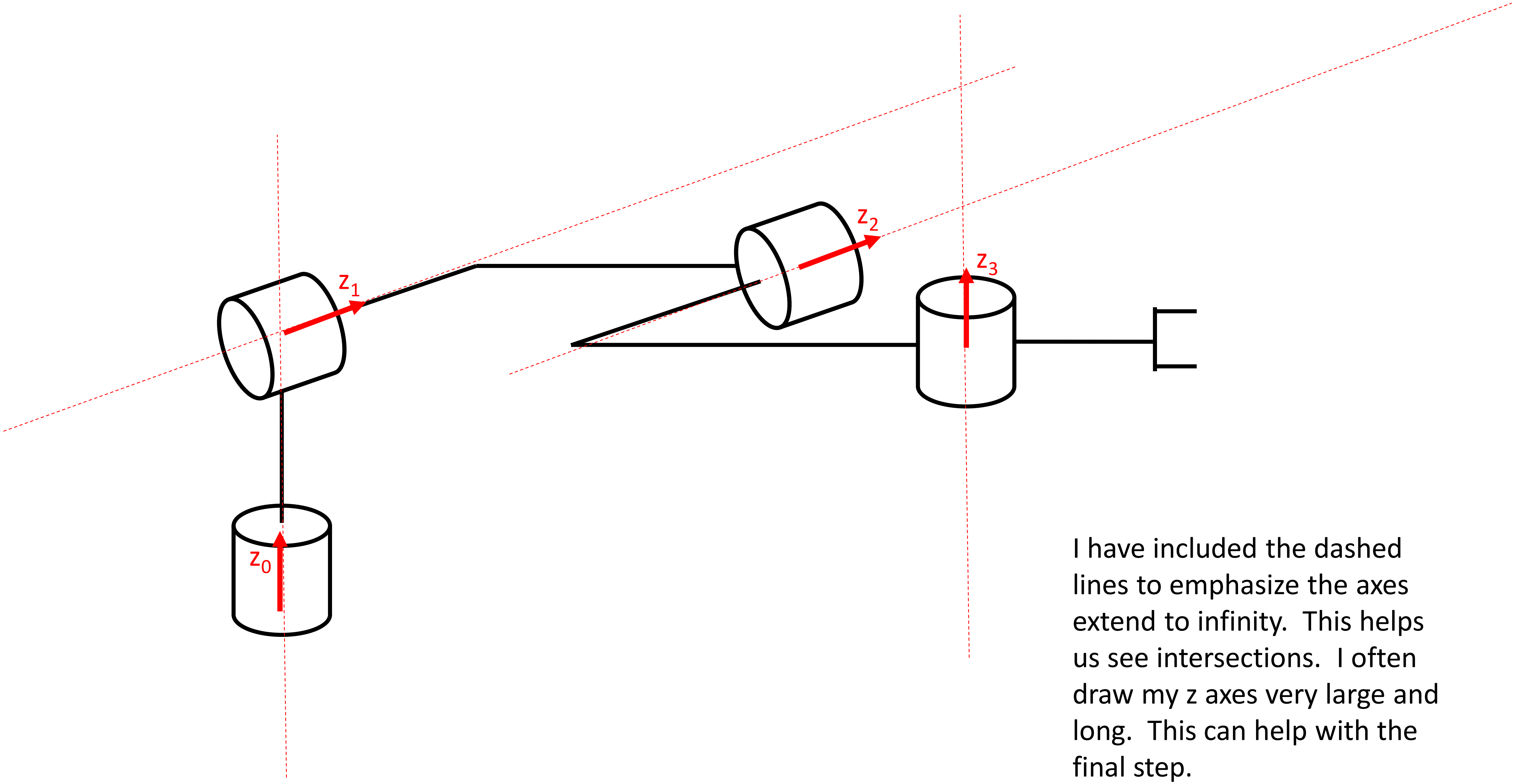
The 6 Step Process

I suggest that you draw or print a copy of the robot diagram and follow along step-by-step. Following each explanation slide is the solution to that step. There is no single solution, but they should give the same final solution (up to an offset).

Step 1: Locate and label the joint axes $z_0, \ldots, z_{n-1}$

- Each z_i is aligned with:
  - The axis of rotation for revolute joints (direction is up to you)
  - The direction of extension for prismatic joints (direction is up to you)

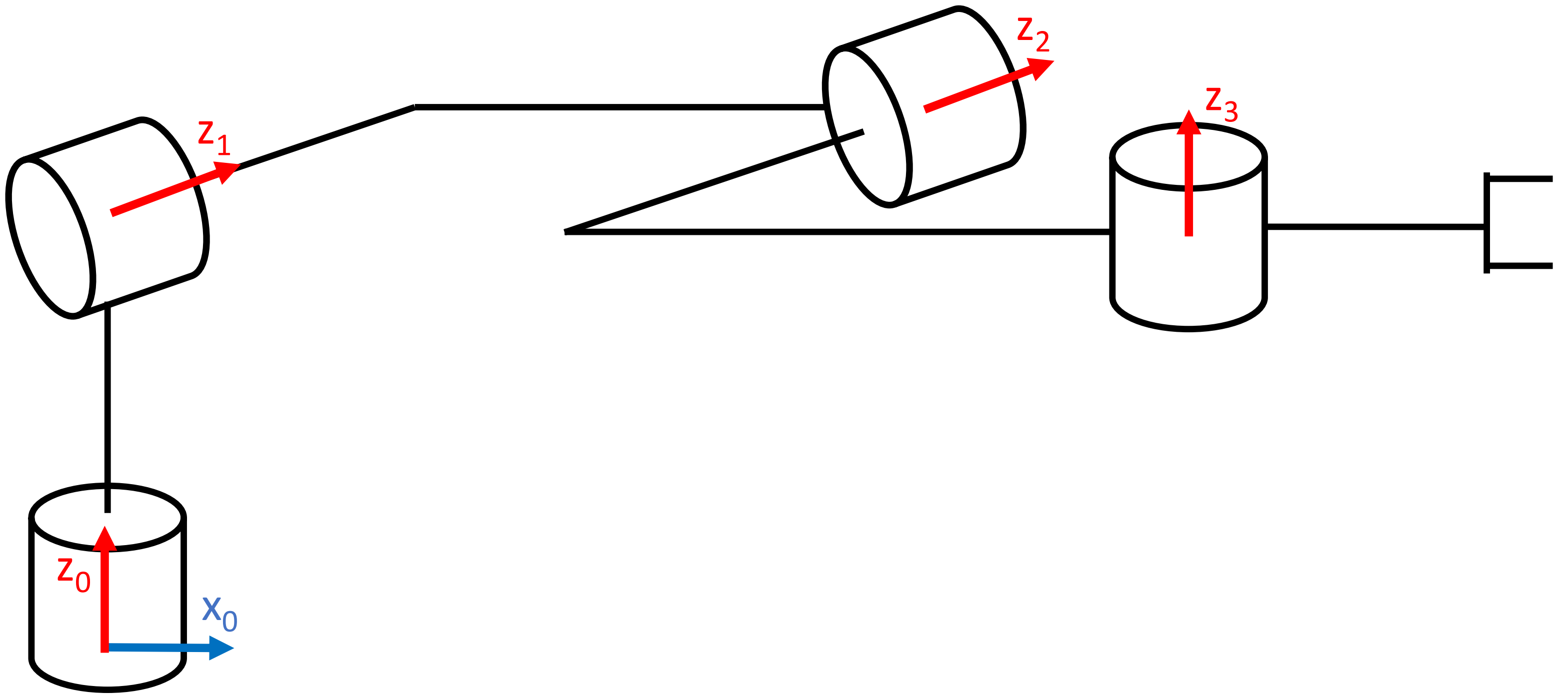
z_0



I have included the dashed lines to emphasize the axes extend to infinity. This helps us see intersections. I often draw my z axes very large and long. This can help with the final step.

Step 2: Establish the base frame

- Set the origin anywhere on the z_0-axis
  - Pick a place that makes your coordinate system make sense
- Select an appropriate direction for x_0
  - Location is selected for convenience.
  - It is normally done with respect to your work surface.
- y_0 is not important, so it is best to leave it out so to avoid clutter (this is true for all the y.)
- For this example, I suggest you set it at the base of the first joint (ie table level)

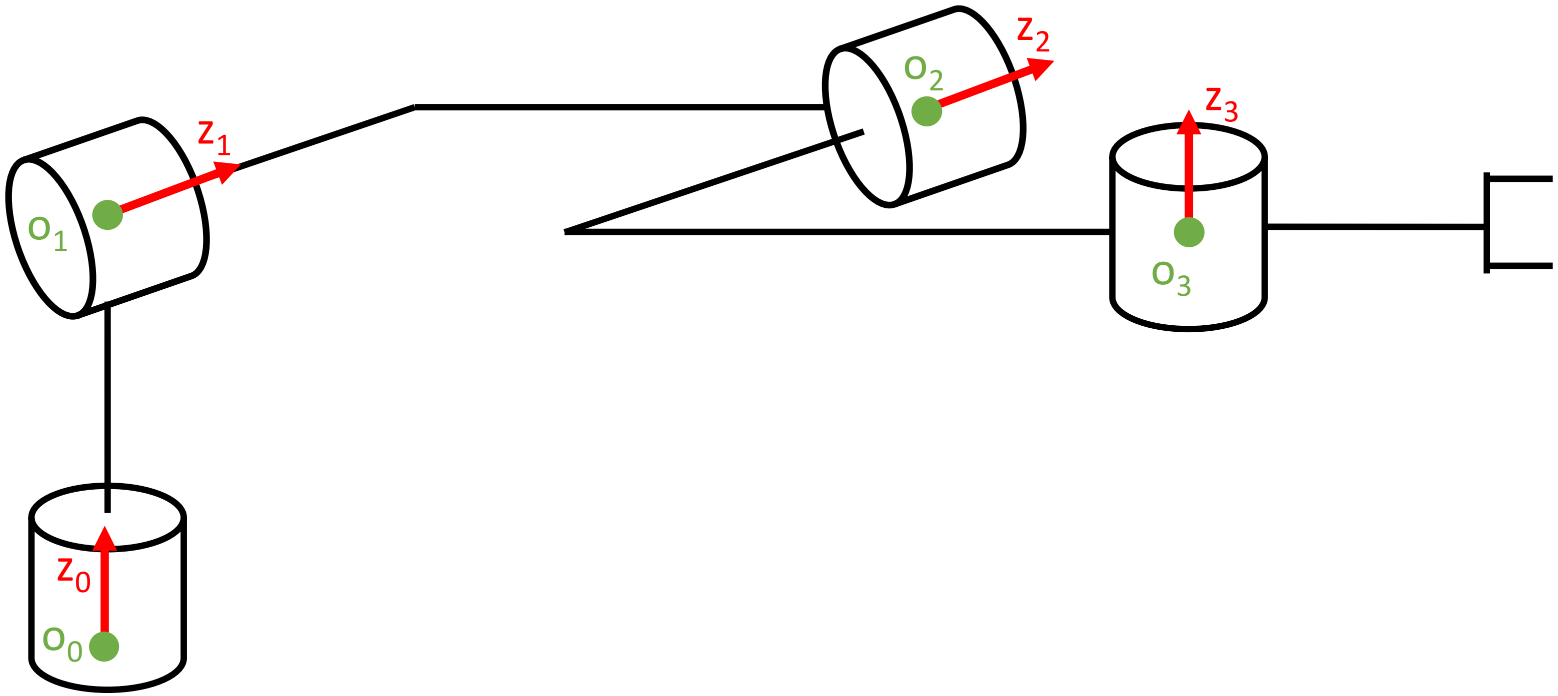


Solution

For Steps 3 and 4, repeat for each joint 1,…, n-1

Step 3: Locating the origin o_i

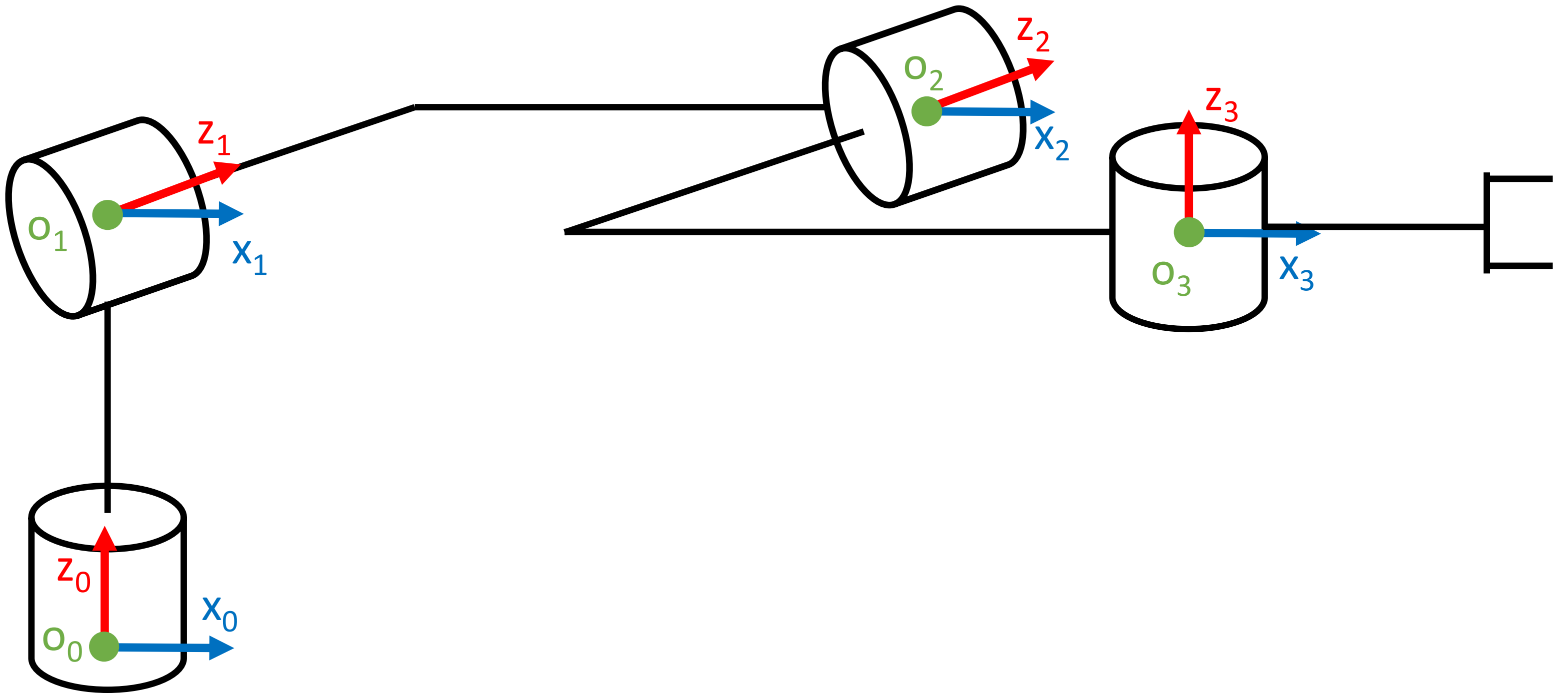
- Locate the origin where the common normal to z_i and z_{i-1} intersect
- If **z_i and z_{i-1} intersect**, locate the origin at the intersection
- If **z_i and z_{i-1} are parallel**, locate the origin at any convenient position along z_i
- Repeat as needed



Solution

Step 4: Establish x_i

- x_i points along the common normal between z_i and z_{i-1} through o_i
- If **z_i and z_{i-1} intersect**, x_i points in the direction normal to the z_{i-1}- z_i plane
- Repeat as needed

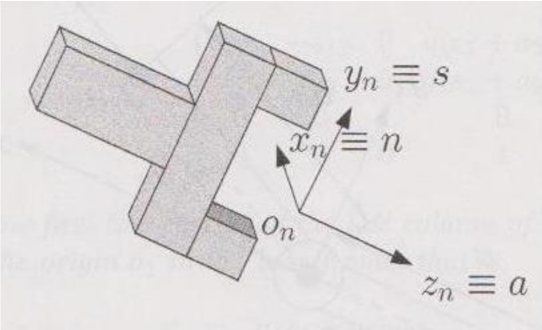


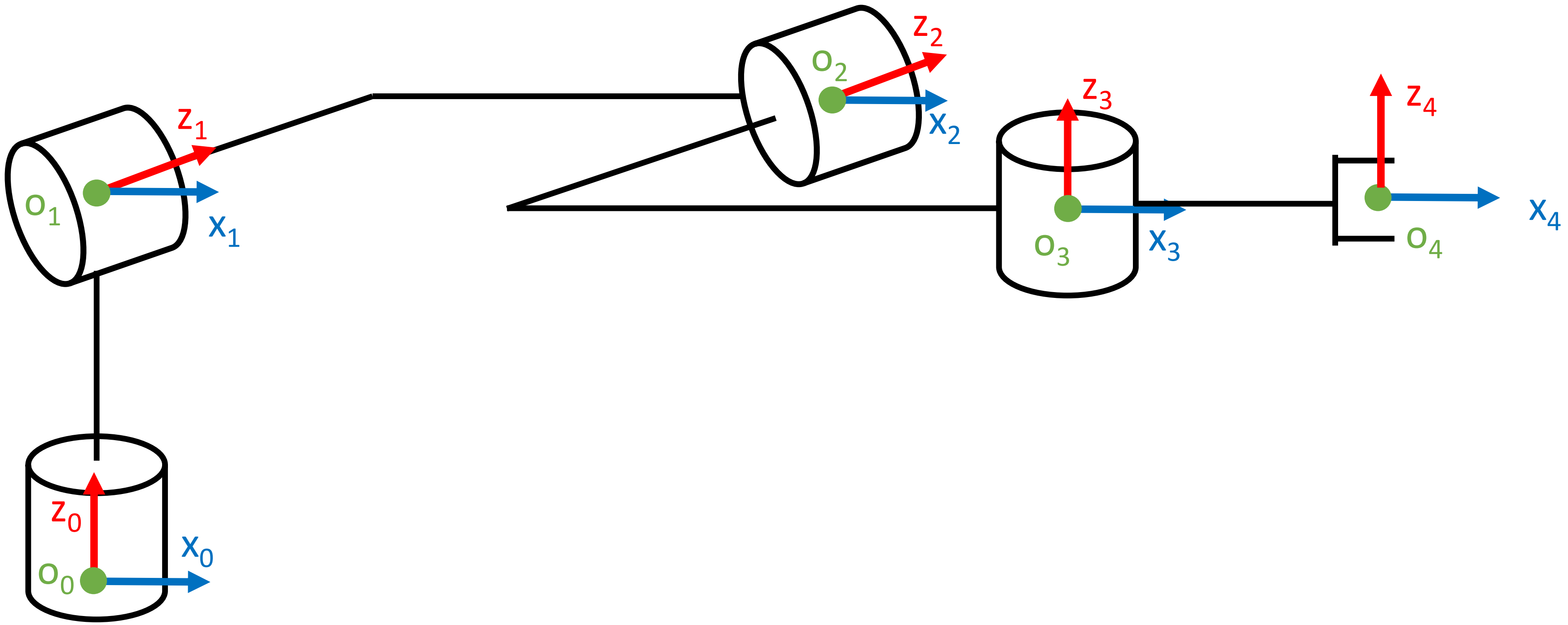
Solution

Step 5: Establish the end effector frame (n)

- There is a lot of flexibility here, but there are conventions used for grippers
- Make z_n parallel to z_{n-1}
- x_n needs to intersect z_{n-1}
- As you gain experience the right choice will be clear and you will be able to adjust to odd cases
- A "normal" robot would also make this more obvious than my silly creation

This is a standard set up, but doesn't work in this case.





Solution

Step 6: Create a table of DH parameters

- a_i - Distance along x_i from the intersection of the x_i and z_{i-1} axes to o_i.
- d_i - Distance along z_{i-1} from o_{i-1} to the intersection of the x_i and z_{i-1} axes. If joint *i* is prismatic, d_i is a variable.
- α_i - The angle from z_{i-1} to z_i measured about x_i (use RHR).
- θ_i - The angle from x_{i-1} to x_i measured about z_{i-1} (use RHR). If joint *i* is revolute, θ_i is a variable.

Joint	a_i	α_i	d_i	θ_i
1				
2				
…				
n-1				

See Measurements Again

Solution to the DH ParameterTable

- The notation θ_1^* or d_1^* is typically used to denote variable parameters
- If you set the frames differently, the numbers will vary, but they will still be the same
- Choosing a different base frame is the biggest difference. It is completely arbitrary and just changes the zero location.

Joint	a_i	α_i	d_i	θ_i
1	0cm	$-\frac{\pi}{2}$	10cm	θ_1^*
2	15cm	0	8cm	θ_2^*
3	13cm	$\frac{\pi}{2}$	-7cm	θ_3^*
4	5cm	0	0cm	θ_4^*

Solution

Calculating T_{04}

- Once you have the parameters, you can calculate A_1, A_2, A_3, and A_4 using the formula pictured here
- $A_i = T_{(i-1)i}$
- Thus $T_{04} = A_1A_2A_3A_4$
- You can do this by hand, but we will use *Robotica* in lab
- When complete, we will extract the translation vector

$$
\begin{aligned}
A_i &= \text{Rot}_{z,\theta_i}\text{Trans}_{z,d_i}\text{Trans}_{x,a_i}\text{Rot}_{x,\alpha_i} \\
&= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\quad \times \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

On to the UR3!

- The Frames have been drawn for you!
- Use Step 6 to fill out a DH Parameter Table.
- Implement them in *Robotica* (Does the figure look correct?).
- Once you finished that, implement it in your code.

Optional Info

For those interested, this the physical interpretation of the parameters: