16-642 Fall 2017: Reference Notes for Kinematics Lecture

George Kantor

18-20 September 2017

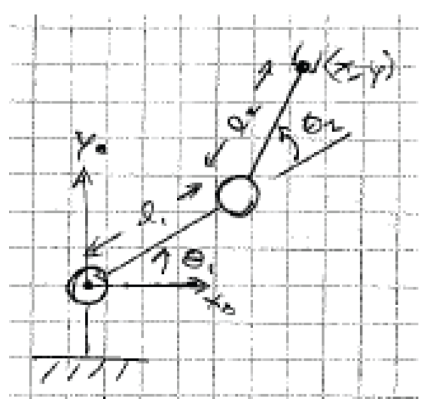
1 Kinematics

We just finished the fundamentals section of the course (and Chapter 2 in Spong et al.) so now we're ready to start talking about the kinematics of robot arms.

Kinematics is the study of motion without regard to the forces that cause it.

1.1 2-link Planar RR Arm

Here is a simple example we will use a lot:



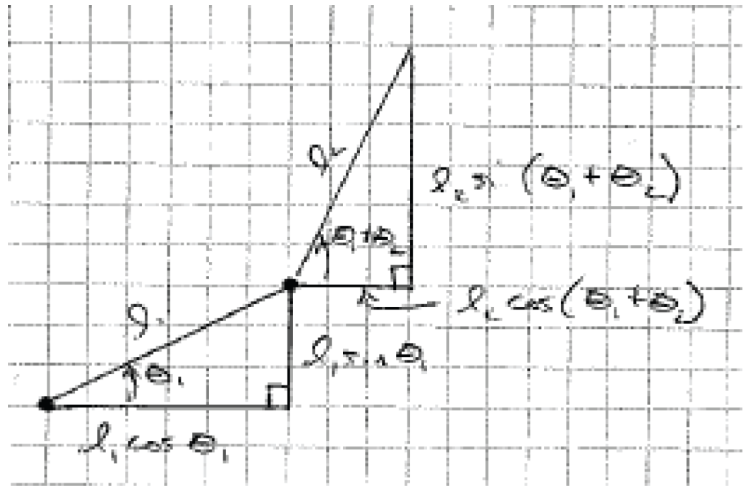
joint coordinates (these are defined by the robot):

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

task coordinates (we pick these, or they are defined by the task):

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

using simple geometry and trig:



$$
\begin{aligned}
x &= \ell_1 \cos\theta_1 + \ell_2 \cos(\theta_1+\theta_2) \\
y &= \ell_1 \sin\theta_1 + \ell_2 \sin(\theta_1+\theta_2)
\end{aligned}
$$

chalk conservation convention:

$$c_i = \cos\theta_i, \qquad s_i = \sin\theta_i$$

$$c_{ij} = \cos(\theta_i+\theta_j), \qquad s_{ij} = \sin(\theta_i+\theta_j)$$

so we can the forward kinematics as

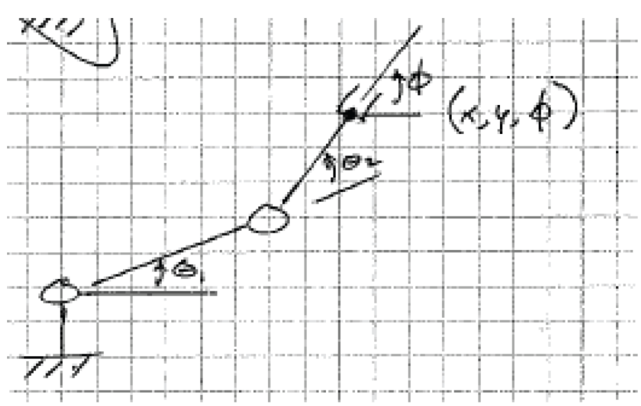
$$\boxed{\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \ell_1 c_1 + \ell_2 c_{12} \\ \ell_1 s_1 + \ell_2 s_{12} \end{bmatrix}}$$

What is the domain and range of this mapping?

- **domain** is the set of possible θ_is (there may be joint angle limits)
- **range** is the set of possible (x, y)s, i.e., the set of points that the robot can *reach*.

Define the **workspace** to be the range of the forward kinematic map.

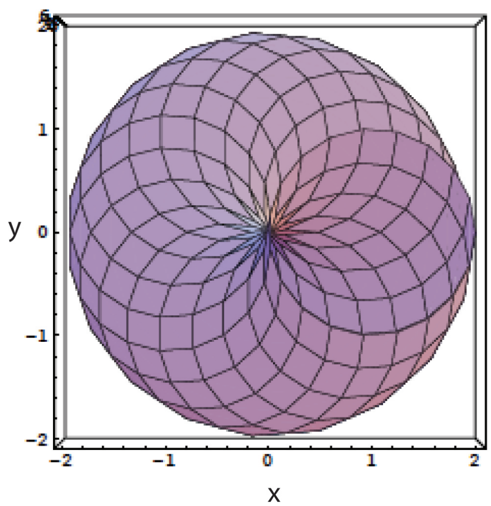
Now, what would happen if we included orientation in the task?



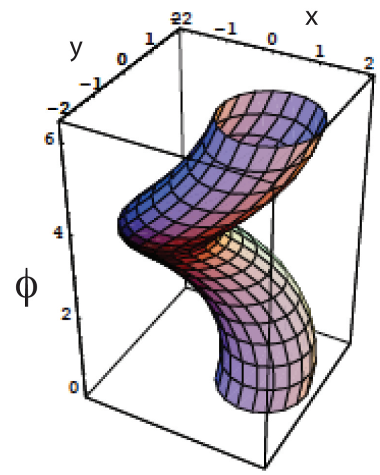
forward kinematics:

$$\begin{bmatrix} x \\ y \\ \phi \end{bmatrix} = \begin{bmatrix} \ell_1 c_1 + \ell_2 c_{12} \\ \ell_1 s_1 + \ell_2 s_{12} \\ \theta_1 + \theta_2 \end{bmatrix}$$

How do we visualize the workspace? We can visualize it as 2D surface in a 3D space:



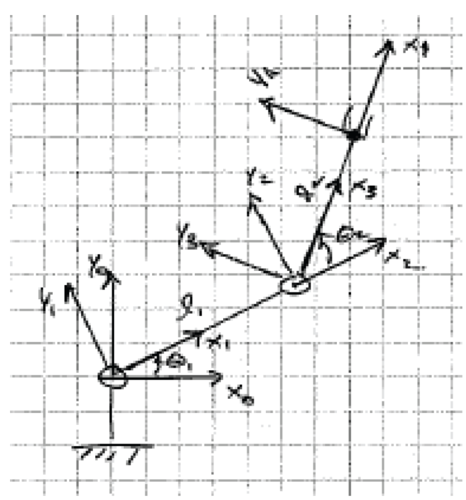
top view



3D view

2 An Alternative Approach to Foward Kinematics

It is also possible to attach frames to the links and use homogeneous transforms:



by inspection:

$$H_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$H_2^1 = \begin{bmatrix} 1 & 0 & \ell_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$H_3^2 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$H_4^3 = \begin{bmatrix} 1 & 0 & \ell_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

from the last lecture we know that

$$\boxed{H_4^0 = H_1^0 H_2^1 H_3^2 H_4^0}$$

3 Denavit-Hartenberg Convention

The idea above is appealing, however it is ad hoc. We'd like to come up with a similar method that is more systematic so that, for example, we could write computer programs to automatically compute the forward kinematics. Today we'll talk about the DH convention, which is a systematic method of attaching frames to each link so that the math all works out easily.

DH places the frames so that there are at most 4 parameters necessary to describe the displacement between frames on adjacent links (instead of 6).

Further, these displacments are composed of a sequence of 4 very simple displacements, making it easier to visualize.

important warning!!! There are a few different variations of the DH convention. In particular, the one we use in this class (called the distal convention) is different than the one that was used two years ago in this class. So be very very careful when looking at old notes and problem set/exam solutions!

4 Assigning Coordinate Frames

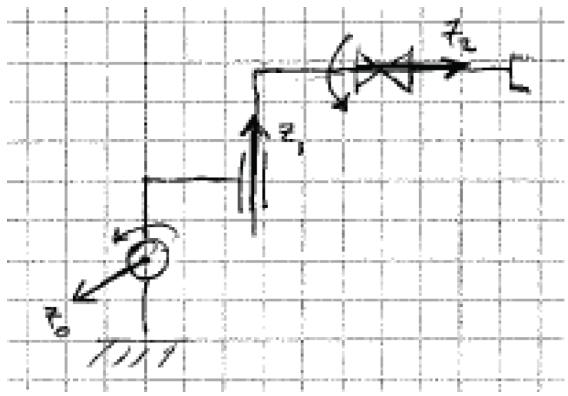
Assume we have an n-link manipulator. We will rigidly attach frames to each link, with frame $\{0\}$ attached to the base/ground, frame $\{1\}$ attached to the first link, frame $\{2\}$ attached to the second link, and so on, up to frame $\{n\}$, which is attached to the last link.

Here is the process for assigning the frames to each link according to the DH convention:

step 1: *assign the z_i axis for $i = 0, 1, 2, \ldots, n-1$.* The axis z_i is defined to be aligned with the axis of joint between link i and link $i+1$. In other words, the axis z_i is aligned with the axis of the $(i+1)$th joint.

- **revolute joint:** axis is axis of rotation, positive direction defined by positive joint angle and right hand rule.
- **prismatic joint:** axis is axis of motion, positive direction defined by positive joint displacement.

Potentially confusing fact: The joint with axis z_i is called joint $i + 1$. Strange but true. This might be less confusing if you remember that z_i is actually attached to the ith link, it just happens to be at the end of that link, so it winds up on top of the $(i + 1)$th joint.



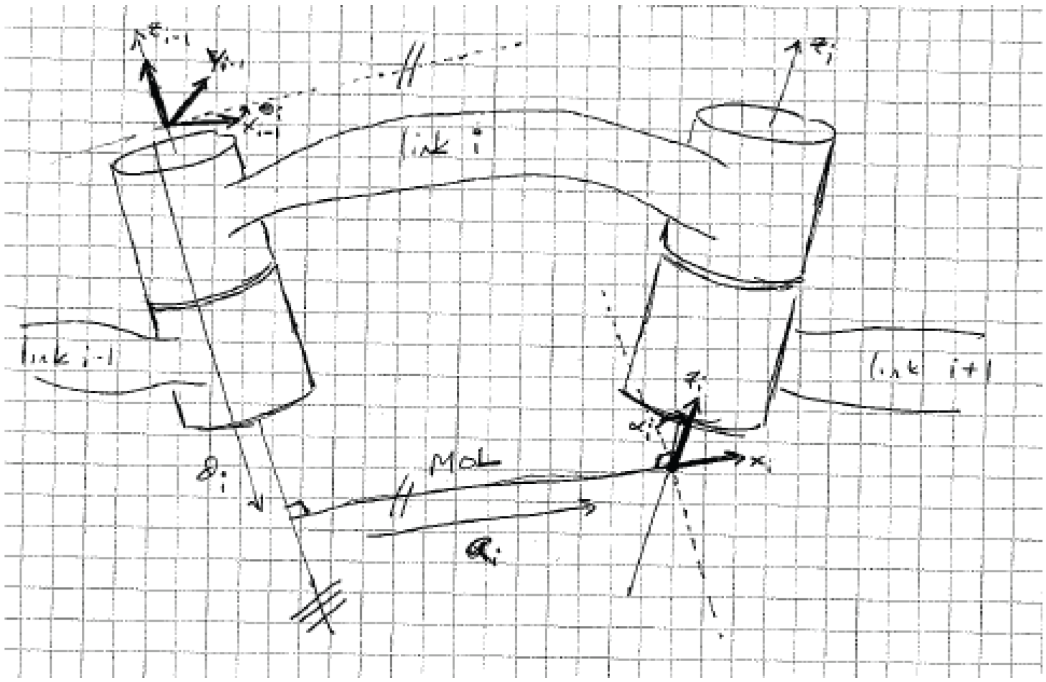
step 2: *assign frame 0.* There is a lot of freedom here. The rules are:

- o_0 can be anywhere on z_0.
- frame must be right handed.
- usually, choose for convenience (this will become more clear).

step 3: *assign frame i for $i = 1$ to $n - 1$ (where n = number of links.)* There are three cases to consider:

case 1: z_i and z_{i-1} are not coplanar. In this case, there exists a unique shortest line segment connecting z_i and z_{i-1}.
This line segment is perpendicular to both axes, so we call it the mutually orthogonal line (MOL).
Frame i can be uniquely defined:

- o_i is a intersection of mutually orthogonal line and z_i.
- x_i points in direction of the MOL away from z_{i-1}.
- y_i satisfies right hand rule.



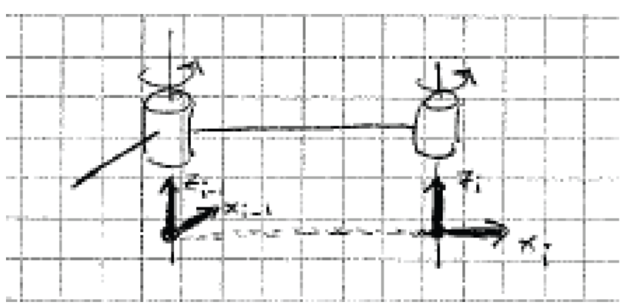
Note that with this definition, the following sequence of motions will yield the displacement between frame $i-1$ and frame i:

1. rotate about z_{i-1} until new x axis is aligned with MOL. **Call this angle** θ_i.
2. translate along z_{i-1} (which also happens to be current z axis) until new o is at intersection of z_{i-1} and MOL. **Call this distance** d_i**.**
3. translate along current x axis until o is at intersection of MOL and z_i axis. **Call this distance** a_i**.**
4. rotate about current x axis until z axis is aligned with z_i. **Call this angle** α_i**.**

The numbers θ_i, d_i, a_i, and α are the Denavit Hartenberg parameters:

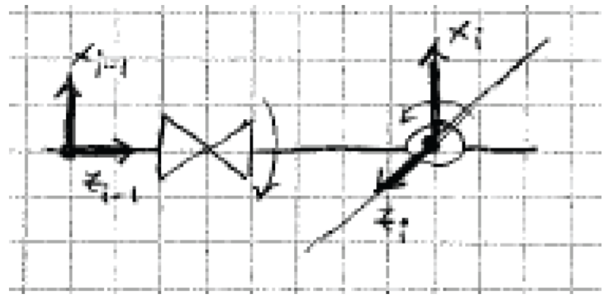
- θ_i is called joint angle
- d_i is called link offset
- a_i is called link length
- α_i is called link twist

case 2: z_i **and** z_{i-1} **are parallel.** In this case there are infinitely many MOLs, so choose any MOL and follow the guidelines for case 1. A common choice is to pick the MOL so the d_i winds up being zero.



case 3: z_i **intersects** z_{i-1}**.** In this case:

- o_i must be at the intersection of the two axes.
- x_i must be chosen orthogonal to both axes, direction is arbitrary.



step 4: Assign the last frame. There is a lot of freedom here. The hard rule is that the last frame must be chosen so that it can be reached from the previous frame using the 4 DH motions. If there is a gripper at the end, then the last frame is usually assigned as follows:

- o_n is placed symmetrically between the fingers of the gripper.
- y_n is chosen point in the directions that the fingers move.
- z_n is chosen to point "out" (as we'll see later, in most robots this will be consistent with picking z along the joint axis of the last joint).
- x_n is chosen according to the right hand rule.

5 The Big Equation

The sequence of simple displacments defined by the DH parameters are all defined in the current frame, which means that each subsequent displacment is applied by multiplying on the right, so:

$$H_i^{i-1} = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i}.$$

where

$$Rot_{z,\theta_i} = \begin{bmatrix} & & & 0 \\ & R_{z,\theta_i} & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Trans_{z,d_i} = \begin{bmatrix} & & & 0 \\ & I & & 0 \\ & & & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Trans_{x,a_i} = \begin{bmatrix} & & & a_i \\ & I & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rot_{x,\alpha_i} = \begin{bmatrix} & & & 0 \\ & R_{x,\alpha_i} & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So when you mulitply this out, you get a very important equation, which is also on page 77 in SHV:

$$H_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

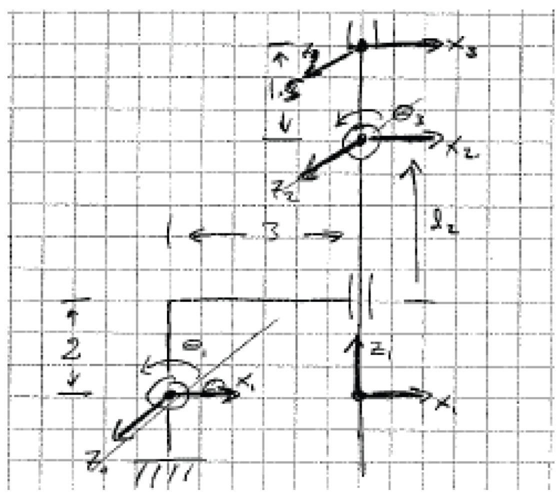
6 The Process

1. assign frames
2. get DH parameters, build table
3. compute each H_i^{i-1} using big equation
4. compute $H_n^0 = H_1^0 H_2^1 \cdots H_n^{n-1}$

7 Some Examples

7.1 RPR manipulator

Consider the RPR manipulator, drawn below in the “zero position”:

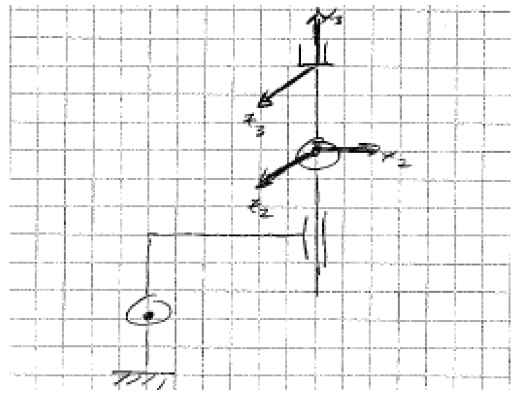


The DH parameters:

i	θ_i	d_i	a_i	α_i
1	θ_1	0	3	-90^o
2	0	$\ell_2 + 2$	0	90^o
3	$\theta_3 + 90^o$	0	1.5	0

Note that when we get to the third row of the table, we realize that we made a bad choice for frame 3: it is impossible to get from frame 2 to frame 3 using the 4 DH motions. We have two choices:

1. choose a new frame:

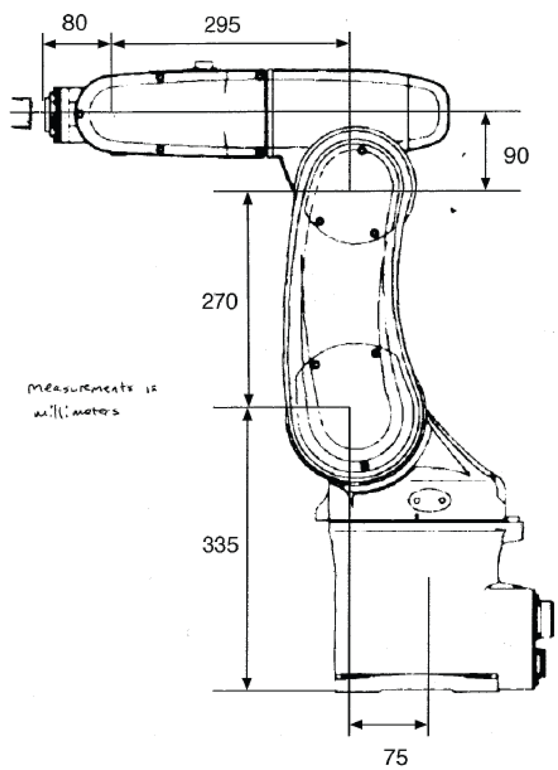


2. forget DH for and just write H_3^2 by inspection:

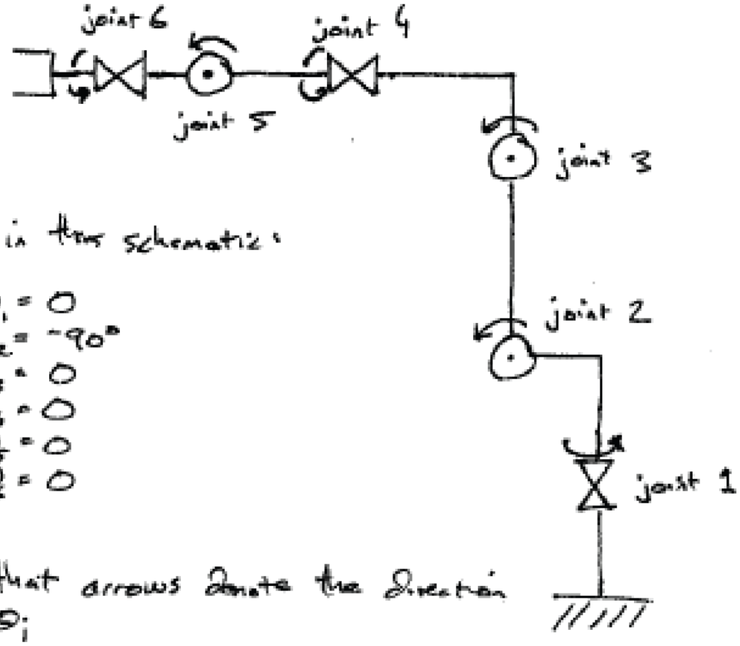
$$H_3^2 = \mathrm{Rot}_{z,\theta_3}\mathrm{Trans}_{y,1.5}$$

7.2 DH for the Denso Arm

Here's the Denso arm, a 6 DOF manupulator that we we use in undergraduate the lab:



The schematic associated with this picture is:

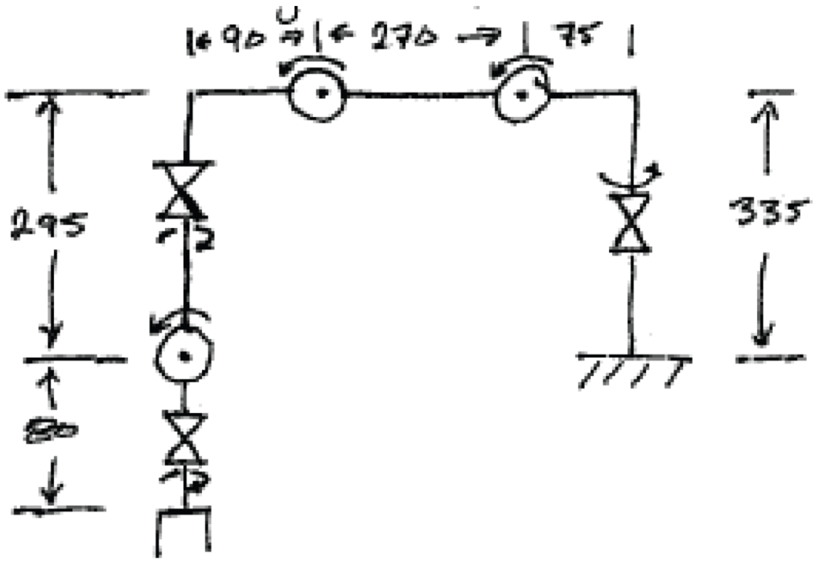


Note that in this schematic:

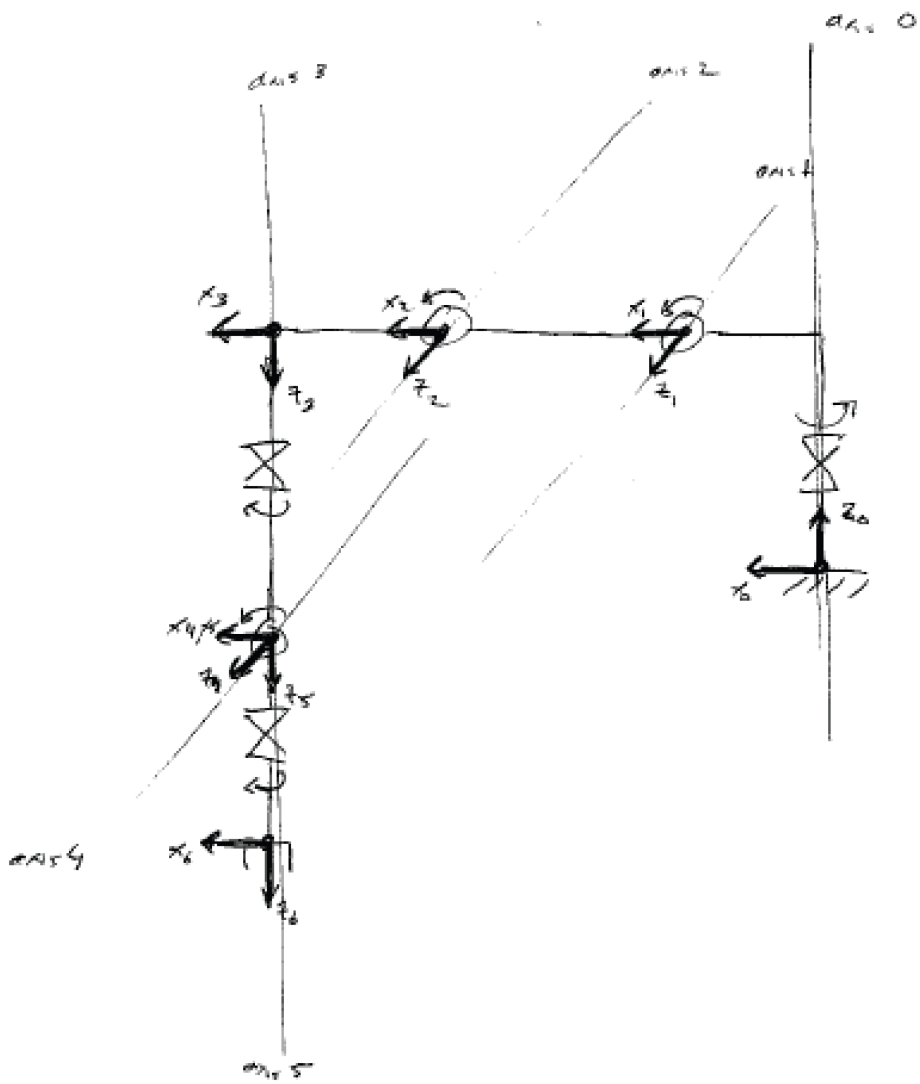
$\theta_1 = 0$
$\theta_2 = -90°$
$\theta_3 = 0$
$\theta_4 = 0$
$\theta_5 = 0$
$\theta_6 = 0$

Note also that arrows denote the direction of positive θ_i

In order to apply DH, it is easiest to work from the schematic with all joint angles set to zero:



Next we sketch in all of the joint axis lines, then we add the frames starting with frame 0.



Next we make a table of DH parameters:

i	θ_i	d_i	a_i	α_i
1	θ_1	335	75	-90^o
2	θ_2	0	270	0
3	θ_3	0	90	-90^o
4	θ_4	295	0	90^o
5	θ_5	0	0	-90^o
6	θ_6	80	0	0

The last step is to compute H_1^0, H_2^1, up to H_6^5 using the big equation. Then you can finally get the forward kinematics by multiplying:

$$H_6^0 = H_1^0 H_2^1 H_3^2 H_4^3 H_5^4 H_6^5.$$

8 Inverse Kinematics

8.1 Setting up the Problem

Let Θ be a vector of joint angles. Forward kinematics tells us how to write the end effector position as a function of Θ, i.e., we can find $H_n^0(\Theta)$.

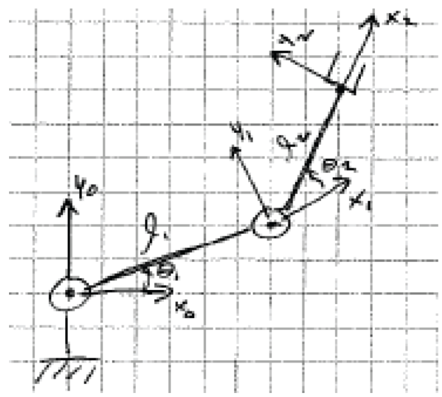
Note that it is common to denote the vector of joint angles with the letter q instead of Θ and $T_n^0(q)$ instead of $H_n^0(\Theta)$. We will use both notations, so you should be familiar with both.

Now, let H be the homogeneous matrix that describes the desired end effector pose. Then inverse kinematics is simply a matter of solving the set of nonlinear equations:

$$T_n^0(q) = H$$

For this class, that is all you really need to know: how to set up the inverse kinematics problem and what it means to solve it. Actually solving it involves sovling systems of nonlinear equations, which is usually pretty messy. The rest of this section was not covered in class, but is included here for your reference.

8.2 Example: Planar RR Arm



Recall that for $q = [\,\theta_1,\ \theta_2\,]^T$, the forward kinematics are

$$T_2^0(q) = \begin{bmatrix} c_{12} & -s_{12} & 0 & \ell_1 c_1 + \ell_2 c_{12} \\ s_{12} & c_{12} & 0 & \ell_1 s_1 + \ell_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Let the desired end effector position be

$$H = \begin{bmatrix} c\phi & -s\phi & 0 & x \\ s\phi & c\phi & 0 & x \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So $T_2^0(q) = H$ is equivalent to

$$c_{12} = c\phi \tag{1}$$

$$s_{12} = s\phi \tag{2}$$
$$\ell_1 c_1 + \ell_2 c_{12} = x \tag{3}$$
$$\ell_1 s_1 + \ell_2 s_{12} = y \tag{4}$$

First two equations yield one equation: $\phi = \theta_1 + \theta_2$.

So we have three equations, two unknowns. Obviously, a solution will not exist for most choices of x, y, ϕ.

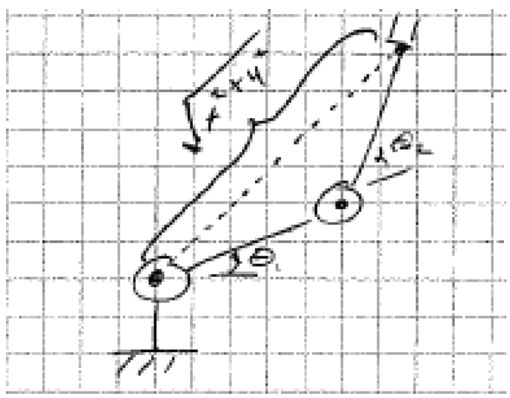
So for now, let's forget about ϕ, and just worry about x and y.

There is not systematic way of solving nonlinear equations, but we do have a few standard tricks to try:

look at the square of the distance:
Square both sides of Equations **??** and **??**, then add them together to get

$$x^2 + y^2 = \ell_2^2 c_{12}^2 + 2\ell_1\ell_2 c_1 c_{12} + \ell_1^2 c_1^2 + \ell_2^2 s_{12}^2 + 2\ell_1\ell_2 s_1 s_{12} + \ell_1^2 s_1^2$$

Intuitively, this quantity should independent of θ_2:



So we should be able to do some mathematical manipulations to make the dependence on θ_1 disappear, then we can solve the resulting expression for θ_2. How do you do that?

use lots of trig identities:
Recall:

$$\cos(\alpha \pm \beta) = \cos\alpha\cos\beta \mp \sin\alpha\sin\beta$$

So

$$c_1 c_{12} + s_1 s_{12} = \cos(\theta_1 - (\theta_1 + \theta_2)) = \cos(-\theta_2) = c_2.$$

Also recall $s^2 + c^2 = 1$.

So we get

$$x^2 + y^2 = \ell_1^2 + \ell_2^2 + 2\ell_1\ell_2 c_2$$

$$\boxed{\implies \quad \theta_2 = \pm\text{acos}\left(\frac{1}{2\ell_1\ell_2}\left(x^2 + y^2 - \ell_1^2 - \ell_2^2\right)\right)}$$

Note that solution only exists if

$$-1 \leq \frac{1}{2\ell_1\ell_2}\left(x^2 + y^2 - \ell_1^2 - \ell_2^2\right) \leq 1$$

which is the same as

$$-2\ell_1\ell_2 + \ell_1^2 + \ell_2^2 \leq x^2 + y^2 \leq 2\ell_1\ell_2 + \ell_1^2 + \ell_2^2$$

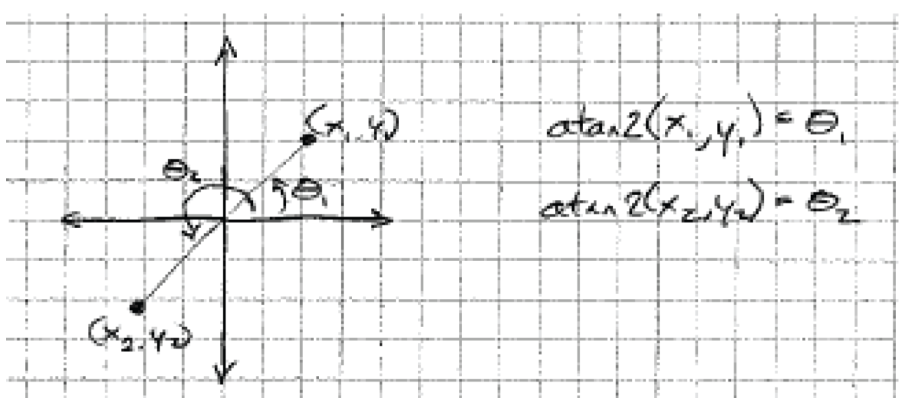
which is the same as

$$(\ell_1 - \ell_2)^2 \leq x^2 + y^2 \leq (\ell_1 + \ell_2)^2$$

This has geometric meaning – I encourage you to think about it and see how it relates to our earlier discussion of the workspace of a planar RR arm.

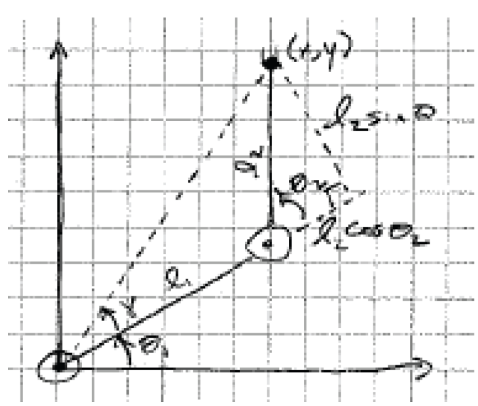
Now let's get back to finding θ_1.

use the 4-quadrant arctangent:



The function $\mathrm{atan2}$ is defined so that $\mathrm{atan2}(y, x) = \theta$ in all four quadrants.

Now, applying atan2 to the case at hand:



$$\theta_1 + \gamma = \operatorname{atan2}(y, x)$$

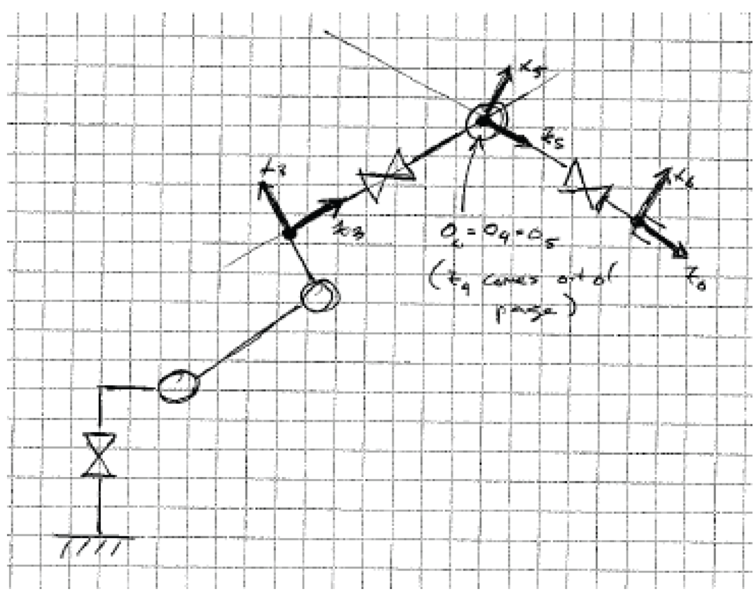
$$\gamma = \operatorname{atan2}(\ell_2 s_2, \ell_1 + \ell_2 c_2)$$

$$\boxed{\implies \quad \theta_1 = \operatorname{atan2}(y, x) - \operatorname{atan2}(\ell_2 s_2, \ell_1 + \ell_2 c_2)}$$

8.3 Kinematic Decoupling

It turns out that for some manipulators, the inverse kinematic problem can be decoupled into a positioning problem and an orientation problem. Two 3 DOF problems are much simpler than one 6 DOF problems!

Specifically, kinematic decoupling works for manipulators that have the last three joint axes intersect at a point, i.e., manipulators that have a *spherical wrist*. Because of this, most 6 DOF manipulators have spherical wrists! Recall the Denso arm:



8.3.1 Setting up the Math

Suppose the object is to position the end effector frame (aka frame 6, aka the "tool frame") to have position

$$o = \begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix}$$

and orientation R. Both are specified with respect to frame 0.

Denote the position of the point where the last three axes intersect as o_c (in the Denso case, $o_c = o_4 = o_5$.)

Note that

$$o_c^6 = \begin{bmatrix} 0 \\ 0 \\ -d_6 \end{bmatrix}$$

which means

$$\begin{bmatrix} o_c^0 \\ 1 \end{bmatrix} = H_6^0 \begin{bmatrix} o_c^6 \\ 1 \end{bmatrix} = \begin{bmatrix} R & o \\ 0\,0\,0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -d_6 \\ 1 \end{bmatrix}$$

which means

$$o_c^0 = o - R \begin{bmatrix} 0 \\ 0 \\ d_6 \end{bmatrix} \tag{5}$$

Let make this really explict. First, denote

$$o_c^0 = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$$

Now recall that from DH we can get H_4^0, which equals

$$H_4^0 = \begin{bmatrix} & x_c \\ R_4^0 & y_c \\ & z_c \\ 0\,0\,0 & 1 \end{bmatrix}$$

Also, note that x_c, y_c, and z_c only depend on θ_1, θ_2, and θ_3. So we can explicitly write out Equation **??**:

$$\boxed{\begin{bmatrix} x_c(\theta_1, \theta_2, \theta_3) \\ y_c(\theta_1, \theta_2, \theta_3) \\ z_c(\theta_1, \theta_2, \theta_3) \end{bmatrix} = \begin{bmatrix} o_x - r_{13}d_6 \\ o_y - r_{23}d_6 \\ o_z - r_{33}d_6 \end{bmatrix}}$$

So we have three equations, three unknowns, we can (using lots of tricks and magic) solve for θ_1, θ_2, and θ_3.

Now we choose the last three angles to make $R = R_6^0 = R_3^0 R_6^3$.

R_3^0 is given by θ_1, θ_2, and θ_3, which are now known.

R_6^3 is determined by θ_4, θ_5, and θ_6, which are unknown.

So we need to solve

$$R_6^3(\theta_4, \theta_5, \theta_6) = (R_3^0)^{-1} R.$$

Note from the drawing that θ_4, θ_5, and θ_6 represent subsequent current frame rotations about the z, y, and z axes, repsectively. In other words, they are the three Euler angles:

$$\theta_4 = \phi$$

$$\theta_5 = \theta$$

$$\theta_6 = \psi$$

So we can find them using Equations on page 55-56 in the Spong book.