Kinematic Diagram

— Shows how the links and joints are connected together, when all of the joint variables have a value of 0.

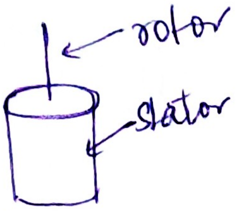
Manipulator = Robotic Arm

→ joints + Links

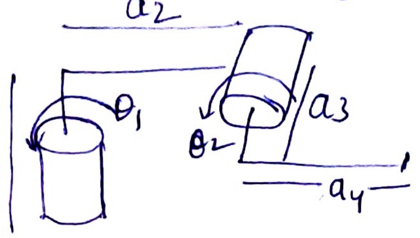
(a manipulator does not always look like a human arm; it consists of a number of joints connect by links)

joints: parts of the robot that allow motion

links: Parts that connect joints together

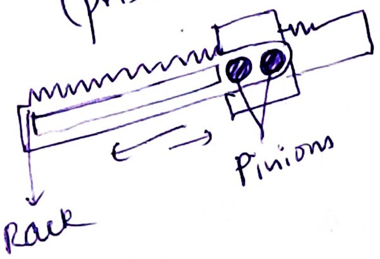


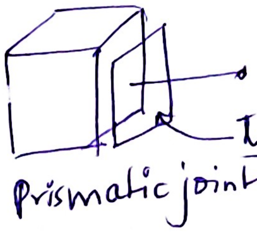
(Revolute joint)



workspace ⇒ the area or volume the end-effector can reach.

Rack & Pinion = a common means to convert rotation motion → linear motion (prismatic joint)

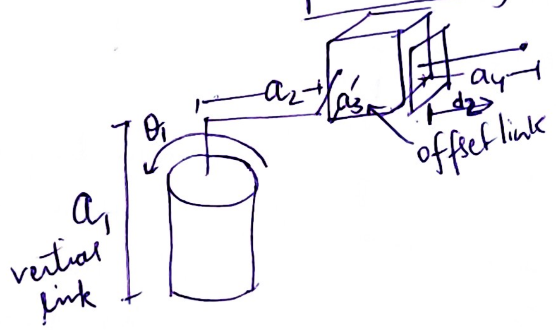




(moves left & right)

This square shows the direction that the prismatic joint is moving

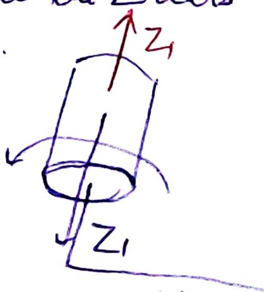
position direction of a revolute joint is obtained through the right hand thumb rule.



Joint variable — the value that changes when a joint moves.

Denavit-Hartenberg Rules to ~~define fram~~ assign frames

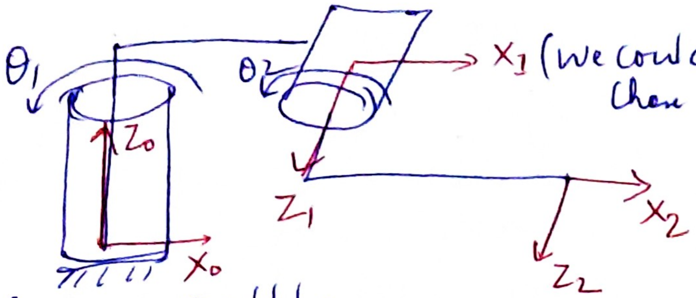
Rule #1: The Z axis must be the axis of rotation for a revolute joint, or the direction of motion for a prismatic joint, (If there is no axis of rotation or direction of motion on the end effector (since there is no joint), then we can follow the first rule. In that case, we can place the Z axis being in the same direction as the previous joint).



(we can place Z_1 in either of the dirs)

Rule #2: The X axis must be perpendicular both to its own Z axis, and the Z axis of the frame before it.

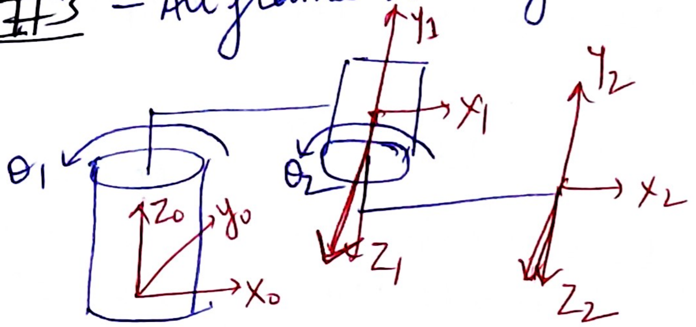
Scanned with ACE Scanner

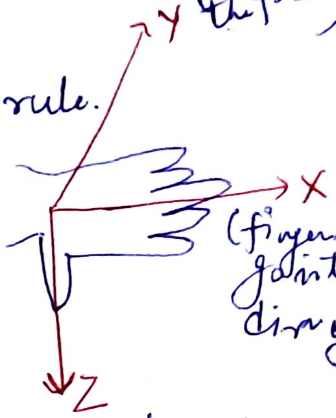


(We could drive also then X_1 to ~~[illegible]~~ go to the left)

(Note - we could have chosen X_0 axis into/out of the page or left side. It's ~~[illegible]~~ an arbitrary choice. We chose it to be in the right dirn)

Rule #3 - All frames must follow the right-hand rule.





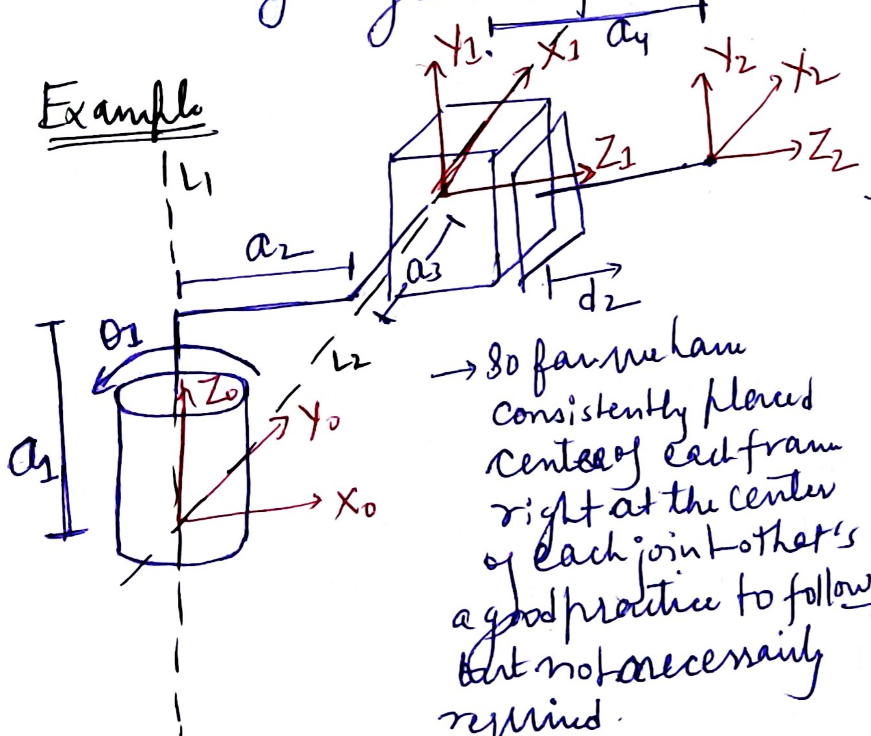
(Palm indicates the y-dirn)

(fingers go in the dirn of x)

(thumb goes in the Z direction)

Rule #4 - Each X axis must intersect the Z axis of the frame before it.

Example



→ So far we have consistently placed center of each frame right at the center of each joint. That's a good practice to follow but not necessarily required.

Checking if Rule #4 holds for joints 0 & 1 (dashed line)

→ If we extend a line through X_1 and Z_0. We expect them to intersect each other. At first glance, it may look like these lines intersect each other but it's not true. It looks like that because we're drawing a 3D object on 2D paper.

→ In 3D, we can see that the dashed line through X_1 does not intersect dashed line through Z_0. They pass by each other.

→ So rule #4 is not followed here.

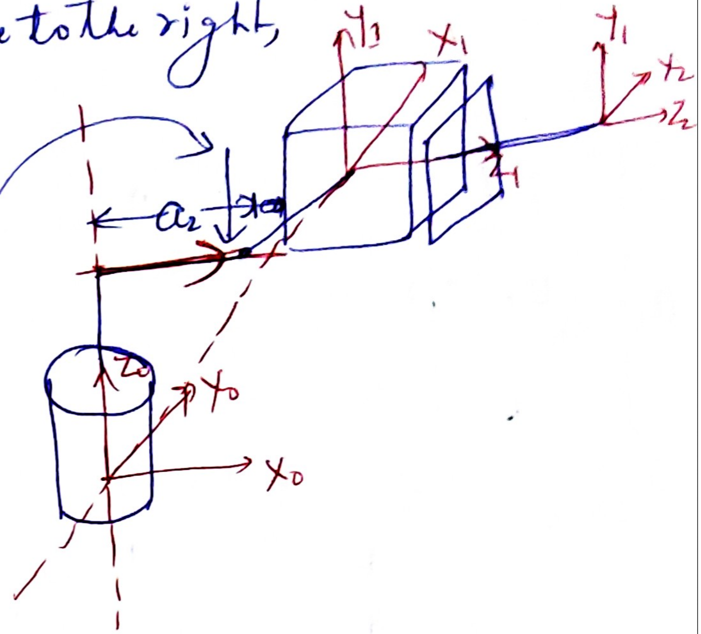
→ In order to follow rule #4, we have to move the center of frame #1.

→ In above case we must move the center of frame 1 in order to obey Rule #4.

→ We can move the center of the frame to any other location that we want.

B) Which dir should we move the center of the frame? And, how far should we move it?

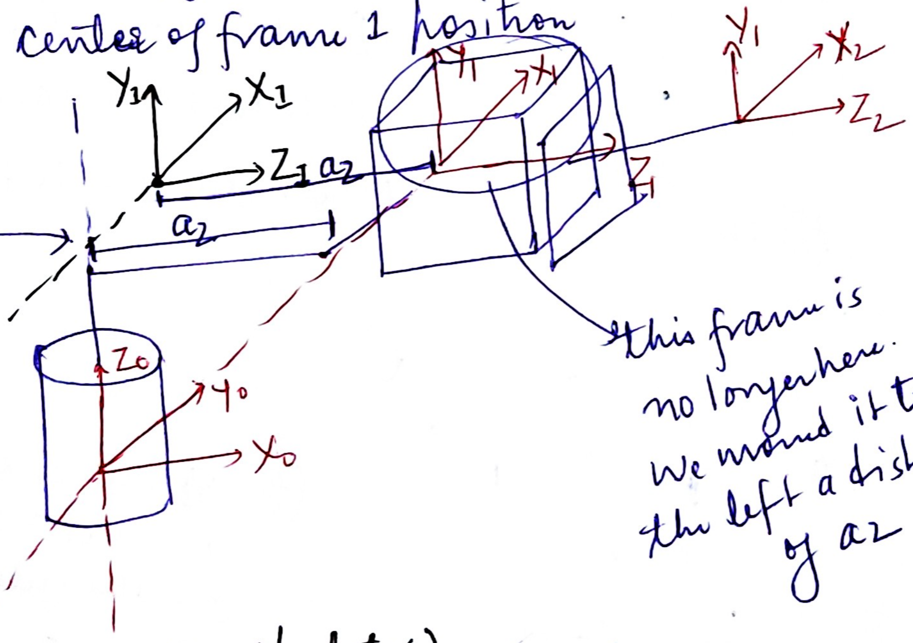
→ Since X_1 line passes by the Z_0 line to the right, that means we are going to need to need the move the center of frame 1 to the left.



How far do we move it?

X_1 line passes by the Z_0 line at. It passes by a distance of a_2. So, in order to get the X_1 line to intersect the Z_0 line, we need to move the center of frame 1 to the left a distance of a_2.

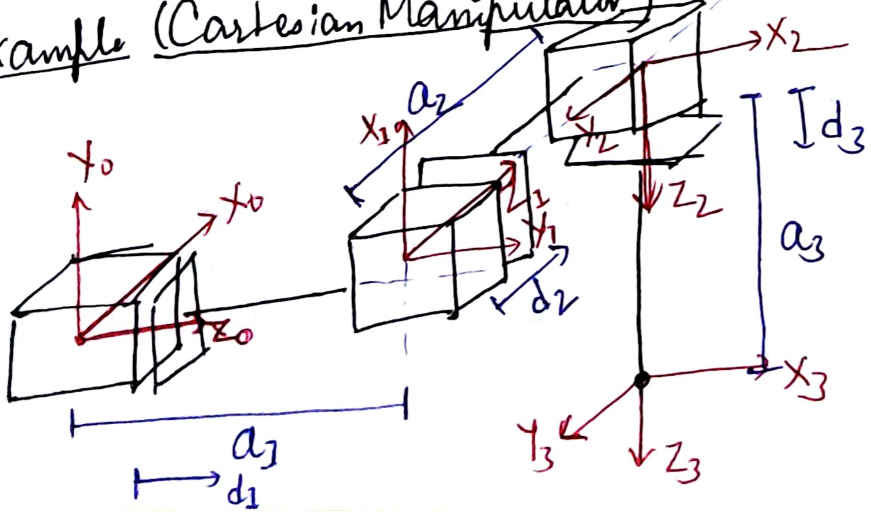
So, new center of frame 1 position



now X_1 intersects Z_0. hence, it follows rule #4

this frame is no longer here. we moved it to the left a distance of a_2.

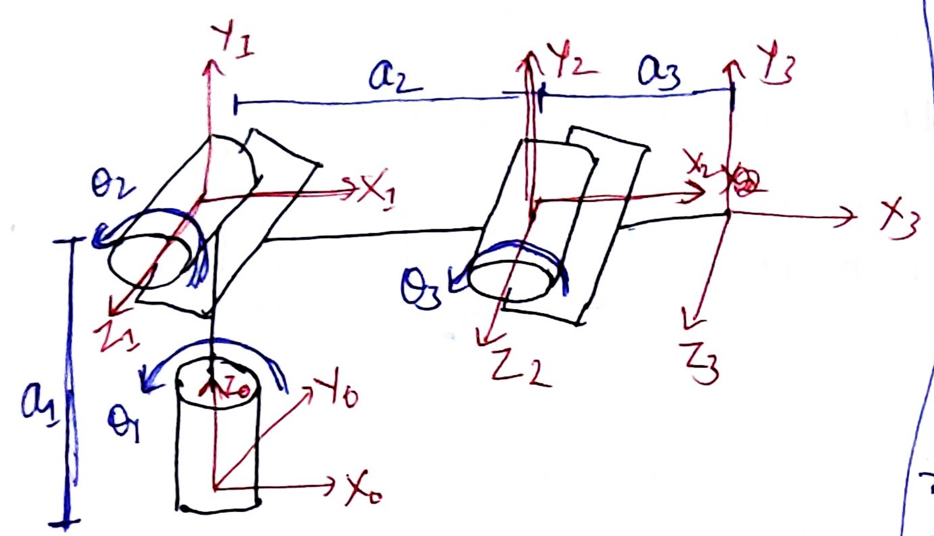
Example (Cartesian Manipulator)



Rule #4 is satisfied in case of all the frames.

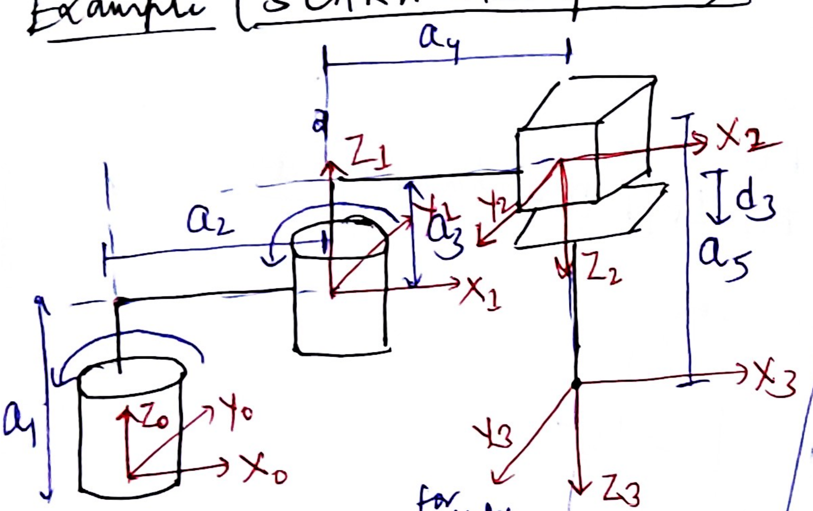
Scanned with ACE Scanner

Example (Articulated Manipulator)



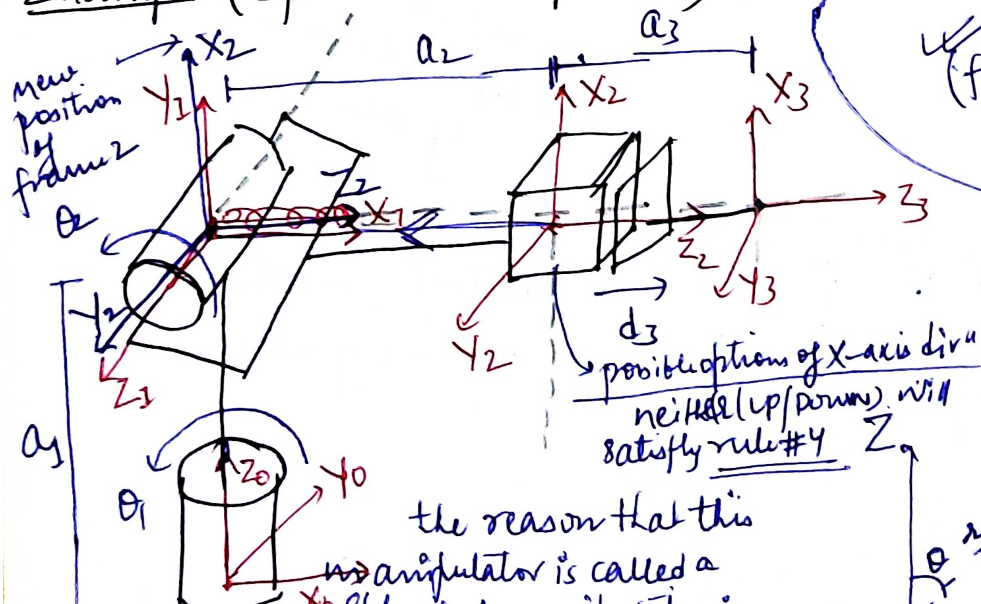
the way we solve below issue is we look at the direction from Z1 to the axis X2. The direction between these two axes is from the right to the left. So, in order to get these axes to intersect we have to move frame 2 to the left. Distance between the two frame is a2. So, we have to slide the entire frame 2 to the left a distance of a2. Now, rule #4 is followed.

Example (SCARA manipulator)



Rule#4 is followed for all the frames.

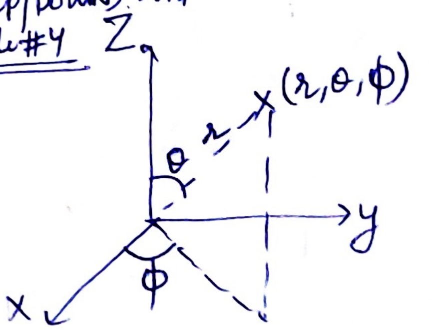
Example (Spherical Manipulator)



(for frame 2, X2 doesnot intersect Z1, hence violates rule #4)

possible options of X-axis dir'n neither (up/down) will satisfy rule#4

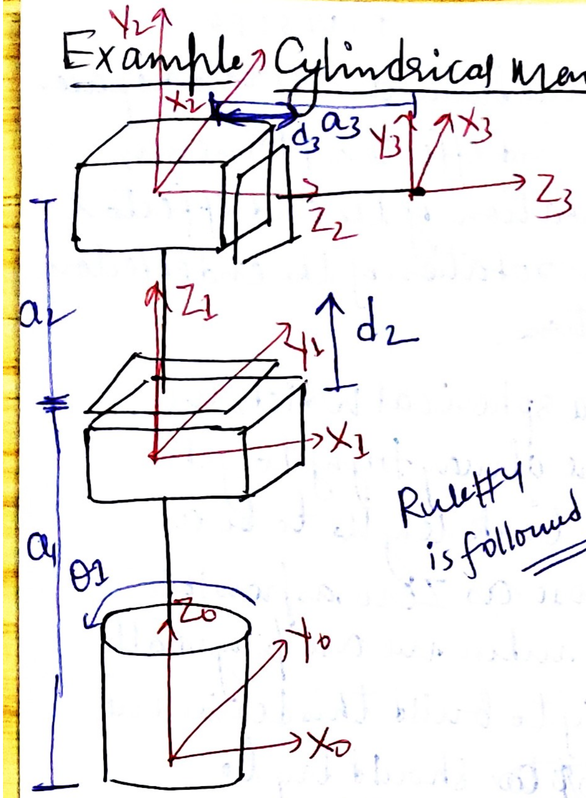
the reason that this manipulator is called a Spherical manipulator is because it uses the Spherical Coordinate system to position the end effector



radial distance r, polar angle θ, azimuthal angle ϕ (phi)

ρ (rho) is often used instead of r.

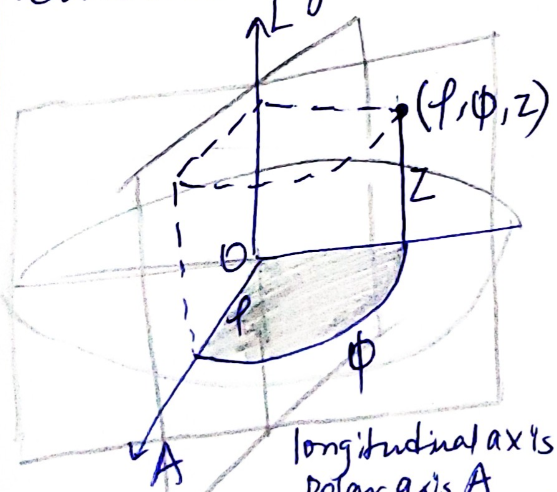
Example: Cylindrical Manipulator



Rule#4 is followed.

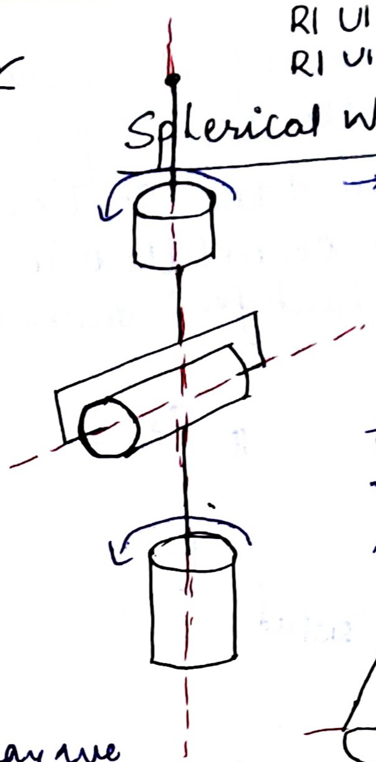
(one revolute joint is followed by two prismatic joint, one of whose motion is along the dirn of the axis of rotation of the first revolute joint while the second prismatic joint moves ⊥r to the first one)

→ Above manipulator is called a cylindrical manipulator because it works on the same principle as a cylindrical coordinate system.



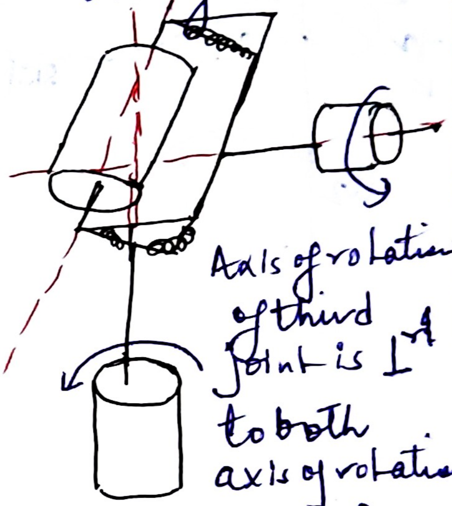
longitudinal axis L
polar axis A
radial distance ρ
angular coordinate ϕ
height Z

Spherical Wrist



→ In this case, axis of rotation of the third joint is collinear with the axis of rotation of the first joint but this will not be always be true.

The way we drew the kinematic diagram above makes it look like the spherical wrist is kinematically redundant as if the rotation of first joint is same as the rotation of third joint.



Axis of rotation of third joint is ⊥r to both axis of rotation of first & second joint.

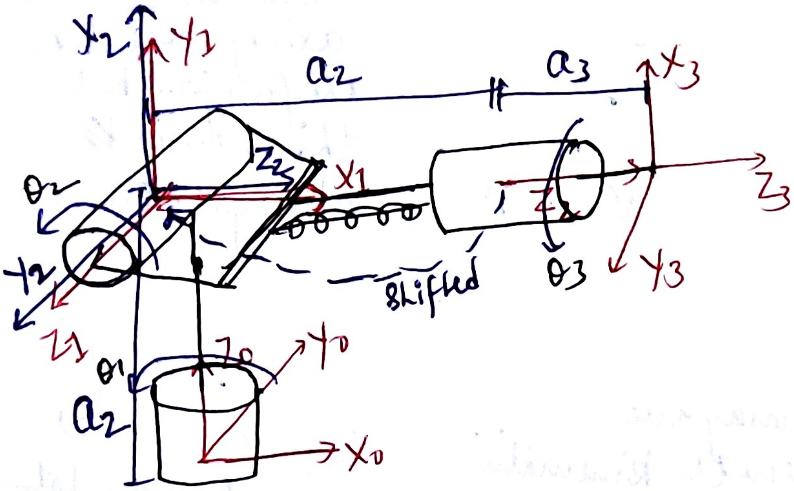
But, when we draw the kinematic diagram as shown in the second picture, we can see that this is not true. The rotation of first & 3rd joints are along different directions.

With this combination of three revolute joints we can rotate the end effector of a manipulator to any rotation we may choose.

In fact, that's the reason why we would add a spherical wrist onto the end of a manipulator.

With a three degree of freedom manipulator, we have a 3D workspace. We can choose the X, Y & Z position of the end effector. However, we cannot control the rotation or orientation of the end-effector.

→ The ability to control the orientation or rotation of the end effector is very important for industrial applications.



→ In a spherical wrist, we are always trying to get the link lengths to be as close as zero as possible.

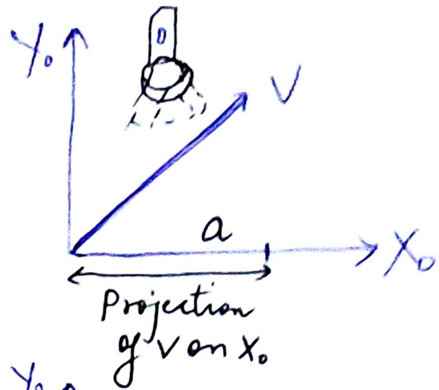
→ ~~So~~ when we are physically trying to build this device we ~~can~~ ~~to~~ should try to build it so that a_1, a_2 & a_3 are as small as possible.

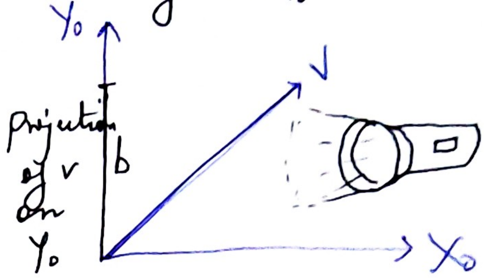
Scanned with ACE Scanner

Projection

	The Projection of v on:
The X axis	a
The Y axis	b

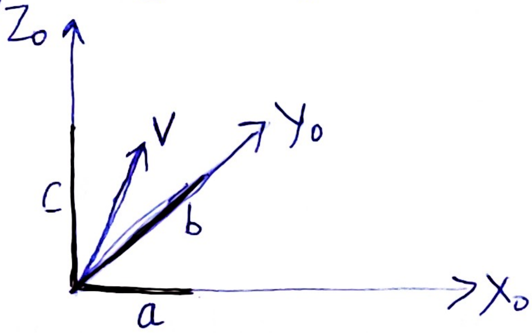
$\tan\theta = b/a$





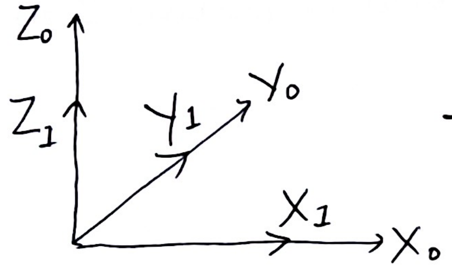
In robotics, we are generally working with 3D space, ~~but~~ not 2D.

	The proj. of v on:
The X axis	a
The Y axis	b
The Z axis	c

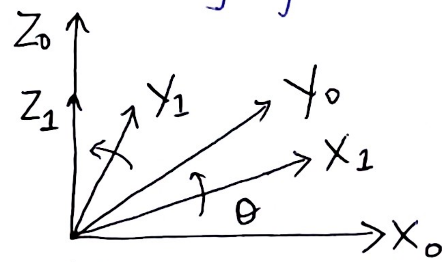


→ In robotics, we are not looking at ~~just~~ rotation of just one vector in 3D space instead, we are looking at the rotation of 1 frame inside another frame. In other words, we have 3 vectors rotating together i.e., X, Y & Z vectors.

e.g.



(Initially, Frame 1 is lined up exactly same as Frame 0)



(Frame 1 is rotated w.r.t. Frame 0)

How to represent rotation or orientation of frame 1 inside Frame 0?

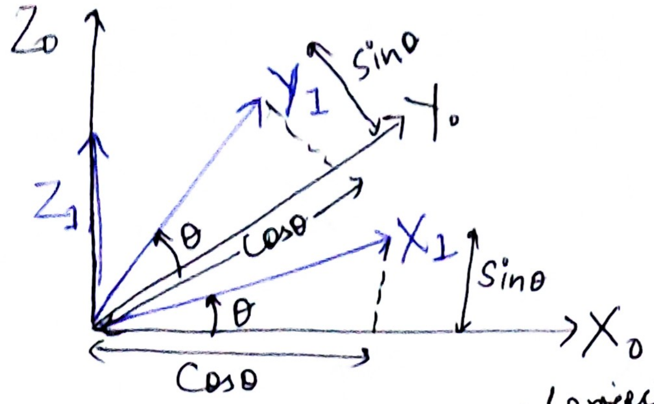
Ans —

The projection of:

$$R_1^0 = \begin{array}{c} \text{on:} \\ X_0 \\ Y_0 \\ Z_0 \end{array} \begin{array}{c} \begin{array}{ccc} X_1 & Y_1 & Z_1 \end{array} \\ \begin{bmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{bmatrix} \end{array}$$

Rotation Matrix

R_1^0 → how frame 1 is rotated inside of frame 0 or relative to frame 0



In coordinate frame, each of the axes is assumed to have a length of 1.

(projection of Y_1 on Y_0 is $-ve$ since it is in negative dir^n)

$$R_1^0 = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}_{3\times3}$$

$Z_1 \perp X_0$

$Z_1 \perp Y_0$

$Z_1 \parallel Z_0$ (Z_1 is lined up exactly with Z_0)

($X_1 \perp Z_0$ so the projection of X_1 on Z_0 is 0)

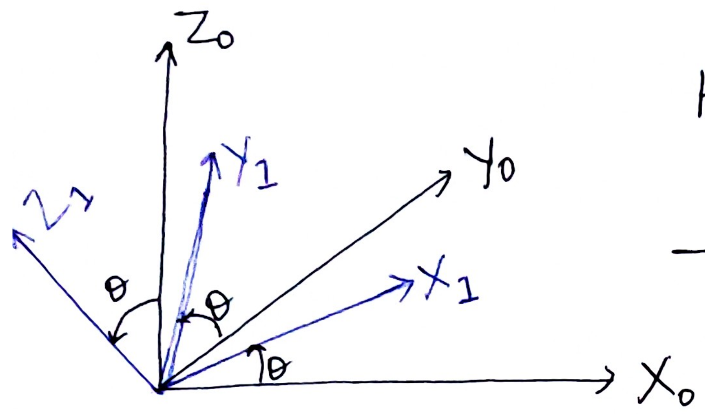
This matrix tells us all of the projections that we need for any angle θ we might use to rotate frame 1 in such a way that Z_1 does not change (rotation around Z-axis).

"Z Rotation Matrix"

$$R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$R_Y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

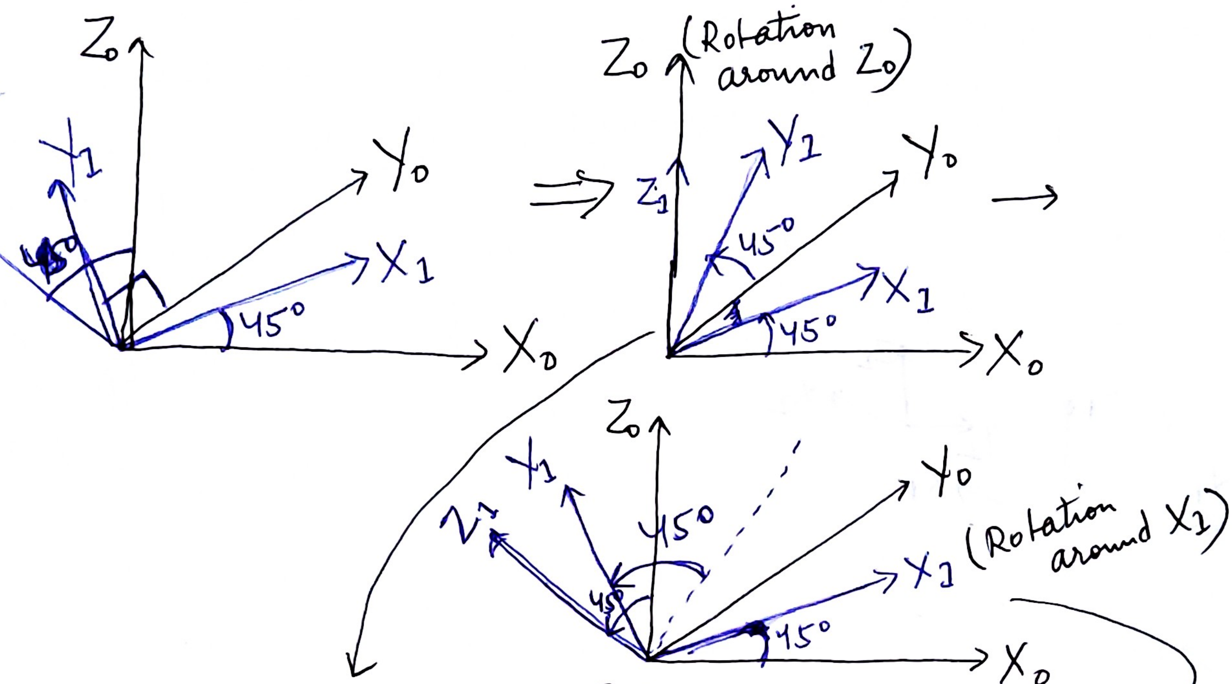
$$R_Z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



How do we write a rotation matrix for some arbitrary rotation?

→ ~~It~~ Any rotation that we can dreamup is some combination of rotation around X, Y and Z.

eg.

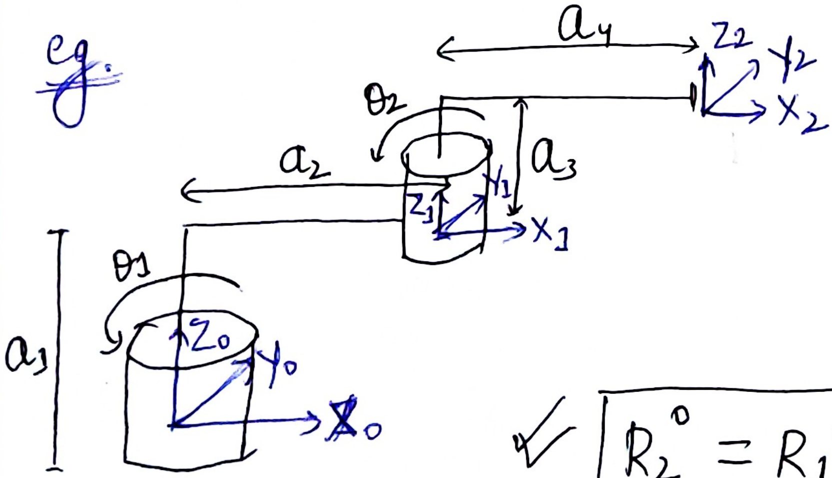


So, $R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$

(Projection of X1, Y1 & Z1 on X0, Y0 & Z0)

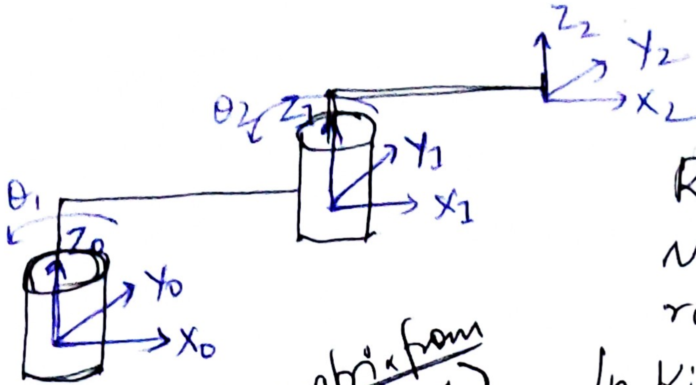
(rotation around Z) (rotation around X)

eg.



Find rotation matrix representing the rotation of end-effector frame relative to the base frame:

✓ $\boxed{R_2^0 = R_1^0 R_2^1}$



Rotation of frame 1 relative to frame 0 when the joint variable is 0 is no rotation at all.

In Kinematic dir^n, X_1, Y_1 & Z_1 are all in exactly the same dir^n as X_0, Y_0 & Z_0

There is a special matrix that means no rotation $\Longrightarrow$ Identity matrix

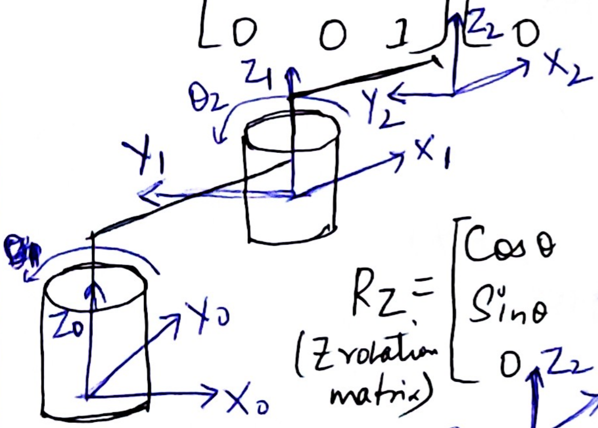
$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(proj. of X_1 on X_0 is 1, Y_1 on Y_0 is 1, Z_1 on Z_0 is 1)

(Complete rotation matrix from frame 0 to frame 1)

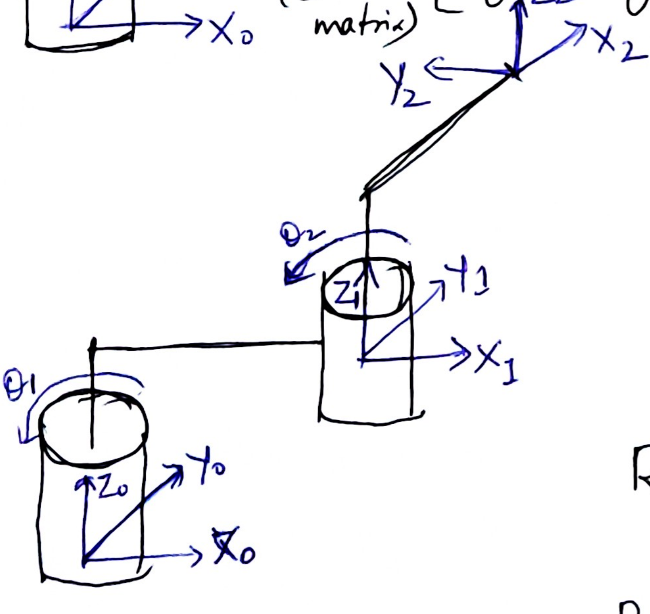
how frame 1 is rotated relative to 0 when joint angle is 0. (left multiply)

$$R_1^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$R_Z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(Z rotation matrix)



Rotation matrix from frame 1 to frame

$$R_Z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_2^1 = I \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

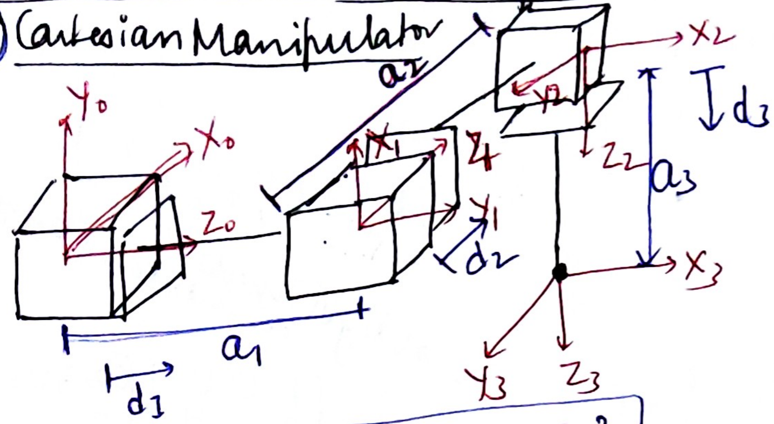
$$R_2^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(how frame 2 is rotated w.r.t. frame 1 before joint variable moves at all)

$$\boxed{R_2^0 = R_1^0 R_2^1}$$

Rotation Matrix Examples

(1) Cartesian Manipulator



$R_3^0 = ?$ $\boxed{R_3^0 = R_1^0 R_2^1 R_3^2}$

Shortcut – what we know about the meaning of a rotation matrix as the projection of the rotated frame on the original frame

$$R_1^0 = \begin{array}{c|ccc} & X_1 & Y_1 & Z_1 \\ \hline X_0 & & & \\ Y_0 & & & \\ Z_0 & & & \end{array}$$

Projection of frame 1

When joint var is 0 on frame 0 (original frame)

$$R_1^0 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Proj. of x_1 on y_0

|| tells us how frame 1 is rotated relative to frame 0 when joint variable is 0.

Identity matrix = means no rotation

second part we need for rotation matrix is a matrix that represents the rotation due to joint variable. (In other words, what happens to these projections as the joint variable moves)

In our case the joint is a prismatic joint that means no matter how this prismatic joint moves it won't cause any rotation at all between frame 0 & frame 1.

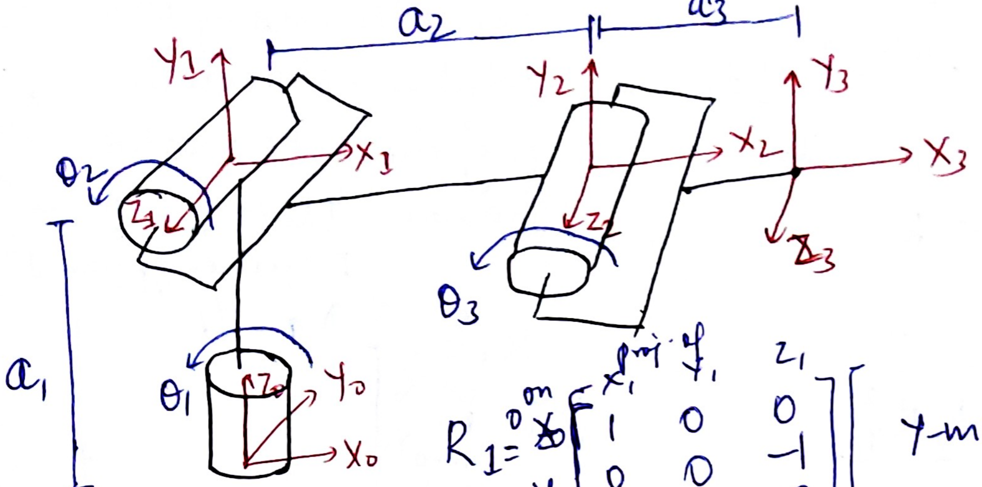
$$R_2^1 = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

(no rotation, hence X by I)

$$R_3^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

So, $R_3^0 = R_1^0 R_2^1 R_3^2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \_\_\_$

Example – Articulated Manipulator



$R_3^0 = ?$

WKT

$$\boxed{R_3^0 = R_1^0 R_2^1 R_3^2}$$

Using (ii) method described bottom of the page.

$$R_1^0 = \underbrace{\begin{array}{c|ccc} \text{proj. of} & x_1 & y_1 & z_1 \\ \hline x_0 & 1 & 0 & 0 \\ y_0 & 0 & 0 & -1 \\ z_0 & 0 & 1 & 0 \end{array}}_{\text{when joint variable is 0}} \underbrace{\begin{bmatrix} \\ \text{Y-matrix} \\ \\ \end{bmatrix}}_{\text{Second part represents rotation due to joint variable.}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 \\ 0 & 1 & 0 \\ -\sin\theta_1 & 0 & \cos\theta_1 \end{bmatrix}$$

(A) $\therefore$
$$R_1^0 = \begin{array}{c|ccc} & x_1 & y_1 & z_1 \\ \hline x_0 & C\theta_1 & 0 & S\theta_1 \\ y_0 & S\theta_1 & 0 & -C\theta_1 \\ z_0 & 0 & 1 & 0 \end{array}$$

(above matrix is telling us that Y_1 axis is always projected exactly onto the Z_0 axis)

Note – We have to be careful here about which axis we say the rotation is around due to the joint variable 'θ_1'.
θ_1 is a rotation around Z_0 but it's also a rotation around Y_1. So, which rotation should we use? The Z form of the matrix or Y form of the matrix?

Ans – We could use either one, but it matters where we place the matrix depending upon which we use.

(i) If we want to say the rotation is around Z_0 axis then we have to put Z matrix like below – (before the above matrix)

$$R_1^0 = \begin{bmatrix} \\ \text{Z-matrix} \\ \\ \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

(ii) If we want to say the rotation is around Y_1 axis then we have to put Y matrix after the calculated matrix like below –

$$R_1^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \\ \text{Y-matrix} \\ \\ \end{bmatrix}$$

$$R_2^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 \\ S\theta_2 & C\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

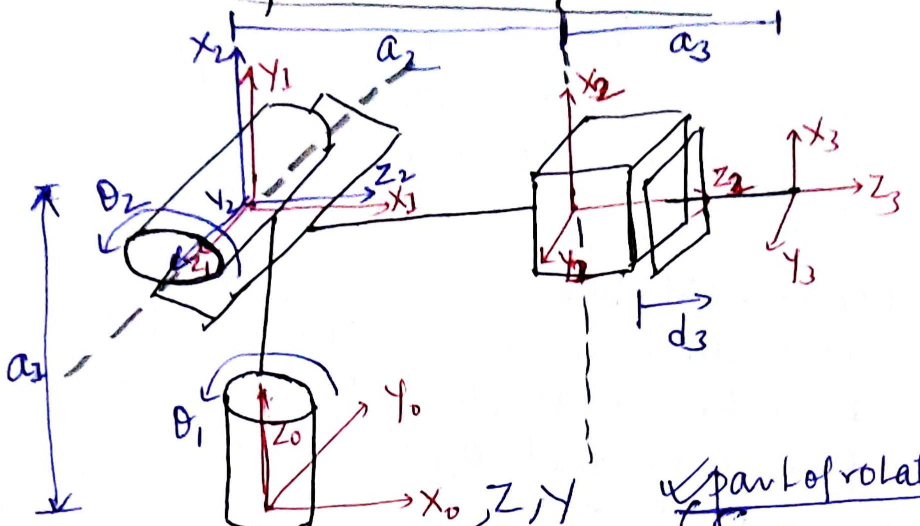
(Let's say we choose rotation around Z_2 axis)

(θ_2 is a rotation around both Z_1 & Z_2, so it does not matter if we multiply second matrix before or after.) or on the right or left. (We get the same answer)

$$R_3^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} C\theta_3 & -S\theta_3 & 0 \\ S\theta_3 & C\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{(rot around } Z_3)} = \begin{bmatrix} C\theta_3 & -S\theta_3 & 0 \\ S\theta_3 & C\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_3^0 = R_1^0 R_2^1 R_3^2 \;\checkmark$$

Example – Spherical Manipulator



X_2 axis doesnot intersect Z_1, thereby violating rule # 4 of DH rules, so we shifted frame 2 to the left (of 2nd joint-prismatic) by a distance of a_2. Now, frame 1 & 2 overlaps hence, following rule #2.

part of rotation matrix due to joint variable.

θ_1 is a rotation around Z_0 or it's a rotation around Y_1. If we want to use the rotation matrix around Z_0 it has to go on the left and if use the rotation around Y_1 it has to go on the right

$$[\text{left}]\begin{bmatrix}1&0&0\\0&0&-1\\0&1&0\end{bmatrix}[\text{right}]$$

$$R_0^1 = \begin{bmatrix}1&0&0\\0&0&-1\\0&1&0\end{bmatrix}$$

(hence frame 1 is rotated relative to frame 0 now before θ_1 has moved at all)

Using the rotation around Z_0, we have

$$R_0^1 = \begin{bmatrix}C\theta_1&-S\theta_1&0\\S\theta_1&C\theta_1&0\\0&0&1\end{bmatrix}\begin{bmatrix}1&0&0\\0&0&-1\\0&1&0\end{bmatrix} = \begin{bmatrix}C\theta_1&0&S\theta_1\\S\theta_1&0&-C\theta_1\\0&1&0\end{bmatrix}$$

$$R_1^2 = \begin{bmatrix}C\theta_2&-S\theta_2&0\\S\theta_2&C\theta_2&0\\0&0&1\end{bmatrix}\begin{bmatrix}0&0&1\\1&0&0\\0&1&0\end{bmatrix} \text{ (Using rotation around } Z_1\text{)} = \begin{bmatrix}-S\theta_2&0&C\theta_2\\C\theta_2&0&S\theta_2\\0&1&0\end{bmatrix}$$

both are same

or,

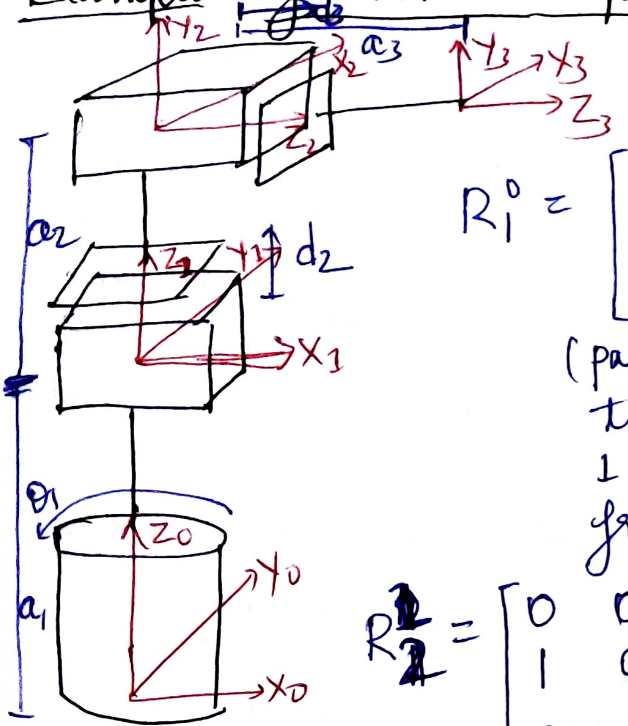
$$R_1^2 = \begin{bmatrix}0&0&1\\1&0&0\\0&1&0\end{bmatrix}\begin{bmatrix}C\theta_2&0&S\theta_2\\0&1&0\\-S\theta_2&0&C\theta_2\end{bmatrix} \text{ (Using standard rotation around } Y_2\text{)} = \begin{bmatrix}-S\theta_2&0&C\theta_2\\C\theta_2&0&S\theta_2\\0&1&0\end{bmatrix}$$

$$R_2^3 = \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} = \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

third joint is a prismatic joint & prismatic joint does not give us any rotation due to the joint variable. So we will use Identity matrix (which mean no rotation)

$$\therefore R_0^3 = R_0^1 R_1^2 R_2^3 = \begin{bmatrix}C\theta_1&0&S\theta_1\\S\theta_1&0&-C\theta_1\\0&1&0\end{bmatrix}\begin{bmatrix}-S\theta_2&0&C\theta_2\\C\theta_2&0&S\theta_2\\0&1&0\end{bmatrix}\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

Example - Cylindrical Manipulator



identity matrix since frame 0 & frame 1 are in the same orientation

$$R_1^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta & -S\theta_1 & 0 \\ S\theta_1 & C\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C\theta & -S\theta_1 & 0 \\ S\theta_1 & C\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(around Z_1)

(part of rotation matrix that tells us how frame 1 is rotated relative to frame 0 not due to θ_1)

Part of rotation matrix that is due to θ_1. (θ_1 is a rotation either around Z_0 or Z_1, In both cases, it's a rotation around Z, so we can either left multiply or right multiply by Z rotation matrix)

$$R_2^1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(prismatic joint, so no rotation, hence use Identity matrix as rotation matrix)

$$= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$R_3^2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

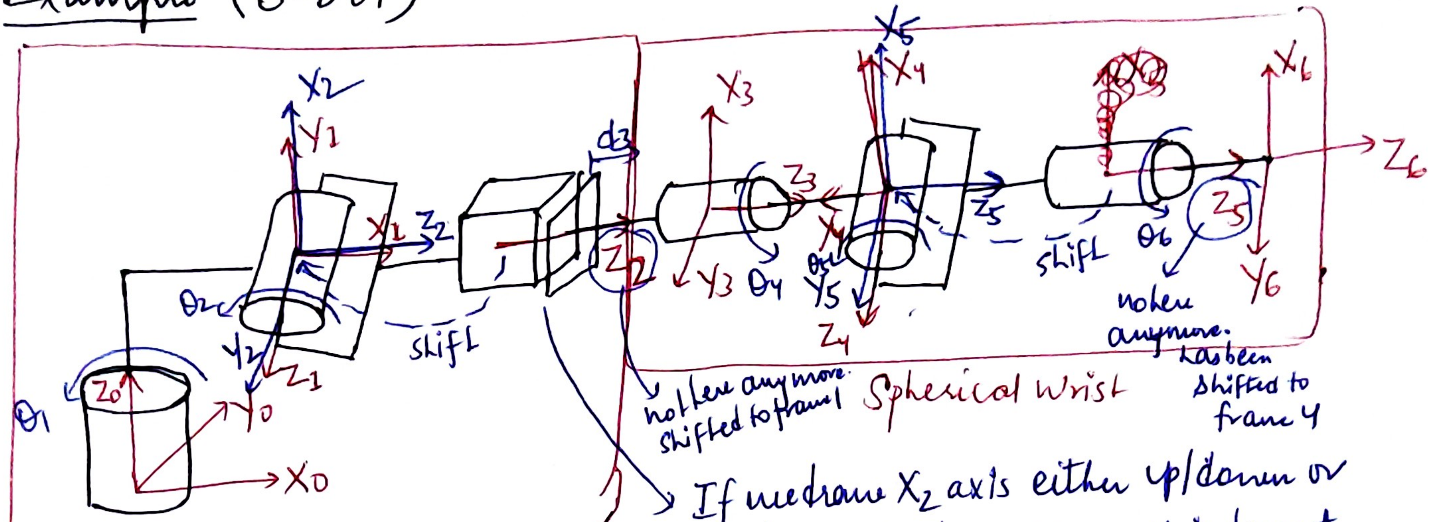
(Same logic as above)

$$\therefore R_3^0 = R_1^0 \, R_2^1 \, R_3^2$$

$$= \begin{bmatrix} C\theta & -S\theta_1 & 0 \\ S\theta_1 & C\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

(R1 U1 S3 P3) - Angela Sodemann

Example (6-DOF)



Spherical manipulator

If we draw X_2 axis either up/down or into/out of the page, it won't intersect Z_1. Hence, won't follow rule #4 of DH rules. therefore, we shift the location of center of frame 2 to the left

$$\boxed{R_6^0 = R_1^0 R_2^1 R_3^2 R_4^3 R_5^4 R_6^5}$$

$$R_1^0 = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 \\ s\theta_1 & c\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} c\theta_1 & 0 & s\theta_1 \\ s\theta_1 & 0 & -c\theta_1 \\ 0 & 1 & 0 \end{bmatrix}$$

(rot. around Z_0) &

$$R_2^1 = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 \\ s\theta_2 & c\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} -s\theta_2 & 0 & c\theta_2 \\ c\theta_2 & 0 & s\theta_2 \\ 0 & 1 & 0 \end{bmatrix}$$

(for rot. around Z_1) (how frame 2 is rot. relative to frame 1 not due to θ_2)

$$R_3^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I_{3\times3}$$

(no rot., so use Identity matrix)

$$R_4^3 = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 \\ s\theta_4 & c\theta_4 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} c\theta_4 & 0 & -s\theta_4 \\ s\theta_4 & 0 & c\theta_4 \\ 0 & -1 & 0 \end{bmatrix}$$

$$R_5^4 = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 \\ s\theta_5 & c\theta_5 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} c\theta_5 & 0 & s\theta_5 \\ s\theta_5 & 0 & -c\theta_5 \\ 0 & 1 & 0 \end{bmatrix}$$

$$R_6^5 = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 \\ s\theta_6 & c\theta_6 & 0 \\ 0 & 0 & 1 \end{bmatrix} I$$

(for rot. around Z_5)

(frame 6 is in exact same orientation as frame 5. This is often the case with most end effector and the frame before it.) ⇒ no rot. so use $I_{3\times3}$.

So, $R_6^0 = \begin{bmatrix} c\theta_1 & 0 & s\theta_1 \\ s\theta_1 & 0 & -c\theta_1 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} -s\theta_2 & 0 & c\theta_2 \\ c\theta_2 & 0 & s\theta_2 \\ 0 & 1 & 0 \end{bmatrix} I_{3\times3} \begin{bmatrix} c\theta_4 & 0 & -s\theta_4 \\ s\theta_4 & 0 & c\theta_4 \\ 0 & -1 & 0 \end{bmatrix}\begin{bmatrix} c\theta_5 & 0 & s\theta_5 \\ s\theta_5 & 0 & -c\theta_5 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} c\theta_6 & -s\theta_6 & 0 \\ s\theta_6 & c\theta_6 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$R^m_n = \begin{array}{c|ccc} \text{on:} & X_n & Y_n & Z_n \\ \hline X_m & \square & \square & \square \\ Y_m & \square & \square & \square \\ Z_m & \square & \square & \square \end{array}$$

the projection of:

(rotation matrix)

the rotation of frame n relative to frame m.

→ we use such matrix to figure out the rotation of the end effector frame relative to the base frame in our robot manipulator but rotation is not the only thing that happens to the end effector frame as the joint moves in our robot manipulator.

→ The other thing that happens is that the position of the end effector changes.

→ The rotation matrix does not express how the end effector position changes.

→ Instead, we use a different expression called the displacement vector.

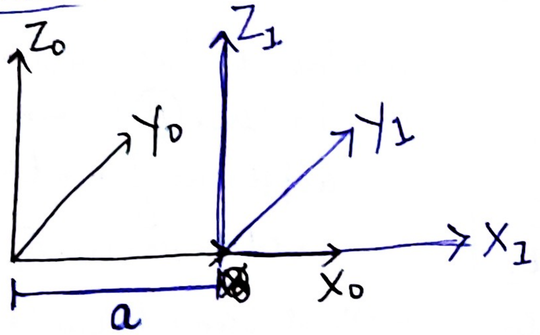
Displacement Vector

$$d^m_n = \begin{bmatrix} \square \\ \square \\ \square \end{bmatrix} \begin{array}{l} \leftarrow X^m_n \\ \leftarrow Y^m_n \\ \leftarrow Z^m_n \end{array}$$

(only one column & 3 rows)

The first row tells us the X position of the n-frame in the m frame, the second row tells us the Y position and the third row tells us the Z position.

Example

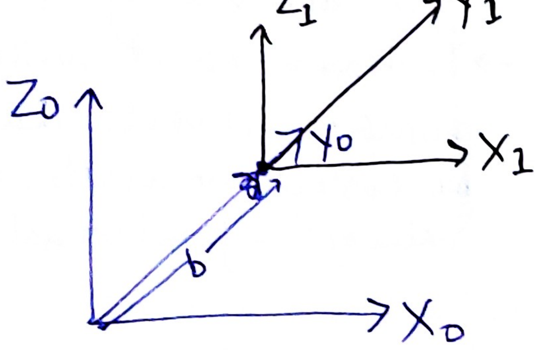


$$d_1^0 = \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix}$$

(the displacement is a distance of a in the X direction, 0 in the Y direction & 0 in the Z direction)

$R_1^0 = I_{3\times3}$ (frame 1 has the same orientation as frame 0, so no rotation)

Example

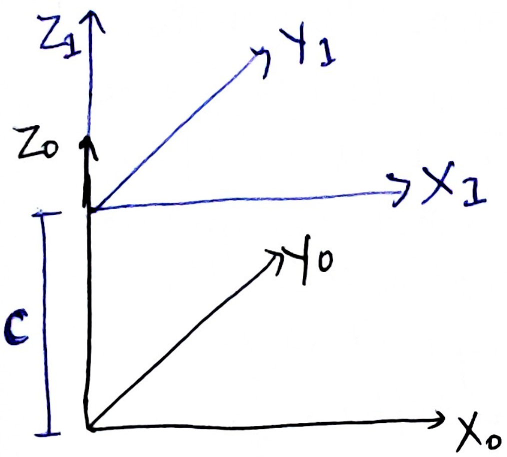


$$d_1^0 = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix}$$

(the displacement vector from 0 to 1 is 0 in the X direction, b in the Y direction & 0 in the Z direction.

$R_1^0 = I_{3\times3}$

Example



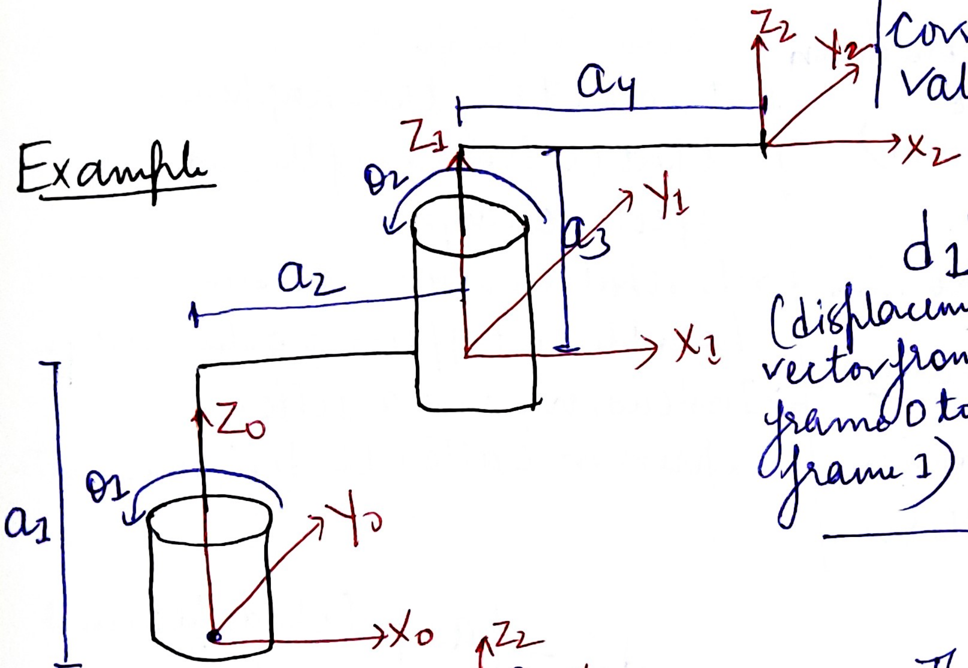
$$d_1^0 = \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}$$

(displacement vector is 0 in X dirn, 0 in Y dirn & C in Z dirn)

we can use the notation to represent the displacement between two frames and any combination of displacements in X, Y and Z directions.

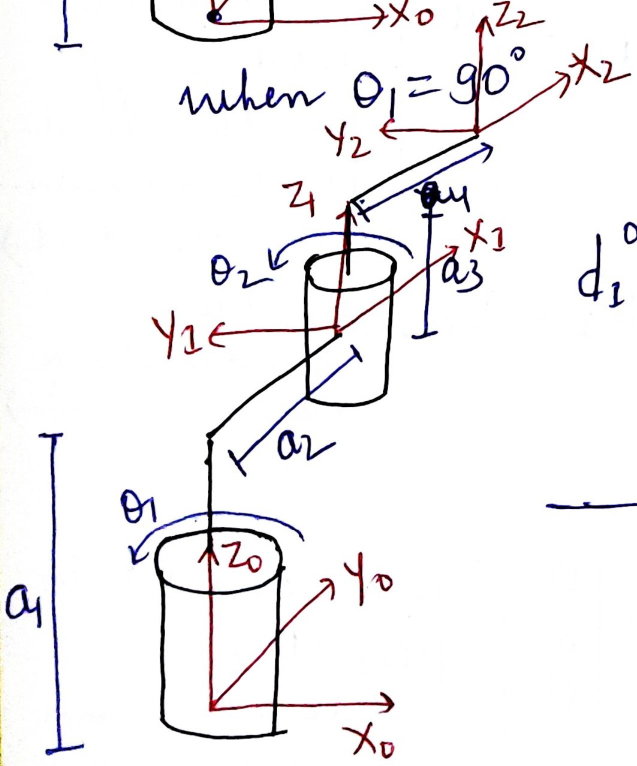
→ The trick to getting this right for our kinematic diagrams is that we need to make sure that our displacement vector is correct no matter what the values of the joint variables are.

Example



$$d_1^0 = \begin{bmatrix} a_2 \\ 0 \\ a_1 \end{bmatrix} \qquad \theta_1 = 0$$

(displacement vector from frame 0 to frame 1) —— (I)



$$d_1^0 = \begin{bmatrix} 0 \\ a_2 \\ a_1 \end{bmatrix}$$

—— (II)

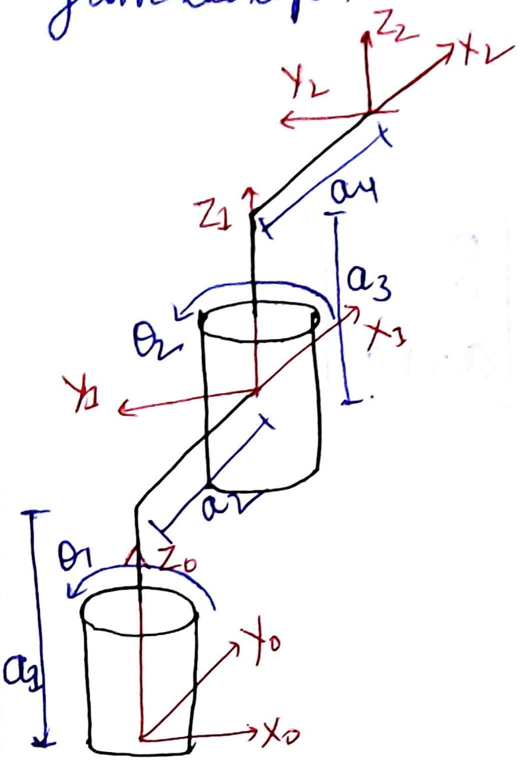
→ The displacement vector in case (II) (when $\theta_1 = 90°$) is different from the displacement vector in case (I) (when $\theta_1 = 0°$)

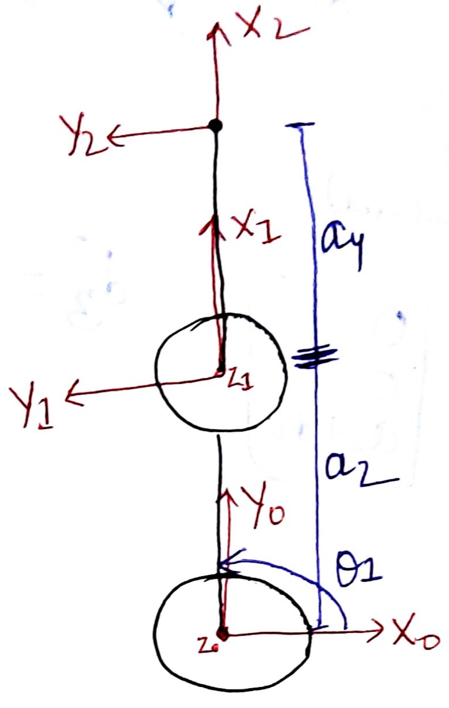
→ we can't have it be that the displacement vector is different for each value of θ_1.

→ Instead we have to write a displacement vector that will be correct no matter what the value of the joint variable is.

How to write displacement vector that is correct no matter what the value of joint variable θ is?

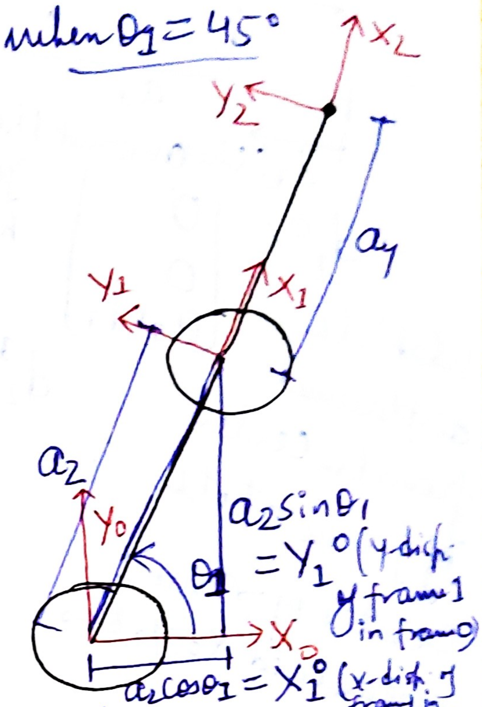
Let's draw the Kinematic diagram from the top down view as if we are looking down at it from the top.





$\theta_1 = 90°$

when $\theta_1 = 45°$



→ we are trying to find here what is the displacement vector between the center of frame 0 and center of frame 1.

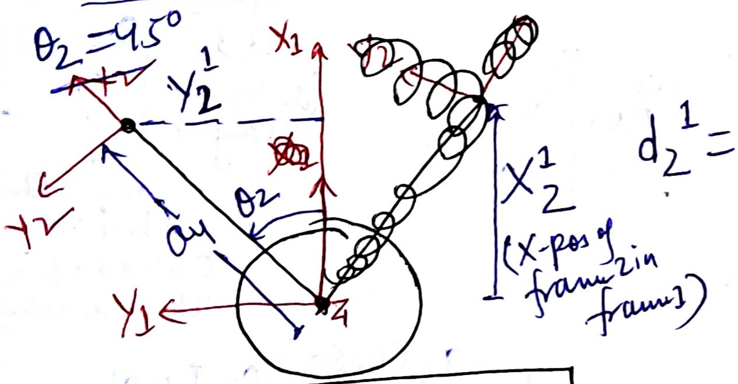
→ we want to get a displacement vector which would be correct for any value of θ_1

$$d_1^0 = \begin{bmatrix} a_2\cos\theta_1 \\ a_2\sin\theta_1 \\ a_1 \end{bmatrix}$$

we can get this (a_1) from the isometric view of the kinematic diagram on the extreme left.

Displacement of frame 2 inside frame 1

we draw top-view in such a way that

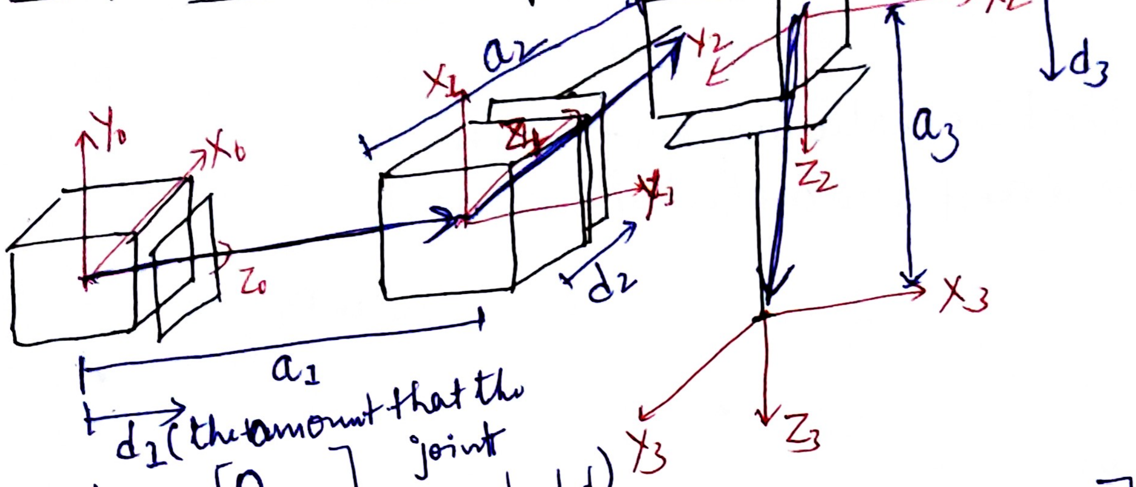


$$d_2^1 = \begin{bmatrix} a_4\cos\theta_2 \\ a_4\sin\theta_2 \\ a_3 \end{bmatrix}$$

$$\boxed{\begin{aligned} d_2^0 &\neq d_1^0 d_2^1 \\ d_2^0 &\neq d_1^0 + d_2^1 \end{aligned}}$$

✓ In order to link together the displacements, we have to lump them together with the rotation matrices inside of a new matrix called the Homogeneous transformation matrix.

Example (cartesian manipulator)



when we have a prismatic joint, the direction of the displacement never changes.

d_1 (the amount that the joint has extended)

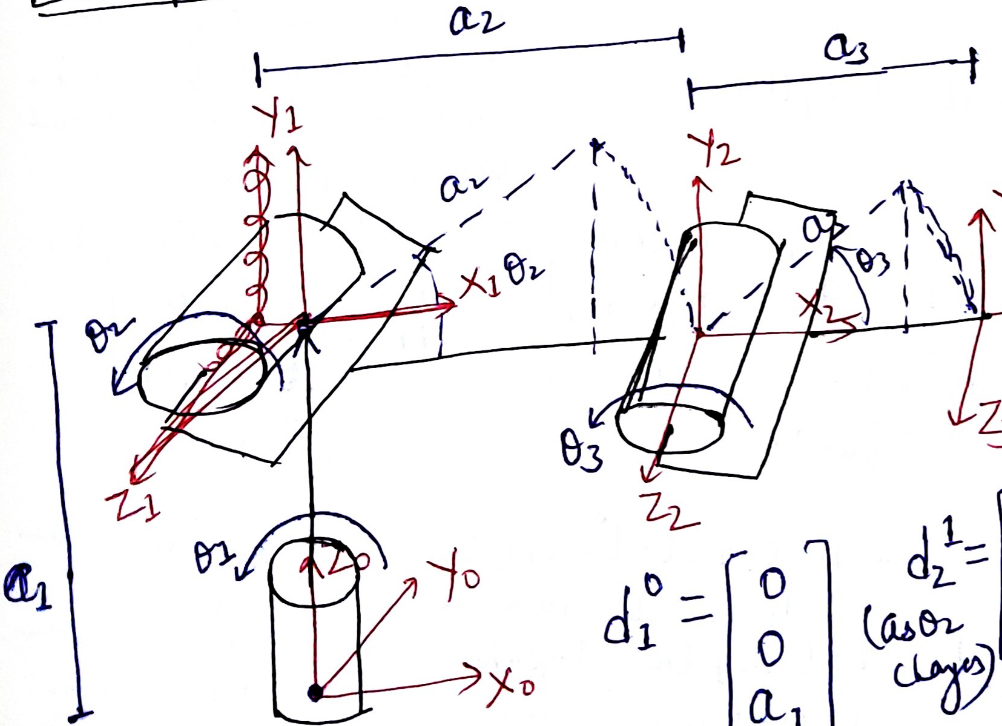
$$d_1^0 = \begin{bmatrix} 0 \\ 0 \\ a_1 + d_1 \end{bmatrix}$$

What is the displacement from the center of frame 0 to the center of frame 1 measured X_0, Y_0 & Z_0 directions.

$$d_2^1 = \begin{bmatrix} 0 \\ 0 \\ a_2 + d_2 \end{bmatrix}$$

$$d_3^2 = \begin{bmatrix} 0 \\ 0 \\ a_3 + d_3 \end{bmatrix}$$

Example – Articulated manipulator



$$d_3^2 = \begin{bmatrix} a_3 \cos\theta_3 \\ a_3 \sin\theta_3 \\ 0 \end{bmatrix}$$

(as θ_3 changes)

Since both the frames are lined up in the Z direction)

$$d_1^0 = \begin{bmatrix} 0 \\ 0 \\ a_1 \end{bmatrix}$$

$$d_2^1 = \begin{bmatrix} a_2 \cos\theta_2 \\ a_2 \sin\theta_2 \\ 0 \end{bmatrix}$$

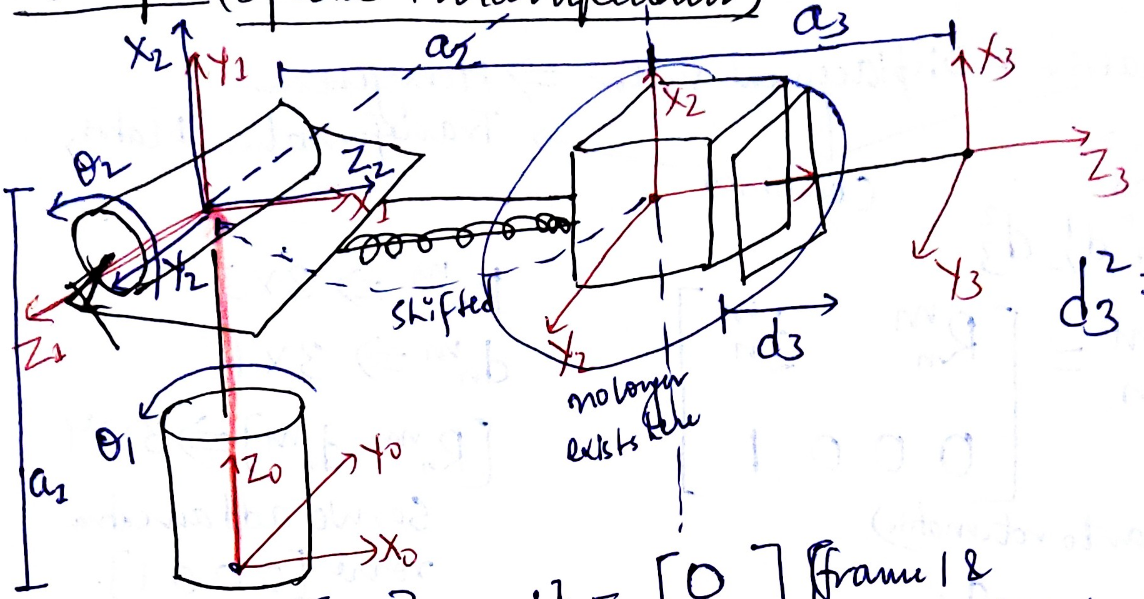
(as θ_2 changes)

(Centres of frame 1 & frame 2 is lined up in the Z direction, hence it won't change & will be 0)

(In this case, changing value of θ_1 will not change the displacement betn center of frame 0 & center of frame 1)

Scanned with ACE Scanner

Example (Spherical manipulator)



Since frame 2 & frame 3 are always lined up in X_2 & Y_2 dir.

$$d_3^2 = \begin{bmatrix} 0 \\ 0 \\ a_2+a_3+d_3 \end{bmatrix}$$

joint variable

$$d_1^0 = \begin{bmatrix} 0 \\ 0 \\ a_1 \end{bmatrix}$$

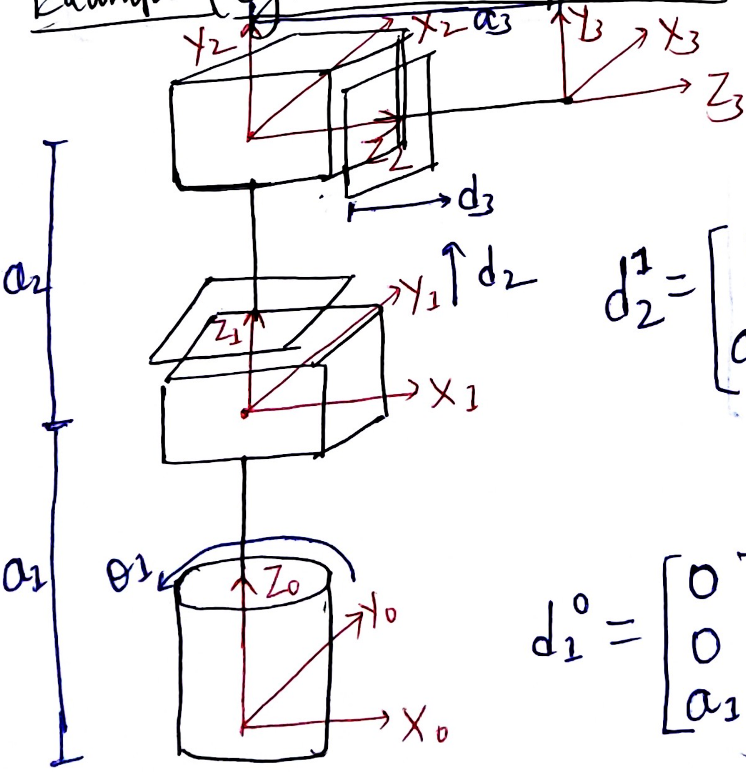
Since the center of frame 1 does not move through space as θ_1 changes we don't have a θ_1 showing up in our displacement vector.

$$d_2^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

(Use shifted frame 2, not the original one)

(frame 1 & frame 2 (shifted one) are directly on top of each other) So, even when we change the value of θ_2 frame 2 will rotate around its current location but won't move through space at all)

Example (Cylindrical Manipulator)



$$d_3^2 = \begin{bmatrix} 0 \\ 0 \\ a_3+d_3 \end{bmatrix}$$

$$d_2^1 = \begin{bmatrix} 0 \\ 0 \\ a_2+d_2 \end{bmatrix}$$

$$d_1^0 = \begin{bmatrix} 0 \\ 0 \\ a_1 \end{bmatrix}$$

Homogeneous Transformation Matrix

Rotation matrix + Displacement vector $\Rightarrow$ Homogeneous Transformation Matrix

Combined

$R_3^0 = R_1^0 R_2^1 R_3^2$

But, $d_3^0 \neq d_1^0 d_2^1 d_3^2$

$$H_n^m = \begin{bmatrix} R_n^m & d_n^m \\ 0\ 0\ 0 & 1 \end{bmatrix}$$

$R_n^m \Rightarrow 3\times3$

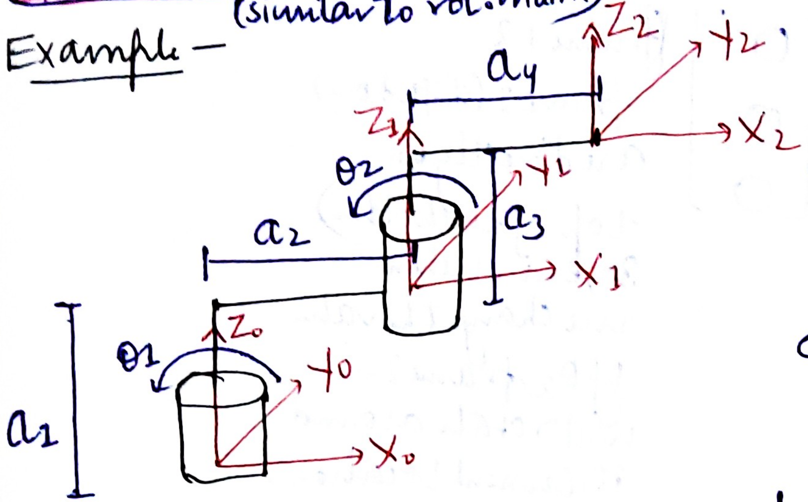
$d_n^m \Rightarrow 3\times1$

$[R_n^m \ \ d_n^m] \Rightarrow 3\times4$

So, we add an extra row $[0\ 0\ 0\ 1]$

☆ $\boxed{H_2^0 = H_1^0 H_2^1}$ (similar to rot. matrix)

Example –



$$R_1^0 = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 \\ S\theta_1 & C\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$d_1^0 = \begin{bmatrix} a_2C\theta_1 \\ a_2S\theta_1 \\ a_1 \end{bmatrix}$$

$$H_1^0 = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & a_2C\theta_1 \\ S\theta_1 & C\theta_1 & 0 & a_2S\theta_1 \\ 0 & 0 & 1 & a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_2^1 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 \\ S\theta_2 & C\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$d_2^1 = \begin{bmatrix} a_4C\theta_2 \\ a_4S\theta_2 \\ a_3 \end{bmatrix}$$

$$H_2^1 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & a_4C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & a_4S\theta_2 \\ 0 & 0 & 1 & a_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$H_2^0 = H_1^0 H_2^1$

$$H_2^0 = \begin{bmatrix} R_2^0 & d_2^0 \\ 0\ 0\ 0 & 1 \end{bmatrix}$$

No other way to get d_2^0 other than to calculate this homogeneous transformation matrix and then pull out d_2^0 from the place where it is in the matrix

We can get R_2^0 by multiplying the rotation matrices like

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

This is how we can tell the location of our robot's end effector for any combination of angles θ_1 & θ_2.

<u>Forward Kinematics</u> — When we are able to tell the location of the robot end-effector given the values of the joints.

(Another way to find Homogeneous Transformation matrix)

Dénavit-Hartenberg Method (DH Parameters)

— fast
— Industrial Standard

Step ① Assign frames according to the 4 Denavit-Hartenberg rules.

Step ② fill out the DH parameter table.

Number of rows = number of frames − 1

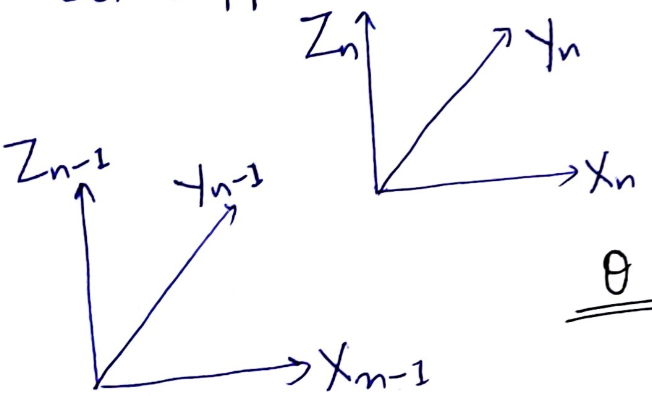
	θ	α	r	d

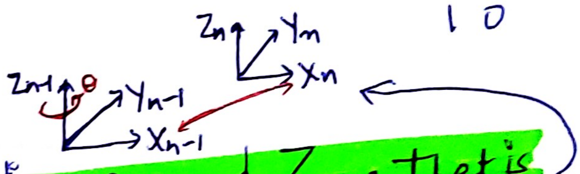
θ, α represent rotations

r, d represent linear distances.

Let suppose we have two frames, frame n & frame $(n-1)$.

Like frame 3 & frame 2
2 1
1 0

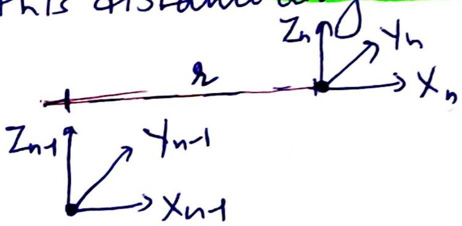




$\underline{\underline{\theta}}$ → rotation around Z_{n-1} that is required to get axis X_{n-1} to match axis X_n. Keep in mind that this has to include also any θ rotation that is due to the joint.

$\underline{\underline{r}}$ → distance between the center of the $(n-1)$ frame & the center of n frame.

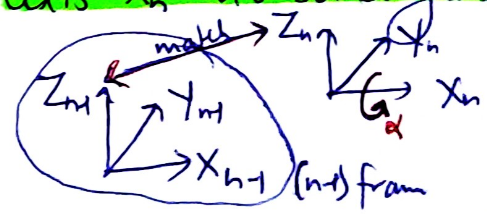
r parameter is defined as this distance along the X_n direction.



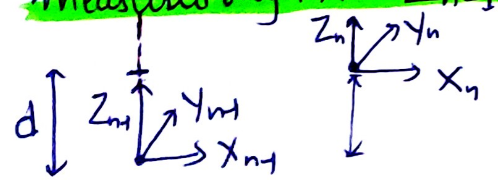
$\underline{\underline{\alpha}}$ → rotation around X_n that is required to get axis Z_{n-1} to match Z_n.

Even though the axis we are rotating around is the X_n axis the frame we are going to rotate is $(n-1)$ frame.

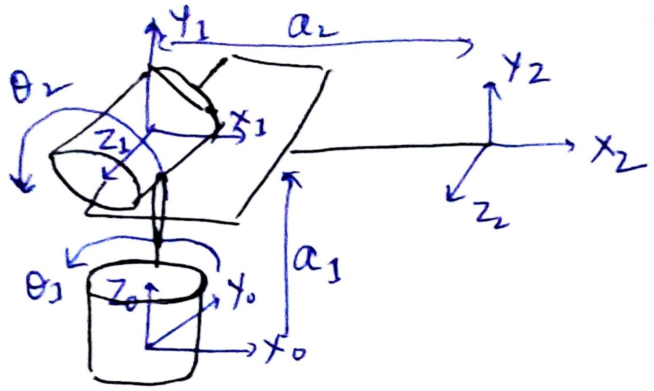
In other words, how do we have to rotate $(n-1)$ frame around the axis X_n in order to get the axis X_{n-1} match Z_n



$\underline{\underline{d}}$ → displacement from the center of frame $(n-1)$ to the center of frame n measured only in the Z_{n-1} direction.



Example - 2-dof manipulator



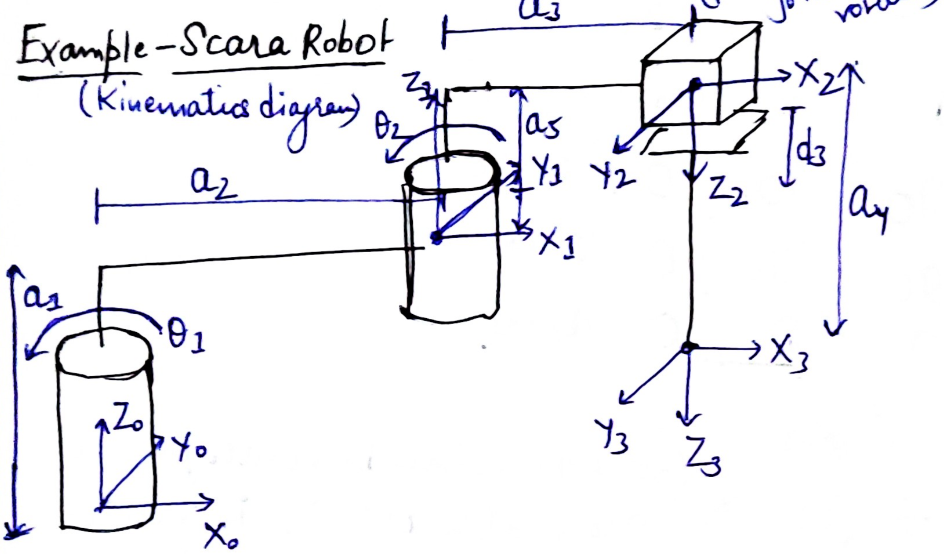
rot. around Z_0 that is req. to get axis X_0 to match axis X_1.

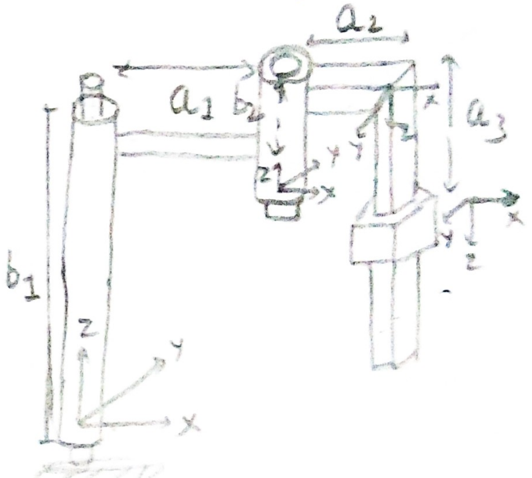
n	θ	α	r	d
1	$0+\theta_1=\theta_1$	90°	0	a_1
2	θ_2	0	a_2	0

how to rotate frame 1 around axis X_2 to get axis X_1

what is the distance between centre of frame 0 to centre of frame 1 in the Z_0 dir^n?

Example - Scara Robot
(Kinematics diagram)





Parameter Table

n	θ	α	r	d
1	θ_1	0	a_2	a_1
2	θ_2	180	a_3	a_5
3	0	0	0	a_4+d_3

Homogeneous transformation matrix

Rotation matrix

displacement vector

$$H_n^{n-1} = \begin{bmatrix} C\theta_n & -S\theta_n C\alpha_n & S\theta_n S\alpha_n & r_n C\theta_n \\ S\theta_n & C\theta_n C\alpha_n & -C\theta_n S\alpha_n & r_n S\theta_n \\ 0 & S\alpha_n & C\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

[homogeneous transformation matrix from $(n-1)^{th}$ frame → n^{th} frame]

Note

$H_3^0 = H_1^0 H_2^1 H_3^2$

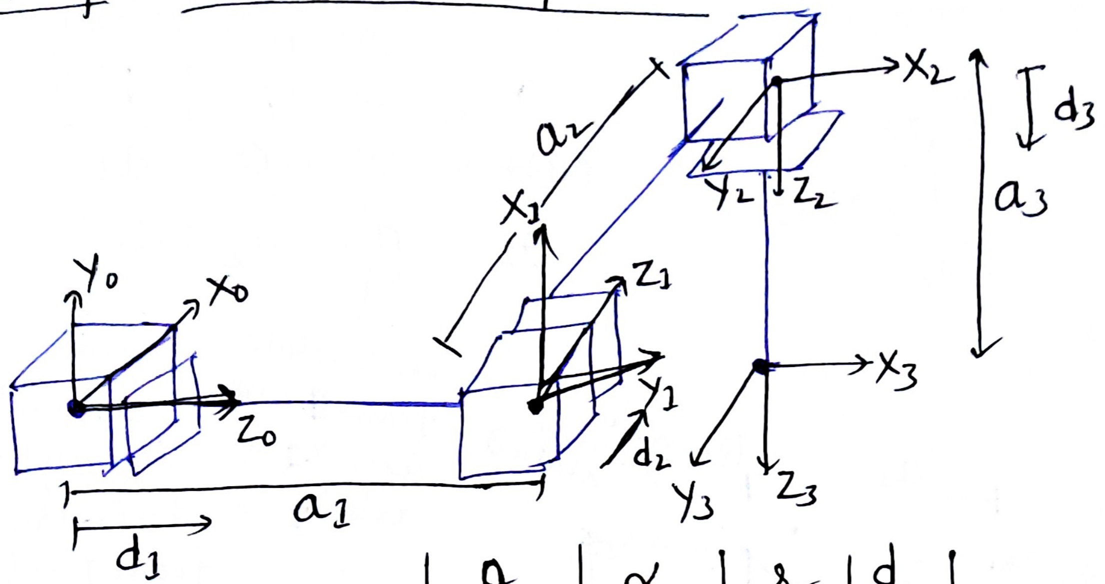
☆ H_2^1 → Homogeneous tran. matrix from frame 1 to frame 2

Put $n=2$ (Use 2nd row of DH table data)

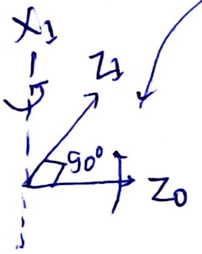
R1 U1 S5 P3

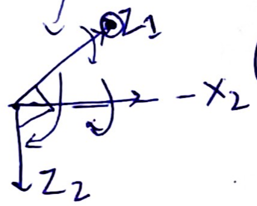
Example – Cartesian Manipulator

Prismatic joints



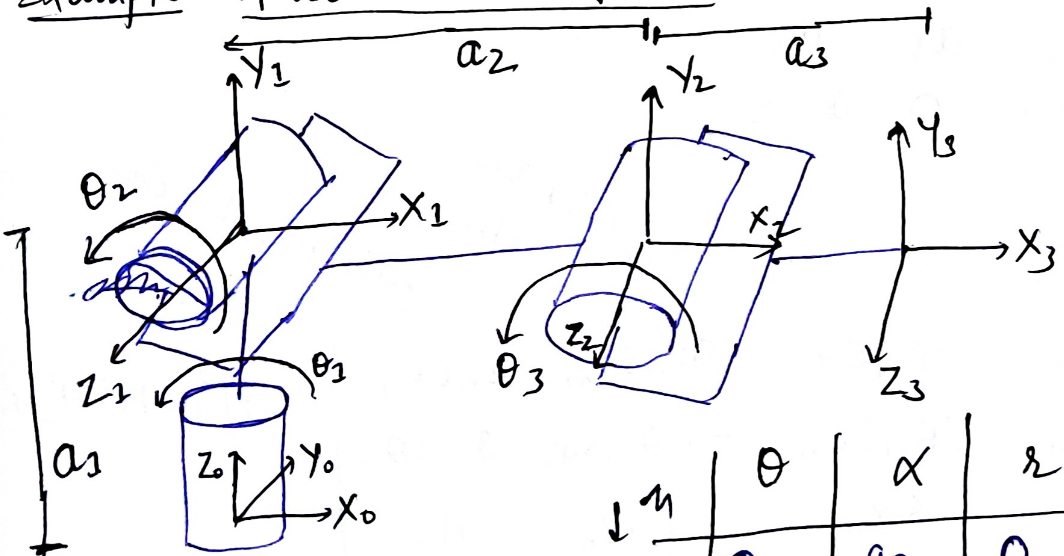
	θ	α	r	d
1	90	90	0	a_1+d_1
2	90	-90	0	a_2+d_2
3	0	0	0	a_3+d_3





(how do we have to rotate frame 1 around the axis X_2 in order to get Z_1 to match Z_2?)

Example – Articulated Manipulator

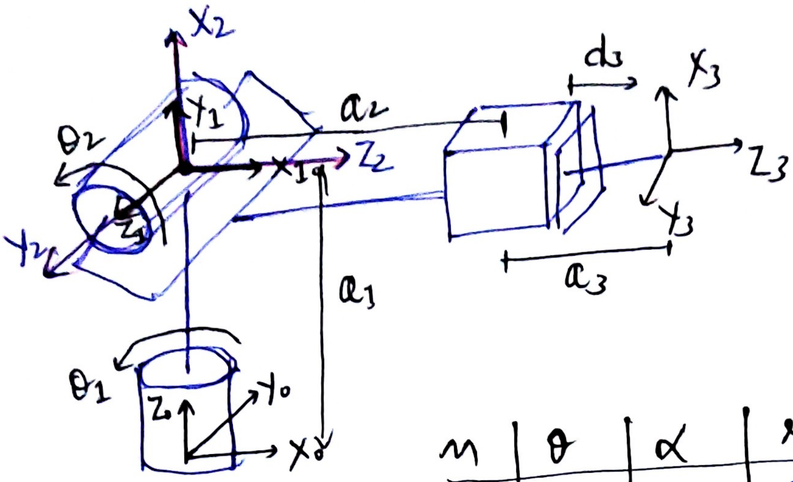


n	θ	α	r	d
1	$0+\theta_1 = \theta_1$	90	0	a_1
2	$0+\theta_2=\theta_2$	0	a_2	0
3	$0+\theta_3=\theta_3$	0	a_3	0

Scanned with ACE Scanner

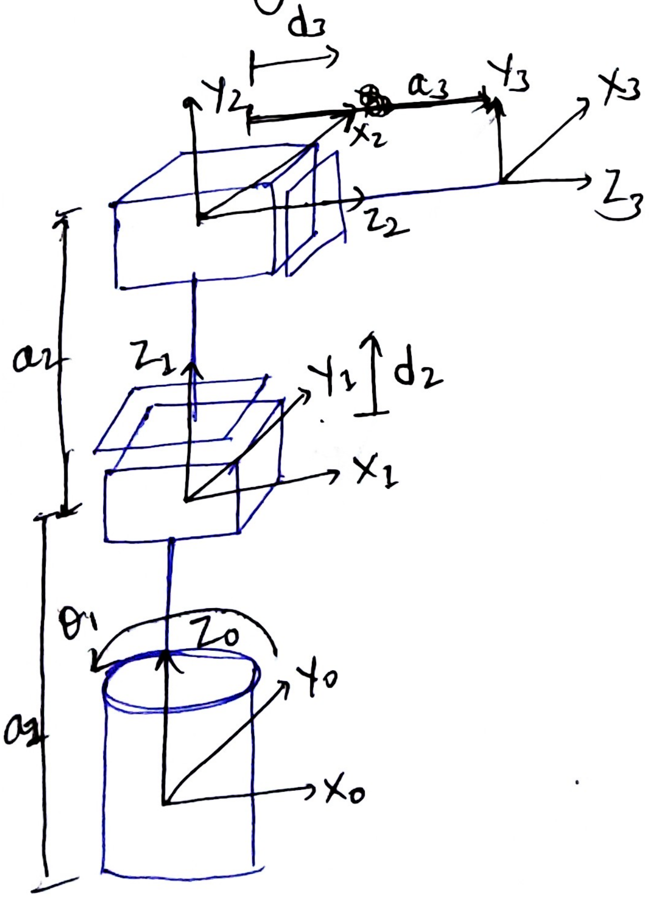
Example Spherical Manipulator

(Remember when we assign the frames a/c to DH Rules, we have to move the two frames back on to the center of the second joint from where we originally drew it & the center of the third joint in order to obey all four of the DH rules for assigning frames).



n	θ	α	r	d
1	$0+\theta_1=\theta_1$	90	0	a_1
2	$90+\theta_2$	90	0	0
3	0	0	0	$(a_2+a_3+d_3)$

Example Cylindrical Manipulator



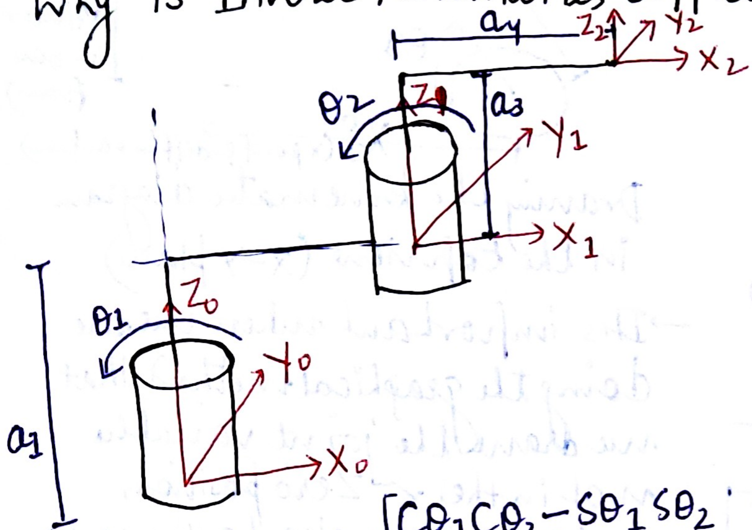
n	θ	α	r (X_n)	d (Z_{n-1})
1	$0+\theta_1=\theta_1$	0	0	a_1
2	90	90	0	a_2+d_2
3	0	0	0	a_3+d_3

Inverse Kinematics

End-Effector position & orientation $\xrightarrow{IK}$ Joint Variables

End-Effector position & orientation $\xleftarrow{FK}$ Joint Variables

Why is Inverse Kinematics difficult?



$$H_1^0 = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & a_2C\theta_1 \\ S\theta_1 & C\theta_1 & 0 & a_2S\theta_1 \\ 0 & 0 & 1 & a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_2^1 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & a_4C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & a_4S\theta_2 \\ 0 & 0 & 1 & a_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_2^0 = H_1^0 H_2^1 = \begin{bmatrix} C\theta_1C\theta_2 - S\theta_1S\theta_2 & -C\theta_1S\theta_2 - S\theta_1C\theta_2 & 0 & a_4C\theta_1C\theta_2 - a_4S\theta_1S\theta_2 + a_2C\theta_1 \\ S\theta_1C\theta_2 + C\theta_1S\theta_2 & -S\theta_1S\theta_2 + C\theta_1C\theta_2 & 0 & a_4S\theta_1C\theta_2 + a_4C\theta_1S\theta_2 + a_2S\theta_1 \\ 0 & 0 & 1 & a_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

displacement vector

x position of endeffector in base frame

Y pos of endeff. in base frame

z position of end effector in the base frame

Let's suppose we know the (X,Y,Z) of the endeffector we want the robot to have.

$$\boxed{\begin{aligned} X_2^0 &= a_4C\theta_1C\theta_2 - a_4S\theta_1S\theta_2 + a_2C\theta_1 \\ Y_2^0 &= a_4S\theta_1C\theta_2 + a_4C\theta_1S\theta_2 + a_2S\theta_1 \end{aligned}}$$

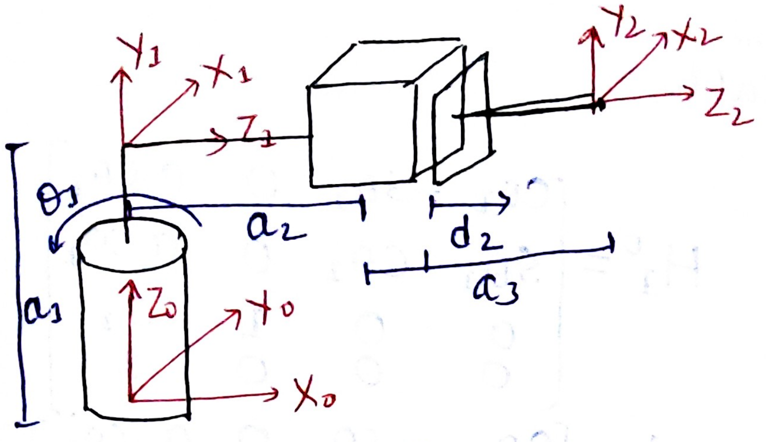
(Nonlinear system of Equations) Difficult problem to solve because of the presence of sin & cos

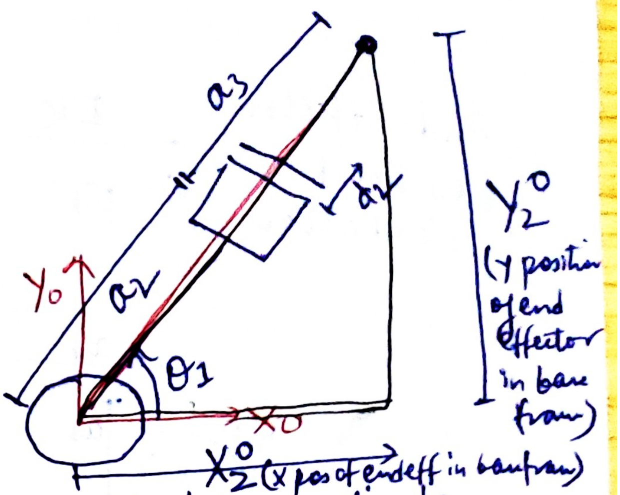
→ The only way we can solve above equation is to do some kind of numerical solutions. We can't solve it analytically.

→ One valid way to ~~...~~ do inverse Kinematics — To write out these equations and then use a numerical method to solve this non-linear system of equations.

→ Another method we can use to actually get an analytical solution — Graphical method.

Example (2 DOF manipulator)





Drawing the kinematic diagram in the top view (x-y plane)

$(X_2^0, Y_2^0, Z_2^0) \rightarrow$ inputs (known) in IK

→ It's important when we are doing the graphical method that we draw the joint variables not in their zero position and infact we also don't want to draw them in a position like 90° or 180°. We would prefer to draw them in a position like 45°. In this way, it would be much easier to solve this problem

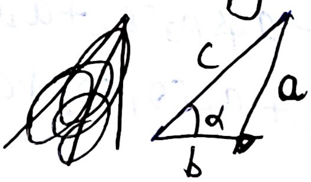
$$(X_2^0)^2 + (Y_2^0)^2 = (a_2 + a_3 + d_2)^2$$

$$\boxed{d_2 = \sqrt{(X_2^0)^2 + (Y_2^0)^2} - a_2 - a_3}$$

$$\tan\theta_1 = \frac{Y_2^0}{X_2^0}$$

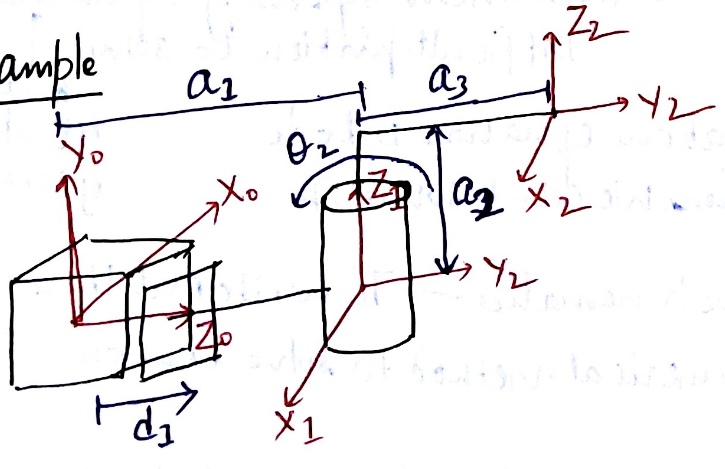
$$\boxed{\theta_1 = \tan^{-1}\left(\frac{Y_2^0}{X_2^0}\right)}$$

Law of Cosines

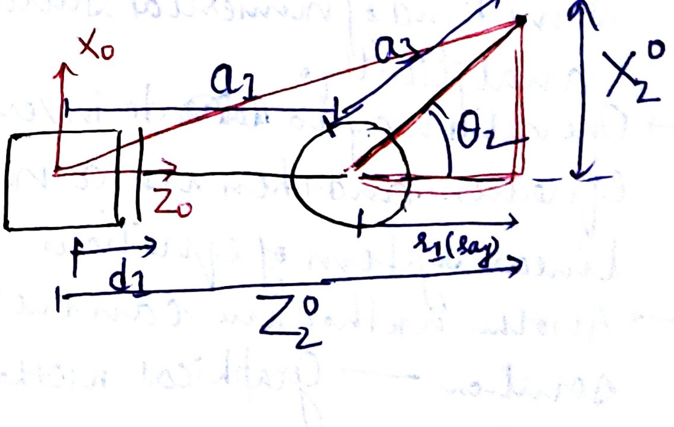


$$a^2 = b^2 + c^2 - 2bc\cos\alpha$$

Example



Top down view



$$\tan\theta_2 = \frac{x_2^0}{r_1} \qquad z_2^0 = a_1 + d_1 + r_1$$

$$\boxed{d_1 = z_2^0 - a_1 - r_1} \quad \text{— (III)}$$

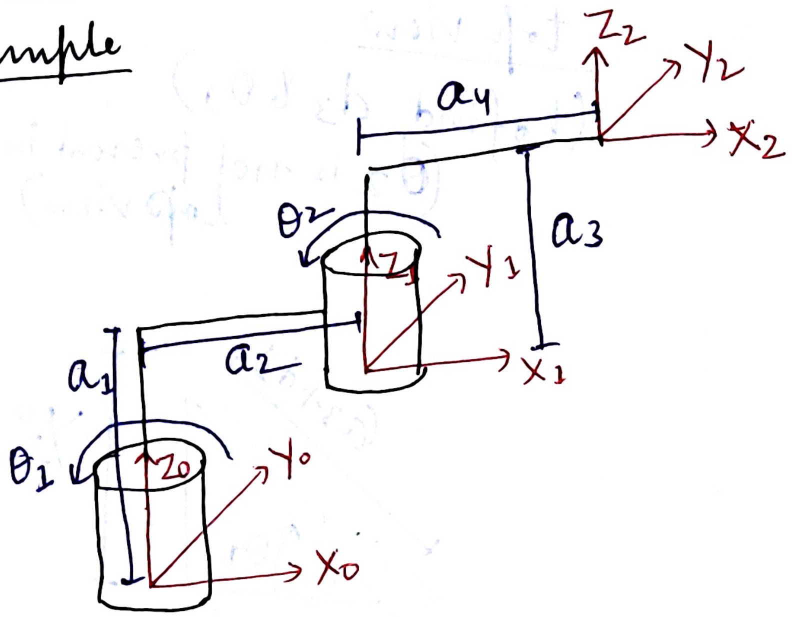
$$r_1^2 + (x_2^0)^2 = a_3^2$$

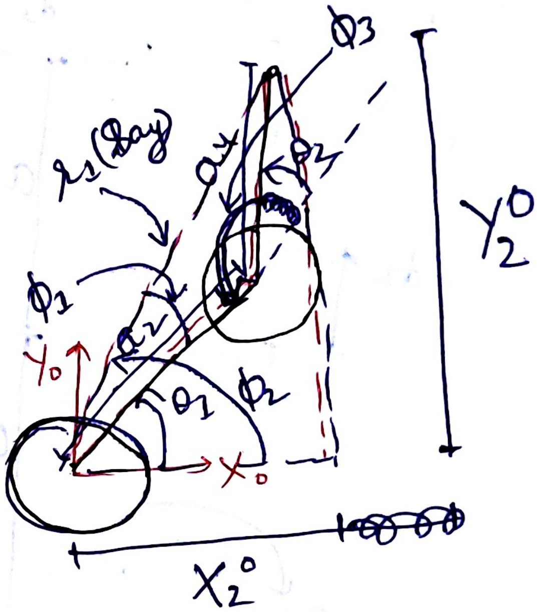
$$\boxed{r_1 = \sqrt{a_3^2 - (x_2^0)^2}} \quad \text{— (I)}$$

$$\boxed{\theta_2 = \tan^{-1}\left(\frac{x_2^0}{r_1}\right)} \quad \text{— (II)}$$

Top view

Example





$$(x_2^0)^2 + (y_2^0)^2 = r_1^2$$

$$\boxed{r_1 = \sqrt{(x_2^0)^2 + (y_2^0)^2}} \quad \text{— (I)}$$

$$\tan\phi_2 = \frac{y_2^0}{x_2^0}$$

$$\boxed{\phi_2 = \tan^{-1}\left(\frac{y_2^0}{x_2^0}\right)} \quad \text{— (III)}$$

$$\theta_1 = \phi_2 - \phi_1$$

$$a_4^2 = a_2^2 + r_1^2 - 2a_2 r_1 \cos\phi_1$$

(Law of Cosine)

$$\cos\phi_1 = \frac{a_2^2 + r_1^2 - a_4^2}{2a_2 r_1}$$

$$\boxed{\phi_1 = \cos^{-1}\left(\frac{a_2^2 + r_1^2 - a_4^2}{2a_2 r_1}\right)} \quad \text{— (II)}$$

so, $\theta_1 \Rightarrow$ (III) — (II)

$\phi_3 + \theta_2 = 180^\circ$

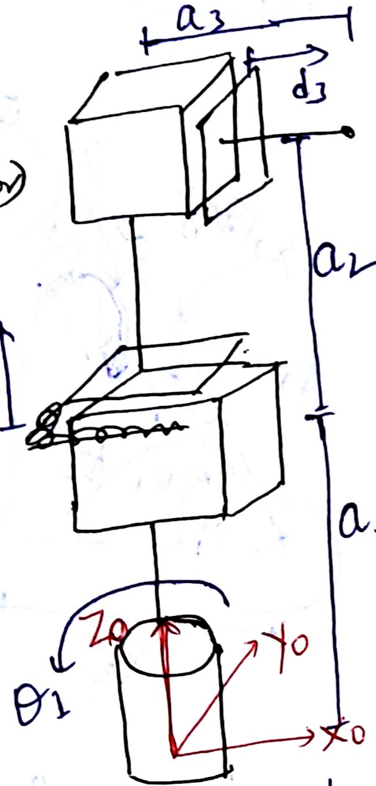
$$\boxed{\theta_2 = 180^\circ - \phi_3} \quad \text{— (V)}$$

$r_1^2 = a_2^2 + a_4^2 - 2a_2a_4\cos\phi_3$ (Law of Cosine)

$$\boxed{\phi_3 = \cos^{-1}\left(\frac{a_4^2 + a_2^2 - r_1^2}{2a_2a_4}\right)} \quad \text{— (IV)}$$

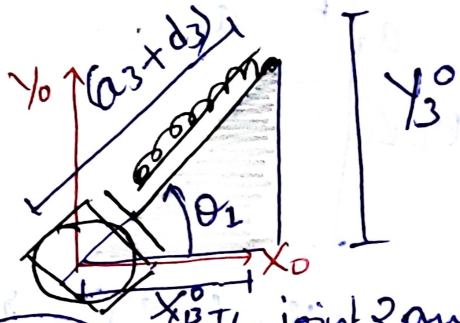
So, $\theta_2 \Rightarrow$ Use (IV) in (V).

Example (Cylindrical Manipulator)



for the IK problems, for position the other frames are irrelevant. (apart from frame 0)

top view (to find d_3 & θ_1) (θ_2 is not present in top view)



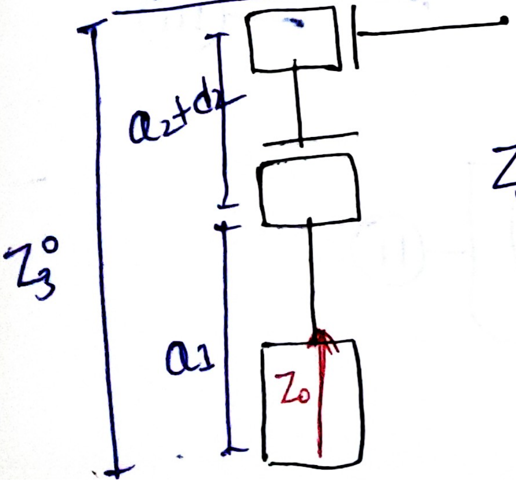
$\theta_1 \neq 0$ (Both joint 2 and joint 3 would be positioned on top of the joint 1)

$$\boxed{\theta_1 = \tan^{-1}\left(\frac{Y_3^0}{X_3^0}\right)} \quad \text{— (I)}$$

$(X_3^0)^2 + (Y_3^0)^2 = (a_3 + d_3)^2$

$$\boxed{d_3 = \sqrt{(X_3^0)^2 + (Y_3^0)^2} - a_3} \quad \text{— (II)}$$

Side view (to find d_2)

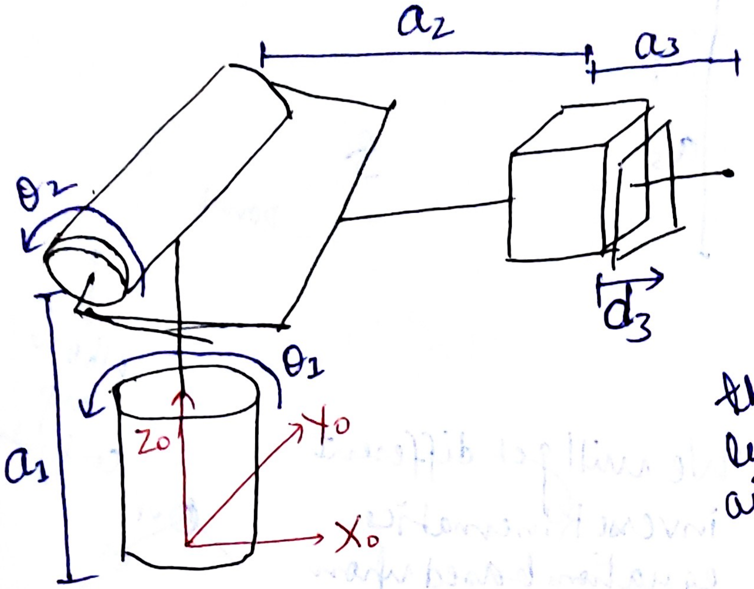


$Z_3^0 = a_1 + d_2 + a_2$

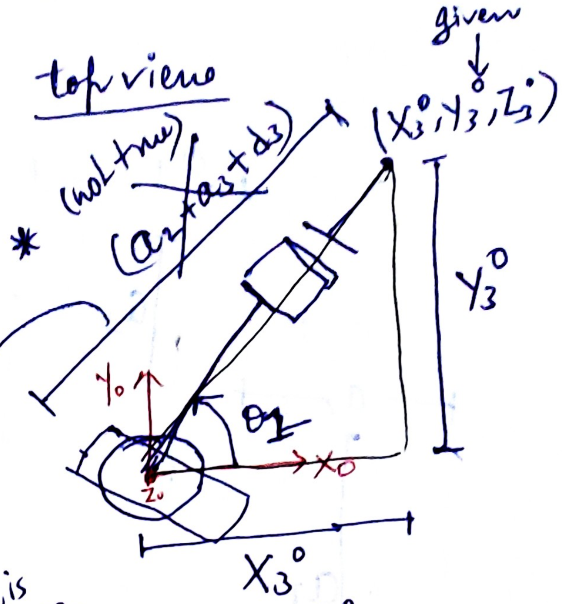
$$\boxed{d_2 = Z_3^0 - a_1 - a_2} \quad \text{— (III)}$$

Scanned with ACE Scanner

Example (spherical manipulator)



top view

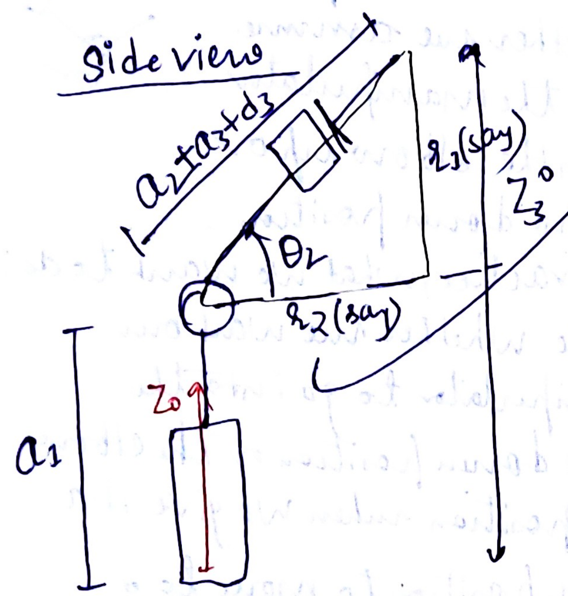


this length is actually r_2

$$\tan\theta_1 = \frac{Y_3^0}{X_3^0}$$

$$\boxed{\theta_1 = \tan^{-1}\left(\frac{Y_3^0}{X_3^0}\right)} \quad \text{— (I)}$$

side view



$$r_1^2 + r_2^2 = (a_2+a_3+d_3)^2$$

$$d_3 = \sqrt{r_1^2 + r_2^2} - a_2 - a_3$$

$$\boxed{r_1 = Z_3^0 - a_1} \quad \text{— (II)}$$

for r_2, take top view into consideration

$$\boxed{r_2 = \sqrt{(X_3^0)^2 + (Y_3^0)^2}} \quad \text{— (III)}$$

$$\text{so, } \left[d_3 = \sqrt{(\text{II})^2 + (\text{III})^2} - a_2 - a_3\right]$$

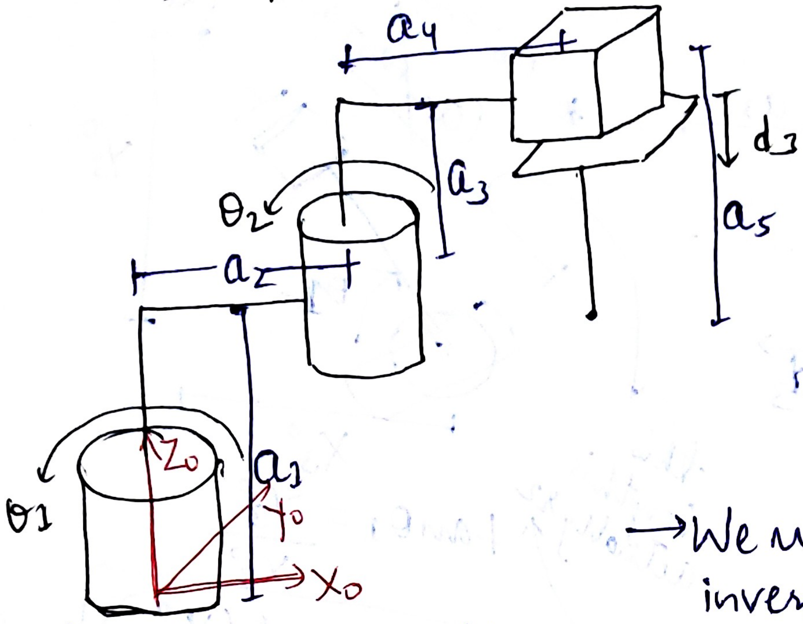
*(Reason)

θ_2 might not be in the zero position.

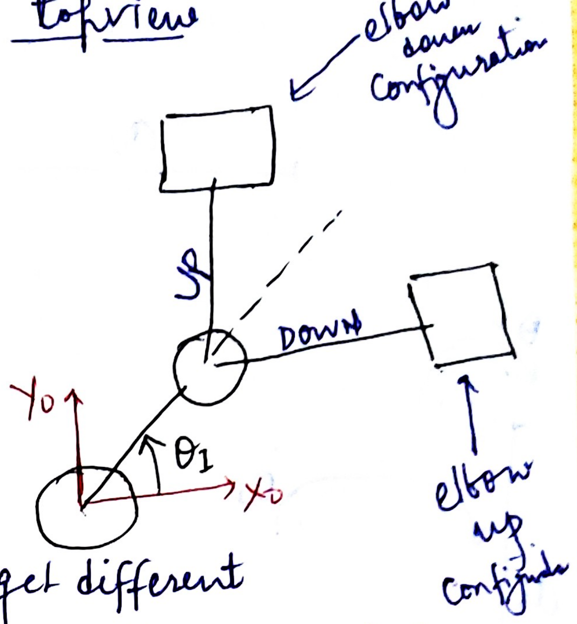
Imagine $\theta_2 = 90$, then the whole part of the arm (which look to have length $a_2+a_3+d_3$) would be pointing up straight towards us and the length of the distance would actually be 0. So, we don't know the length of hypothenuse yet.

RI UI S6 P2

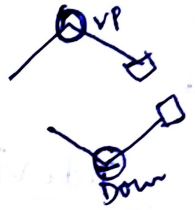
Example SCARA manipulator



topview

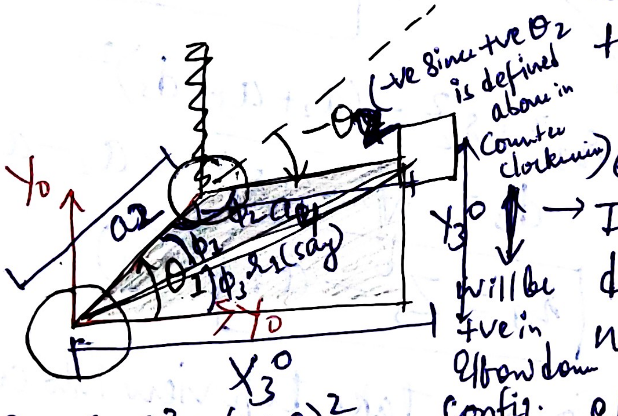


→ We will get different inverse Kinematics equations based upon whether we assume that the manipulator is in the elbow up or elbow down position.



→ In practice, what we want to do is decide whether we want our manipulator to go into the elbow down position or the elbow up position when we give it a target position to move to.

Elbow up Configuration can



$$r_1^2 = (X_3^0)^2 + (Y_3^0)^2$$

$$\boxed{r_1 = \sqrt{(X_3^0)^2 + (Y_3^0)^2}}$$

$$\phi_1 + \phi_3 = \theta_1$$

$$\boxed{\phi_3 = \tan^{-1}\left(\frac{Y_3^0}{X_3^0}\right)} \quad \text{— (I)}$$

$$a_4^2 = a_2^2 + r_1^2 - 2a_2 r_1 \cos\phi_1$$

$$\Rightarrow a_4^2 = a_2^2 + (X_3^0)^2 + (Y_3^0)^2 - 2a_2\sqrt{(X_3^0)^2 + (Y_3^0)^2}\cos\phi_1$$

$$\phi_1 = \cos^{-1}\left(\frac{a_2^2 + (X_3^0)^2 + (Y_3^0)^2 - a_4^2}{2a_2\sqrt{(X_3^0)^2 + (Y_3^0)^2}}\right) \quad \text{— (II)}$$

So, $\boxed{\theta_1 = (I) + (II)}$ — (III)

Now, $-\theta_2 + \phi_2 = 180$

$$\boxed{\theta_2 = \phi_2 - 180} \quad \text{— (IV)}$$

$$\underbrace{r_1}_{\text{already Known}}^2 = a_2^2 + a_4^2 - 2a_2 a_4 \cos\phi_2$$

$$\phi_2 = ? \quad \text{— (V)}$$

Put (V) in (IV) to get θ_2.

So, we have θ_1, θ_2.

Inverse Kinematics for Orientation or 6-DOF Inverse Kinematics

In order to be able to calculate Inverse Kinematics at all with more than three degrees of freedom robotic arm, we start by making an important assumption.

→ The first three joints are entirely responsible for POSITIONING the end effector and any additional joints are responsible for ORIENTATION of the end effector.

Note - Above assumption is the main reason why a spherical wrist is so common a device in manipulator designs we see in industry.

→ A spherical wrist is designed to try and make the link lengths as close to 0 as possible and is designed to allow three independent rotations. This is done so that it will be true that the spherical wrist has no effect on the positioning of the end-effector in the three linear positioning directions. It only has an effect on the rotation of the end effector.

procedures :-

Step #1 Draw a Kinematic diagram of only the first 3 joints, and do inverse Kinematics for position.

Step #2 Do forward Kinematics on the first three joints to get the rotation part, (R_3^0).

Step #3 Find the inverse of the R_3^0 matrix.

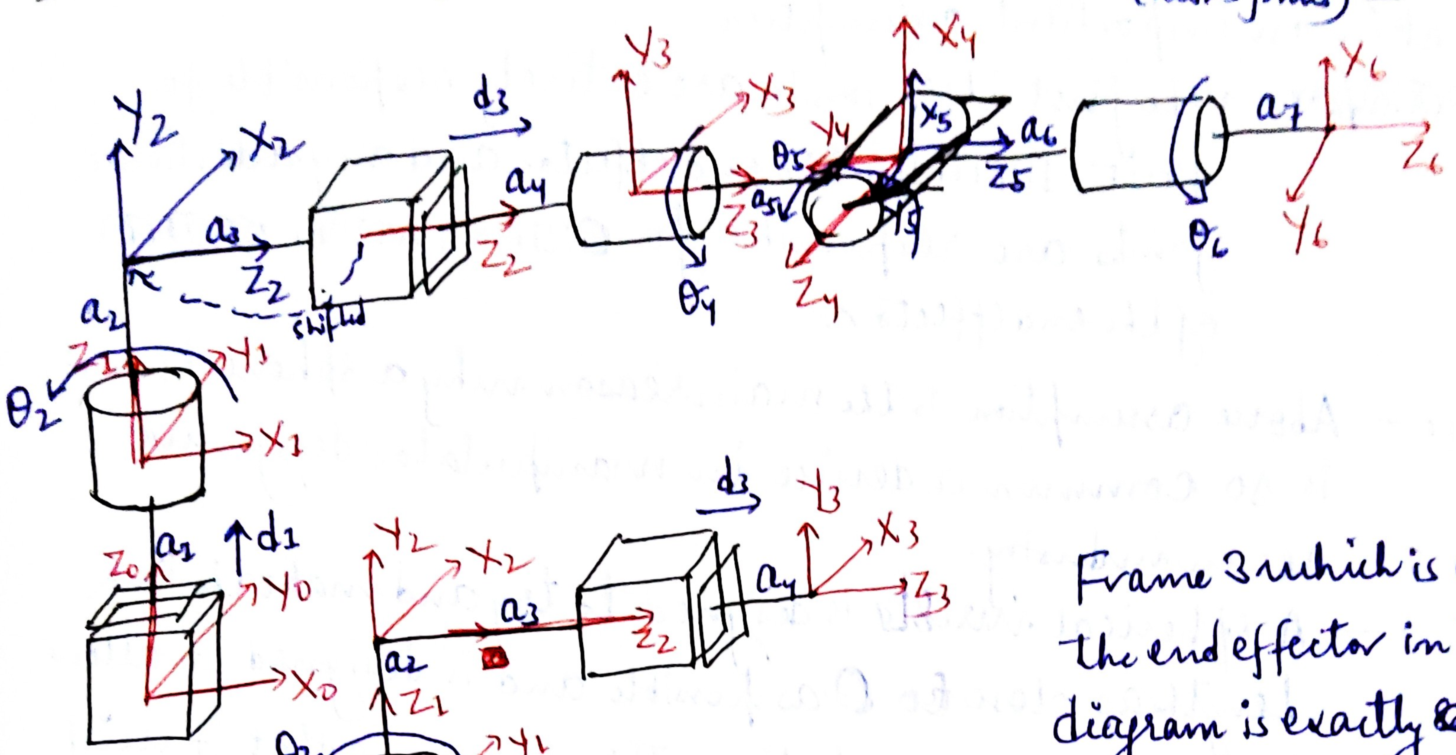
Step #4 Do FK on the last three joints and pull out the rotation part, R_6^3.

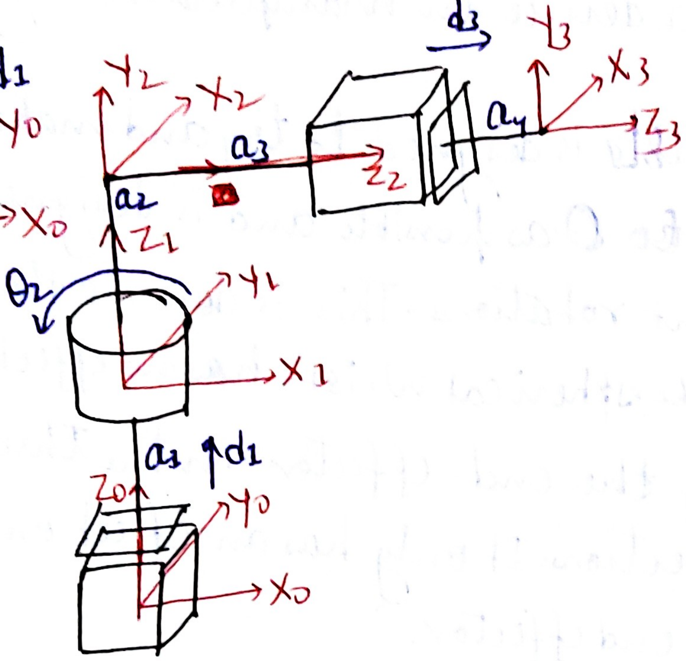
Step #5 Specify what we want the rotation matrix R_6^0 to be.

Step #6 Given a desired X, Y and Z position, solve for the first three joints using the IK eqns from Step #1.

Step #7 Plug in those variables and use the rotation matrix to solve for the last three joints.

Example (6DOF ⇒ Cylindrical manipulator with a Spherical wrist (last 3 joints))





Frame 3 which is on the end effector in this diagram is exactly the same as the frame 3 in the above whole manipulator with the wrist.

Step ①

Step ⑪ $R_3^0 = R_1^0 R_2^1 R_3^2$

$$R_1^0 = I_{3\times3} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I_{3\times3}$$

$$R_2^1 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} -\sin\theta_2 & 0 & \cos\theta_2 \\ \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \end{bmatrix}$$

$$R_3^2 = I \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I_{3\times3}$$

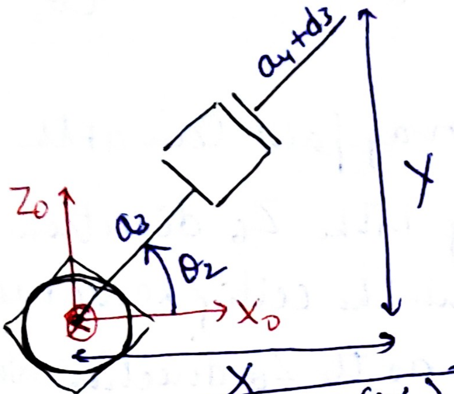
So, $$R_3^0 = I \begin{bmatrix} -\sin\theta_2 & 0 & \cos\theta_2 \\ \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \end{bmatrix} I = \begin{bmatrix} -\sin\theta_2 & 0 & \cos\theta_2 \\ \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \end{bmatrix}$$

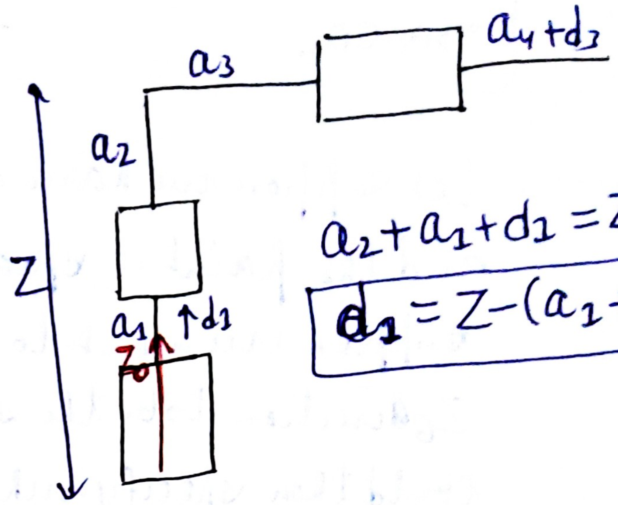
IK9

Inverse Kinematic

part of Step#1

topview | **Sideview**





$$\boxed{\theta_2 = \tan^{-1}\left(\frac{Y}{X}\right)}$$

$$\boxed{d_3 = \sqrt{X^2+Y^2} - a_3 - a_4}$$

$$a_2 + a_1 + d_1 = Z$$

$$\boxed{d_1 = Z - (a_1 + a_2)}$$

Step#3 Inverse of R^0_3 $(R_3^{0^{-1}})$ → Why do we need R_3^{0-1}?

$R^0_6 = R^0_3 R^3_6$

R^0_6: What we want the rot. of the end effector to be

R^0_3 → We already calculated this.

So, we would like to solve this eqn for R^3_6.

$$R^0_6 = R^0_3 R^3_6$$

$$R_3^{0-1} R^0_6 = R_3^{0-1} R^0_3 R^3_6$$

$$\boxed{R^3_6 = R_3^{0-1} R^0_6}$$

Step#4 $R^3_6 = R^3_4 R^4_5 R^5_6$

$$R^3_4 = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 \\ \sin\theta_4 & \cos\theta_4 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} -\sin\theta_4 & 0 & -\cos\theta_4 \\ \cos\theta_4 & 0 & -\sin\theta_4 \\ 0 & -1 & 0 \end{bmatrix}$$

$$R^4_5 = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 \\ \sin\theta_5 & \cos\theta_5 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \cos\theta_5 & 0 & \sin\theta_5 \\ \sin\theta_5 & 0 & -\cos\theta_5 \\ 0 & 1 & 0 \end{bmatrix}$$

$$R^5_6 = \begin{bmatrix} \cos\theta_6 & -\sin\theta_6 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_6 & -\sin\theta_6 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

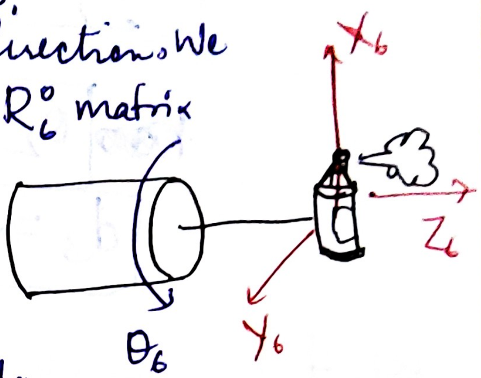
$$R^3_6 = R^3_4 R^4_5 R^5_6 = \begin{bmatrix} -\sin\theta_4 & 0 & -\cos\theta_4 \\ \cos\theta_4 & 0 & -\sin\theta_4 \\ 0 & -1 & 0 \end{bmatrix}\begin{bmatrix} \cos\theta_5 & 0 & \sin\theta_5 \\ \sin\theta_5 & 0 & -\cos\theta_5 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} \cos\theta_6 & -\sin\theta_6 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -S\theta_4 C\theta_5 & -C\theta_4 & -S\theta_4 S\theta_5 \\ C\theta_4 C\theta_5 & -S\theta_4 & C\theta_4 S\theta_5 \\ -S\theta_5 & 0 & C\theta_5 \end{bmatrix}\begin{bmatrix} C\theta_6 & -S\theta_6 & 0 \\ S\theta_6 & C\theta_6 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_6^3 = \begin{bmatrix} -S\theta_4 C\theta_5 C\theta_6 - C\theta_4 S\theta_6 & +S\theta_4 C\theta_5 S\theta_6 - C\theta_4 C\theta_6 & -S\theta_4 S\theta_5 \\ C\theta_4 C\theta_5 C\theta_6 - S\theta_4 S\theta_6 & -C\theta_4 C\theta_5 S\theta_6 - S\theta_4 C\theta_6 & C\theta_4 S\theta_5 \\ -S\theta_5 C\theta_6 & +S\theta_5 S\theta_6 & C\theta_5 \end{bmatrix}$$

$\theta_5 = ?$

Step #5 Let suppose we have a spray-paint Can at the end effector and the paint is spraying in the Z_6 direction and suppose we want to paint the ceiling so we want the Z_6 direction to be the same as the Z_0 direction. We could then specify what we want the R_6^0 matrix to be in order to accomplish this.



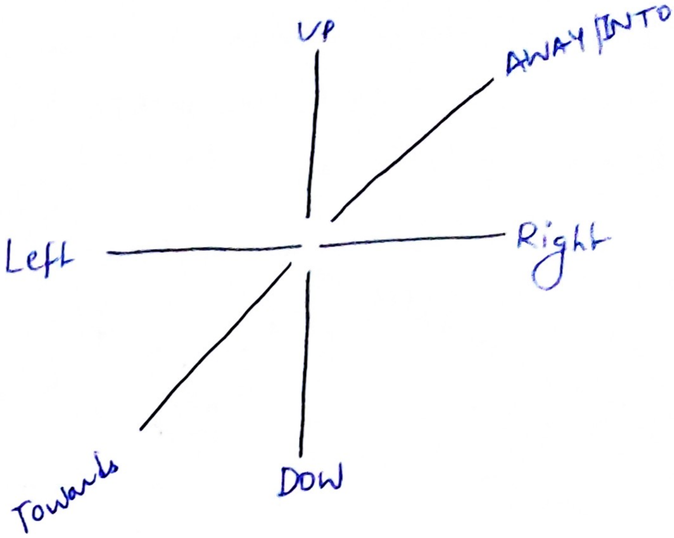
for eg: $R_6^0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ could be this, that would put Z_6 ~~direction~~ axis in the same direction as the Z_0 ~~direction~~ axis While X_6 is in the opposite direction as X_0 and Y_6 is in the Opposite direction as Y_0.

we can make R_6^0 anything we want it to be as long as it's a valid rotation matrix.

(I) Every row & every column needs to have a vector length of 1

$$R_6^0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \sqrt{(-1)^2+0^2+0^2} = 1$$

(II) A matrix needs to describe a right hand coordinate frame



EGR 455: Robotic Systems 1

Exam 1: Kinematics

Name:

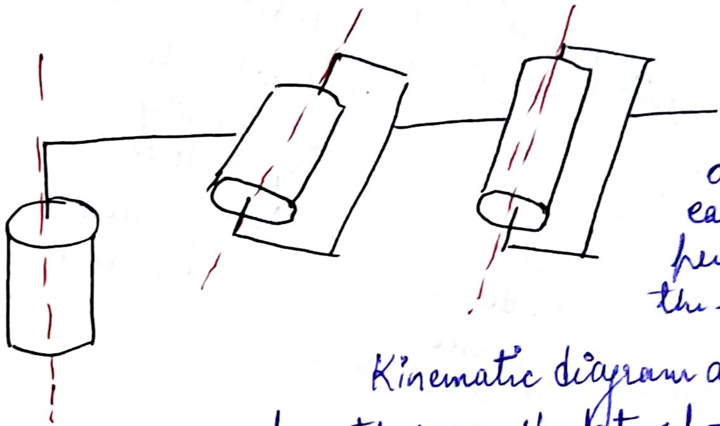
Equations:

$$R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix} \quad R_Y = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} \quad R_Z = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Equations for Graduate Students:

$$H = \begin{bmatrix} C\theta & -S\theta C\alpha & S\theta S\alpha & rC\theta \\ S\theta & C\theta C\alpha & -C\theta S\alpha & rS\theta \\ 0 & S\alpha & C\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1. One of the biggest robotics companies in the world today is a company called 'Kuka'. Shown here is one of Kuka's manipulators. Draw a kinematic diagram of this manipulator. Include only the three joints labeled with arrows. You don't have to label the diagram with frames, joint variables, or link lengths.



axis of rotation of joints 2 & 3 are parallel to each other but perpendicular to the first joint.

Kinematic diagram always shows how the manipulator looks when all the joints are set to their zero angle. There is no particular reason to suppose that this picture is also showing the manipulator the way it is when all of its joints are zero. So, as long as we have the relationship between the axis of rotation correct & as long as we have links showing any place there is an actual link in the picture, then our kinematic diagram will be right for the given picture.

2. Is the manipulator in question 1 a standard type? If so, what type is it?

Yes, Articulated manipulator.

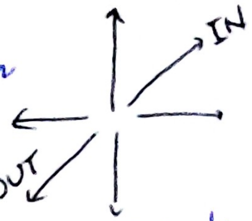
Notes (~~for~~ assigning frames by DH rules)

(I) We can't move the Zero from where it is → to get X axis to intersect. Zero frame is generally immovable/ground/floor.

(II) We can't move the frame that is on the end effector. If we don't have a frame that is actually located on the end effector, we have to go through an additional step later on to figure out where the end effector is.

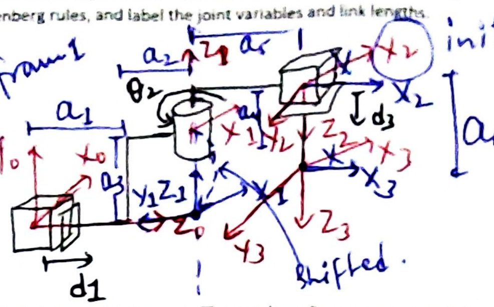
If we leave the frame on the end effector then we can tell where the end effector is just by looking at homogeneous transformation matrix.

It's important we move one frame as little as possible in order to get x axis to intersect the previous Z frame is by moving the frame in the Z direction.



3. Shown here is a kinematic diagram that is unlabeled. Draw frames on this diagram according to the Denavit-Hartenberg rules, and label the joint variables and link lengths.

~~Theta~~ We are using θ_2 as joint variable for frame 1 (rotated) to signify that when joint 1 is rotated, the frame actually moved is (rotated) frame 2 not frame 1.



initial selection of X_2 in this direction violate Rule #4 X_2 doesnt intersect Z_1. So, we try this by first changing the direction of X_2.

We will also change the direction of X_3 & set it along the X2 direction. It will simplify maths.

→ There are 2 ways we can solve problem of X axis not intersecting Z axis of previous frame & thereby violating rule #4.

(I) Try & change the direction of X_1 (preferred method)

(II) To move the frame (as little as possible).

→ How do we decide whether to lump two link length into one, eg. replace a_1 & a_2 by a lumped a ($a = a_1 + a_2$)?

✓ The rule that tells us whether or not we can lump together different link lengths into one length is to ask ourself whether those two link length will always be in the same direction no matter what's the joints are doing.

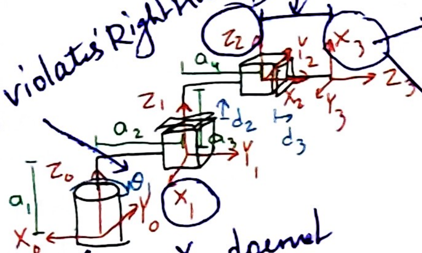
eg. We cannot lump together a2 & a5 in Ques (3) since when joint 2 starts to turn a2 and a5 will be in different direction

4. Shown here is a labeled kinematic diagram. There at least five errors in this diagram. Find five errors and write (in words) what is wrong.

link length not labelled

Z needs to be the dirn of motion.

θ_1 violates Right Hand Rule.



not ⊥ to Z_2 violating rule #4.

Also, X_3 needs to intersect Z_2.

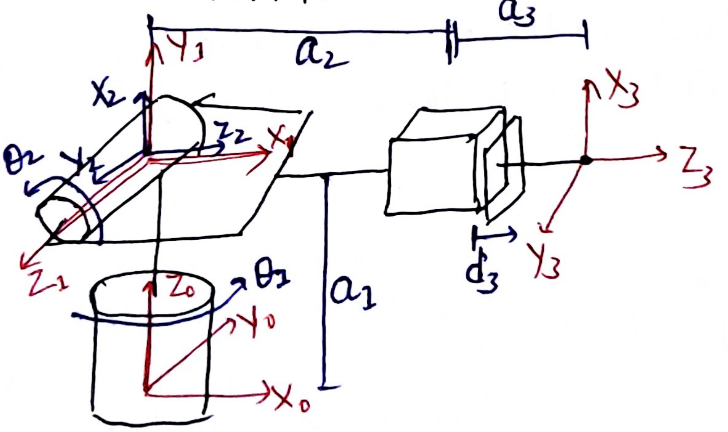
~~Z axis~~

the 0 frame violated the Right Hand Rule.

X_1 doesnot intersect Z0 which it should.

(5) Shown here is a labeled Kinematic diagram. Find the complete homogeneous transformation matrix (HTM) from the base-frame to the end-effector frame by first finding the rotation matrices & the displacement vectors, then assembling into the HTM.

2



[redacted]

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

$$H_1^0 = \begin{bmatrix} R_1^0 & & & d_1^0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_1^0 = \cancel{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}} \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 \\ S\theta_1 & C\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} C\theta_1 & 0 & S\theta_1 \\ S\theta_1 & 0 & -C\theta_1 \\ 0 & 1 & 0 \end{bmatrix}$$

(Standard rot. around Z_0)

$$R_2^1 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 \\ S\theta_2 & C\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} -S\theta_2 & 0 & C\theta_2 \\ C\theta_2 & 0 & S\theta_2 \\ 0 & 1 & 0 \end{bmatrix}$$

$$R_3^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I_{3\times3}$$

no rotation,
only translation
(Identity matrix)

$$d_1^0 = \begin{bmatrix} 0 \\ 0 \\ a_1 \end{bmatrix} \qquad d_2^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad d_3^2 = \begin{bmatrix} 0 \\ 0 \\ a_2 + a_3 + d_3 \end{bmatrix}$$

frame 2 is on top of frame 1. No matter how θ_2 turns, the centre of frame 2 not going to move anywhere. So, displacement is (0,0,0).

(displacement from the center of frame 2 to center of frame 3 is still going to be in Z_2 direction no matter what happens to θ_2)

so, $$H_1^0 = \begin{bmatrix} C\theta_1 & 0 & S\theta_1 & 0 \\ S\theta_1 & 0 & -C\theta_1 & 0 \\ 0 & 1 & 0 & a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

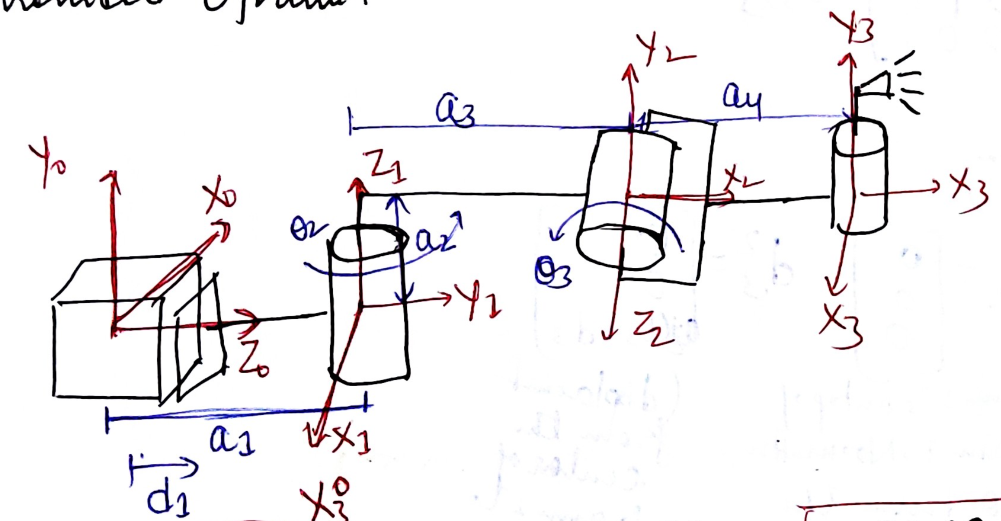
$$H_2^1 = \begin{bmatrix} -S\theta_2 & 0 & C\theta_2 & 0 \\ C\theta_2 & 0 & S\theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_3^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a_2+a_3+d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

⑥ Create a Denavit-Hartenberg parameter table for question ⑤.

	θ	α	r	d
1	$0+\theta_1=\theta_1$	90	0	a_1
2	$90+\theta_2$	90	0	0
3	0	0	0	$a_2+a_3+d_3$

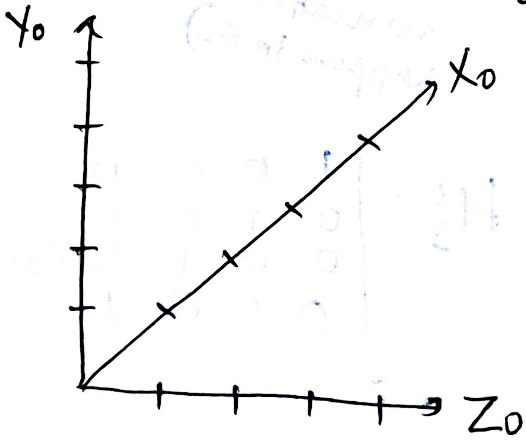
⑦ Shown here is a ~~kinetic~~ kinematic diagram along with its homogeneous transformation matrix. There is a spray paint can attached to the end-effector as shown in the diagram. When the link lengths and joint variables are set as shown, where is the spray paint can located and in what direction is it pointing. Answer by drawing the spray-paint can on the provided 0 frame.



$a_1=1$, $a_2=2$, $a_3=1$
$a_4=2$, $d_1=2$,
$\theta_2=90°$
$\theta_3=-90°$

$$H_3^0=\begin{array}{c} X_3^0 \\ \left[\begin{array}{cccc} S\theta_2 C\theta_3 & -S\theta_2 S\theta_3 & -C\theta_2 & a_4 C\theta_3 S\theta_2 + a_3 S\theta_2 \\ S\theta_3 & C\theta_3 & 0 & a_4 S\theta_3 + a_2 \\ C\theta_2 C\theta_3 & -C\theta_2 S\theta_3 & S\theta_2 & a_4 C\theta_2 S\theta_3 + a_3 C\theta_2 + a_1 + d_1 \\ 0 & 0 & 0 & 1 \end{array}\right]\end{array}$$

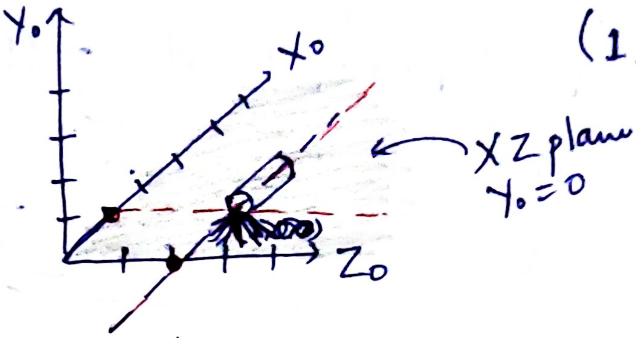
displacement vector telling us where the third center of the third frame is located relative to 0th frame



$X_3^0 = a_4 C\theta_3 S\theta_2 + a_3 S\theta_2 = 1(1) = 1$

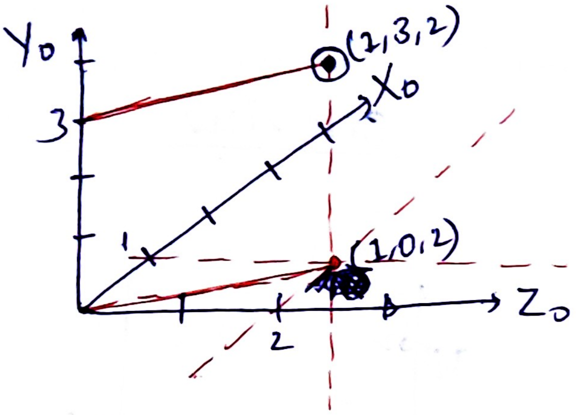
$Y_3^0 = a_4 S\theta_3 + a_2 = 2(-1) + 2 = 0$

$Z_3^0 = a_4 C\theta_2 S\theta_3 + a_3 C\theta_2 + a_1 + d_1 = 2(0)(\ ) + 1(0) + 1 + 1 = 2$

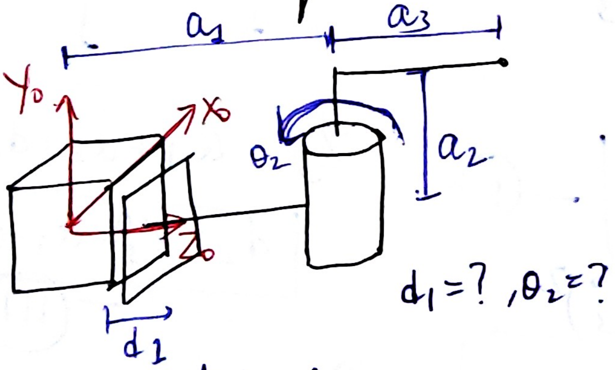


(1, 0, 2)

Example
(1, 3, 2)



⑧ Shown here is the Kinematic diagram of a 2-degree-of-freedom manipulator. Find the inverse Kinematics equations.



→ In order to find out the direction in which spray-paint ~~can~~ can is ~~pointing~~ spraying, we have to find out the direction in which X_3 is pointing.

✓ X_3 is represented by the first column in the homogeneous transformation matrix.

$$X_3^0 = \begin{bmatrix} S\theta_2 C\theta_3 \\ S\theta_3 \\ C\theta_2 C\theta_3 \end{bmatrix} = \begin{array}{c} X_0 \\ Y_0 \\ Z_0 \end{array}\begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \quad (\text{for } \theta_2 = 90°, \ \theta_3 = -90°)$$

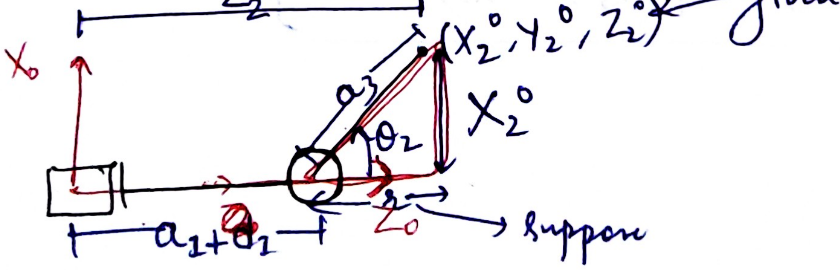
So, spray-paint can is ~~pointing~~ spraying in the opposite of the Y_0 direction

→ Also, spray-paint can is pointing along Y_3 direction.

$$Y_3^0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (\text{same as } X_0 \text{ direction})$$

top view

(Don't use intermediate frame)



Scanned with ACE Scanner

$\sin\theta_2 = \frac{X_2^0}{a_3} \Rightarrow \boxed{\theta_2 = \sin^{-1}\left(\frac{X_2^0}{a_3}\right)}$ — (I)

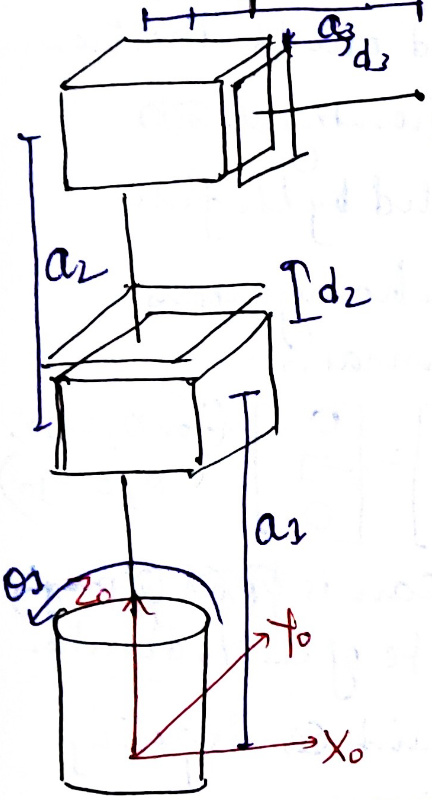
$\boxed{d_1 = Z_2^0 - a_1 - r}$ — (III)

$\cancel{Z_2^0 = a_1 + d_1 + a_3\cos\theta_2}$

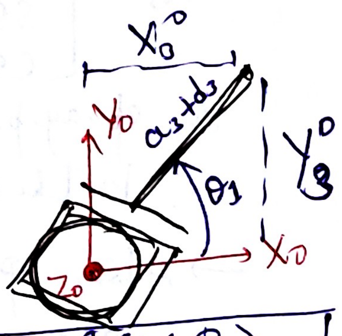
$\Rightarrow \boxed{d_1 = (Z_2^0 - a_1 - a_3\cos\theta_2)}$ — (II)

$\tan\theta_2 = \frac{X_2^0}{r} \Rightarrow \boxed{r = \frac{X_2^0}{\tan\theta_2}}$ — (II)

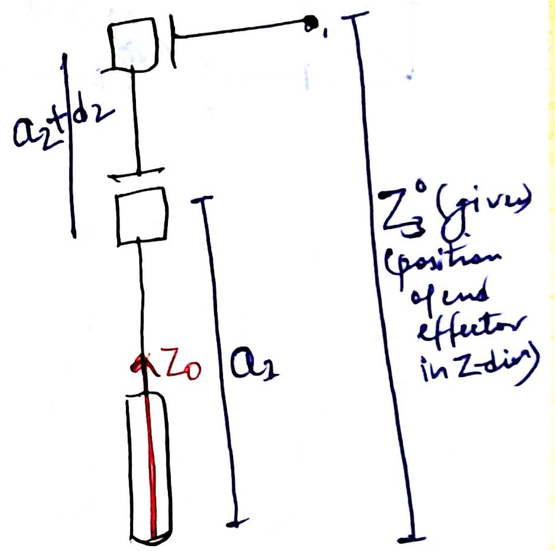
(9) Shown here is the Kinematic diagram of a 3-degree-of-freedom manipulator. Find the inverse ~~Kinematics~~ Kinematics equation.



top view



side view



$\boxed{\theta_1 = \tan^{-1}\left(\frac{Y_3^0}{X_3^0}\right)}$ — (I)

$(a_3+d_3)^2 = (X_3^0)^2 + (Y_3^0)^2$

$\boxed{d_3 = \sqrt{(X_3^0)^2 + (Y_3^0)^2} - a_3}$ — (II)

$Z_3^0 = a_1 + a_2 + d_2$

$\boxed{d_2 = Z_3^0 - a_1 - a_2}$ — (III)

Jacobian Matrix

(figure out the velocities of the joints that will get the endeffector to move at the right velocity taking into account the position of manipulator at each point in time).

Relates end effector velocities to joint velocities.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ w_x \\ w_y \\ w_z \end{bmatrix} = J \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

Jacobian Matrix

How fast is the joint moving?

(we use q to refer to a joint we dont know yet whether it's a revolute joint or a prismatic joint. So, if our manipulator is all revolute joints we will replace then qs with θs and if our manipulator is all prismatic joints we will replace them qs with ds.

✓ If we have 2 revolute joints & one prismatic joint we will replace the first two qs with θs and third q with d.

How fast the end effector is moving in the x-direction? (Here x, y & z are directions are measured in the base frame)

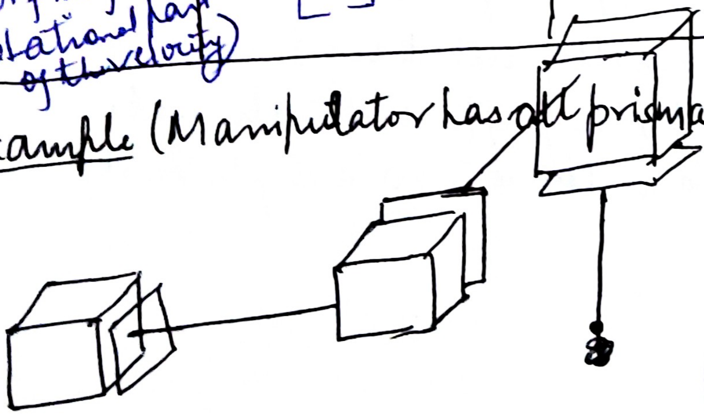
How fast the end-effector is rotating around the x-axis during the motion?

total number of joints in the manipulator

'i' denotes which joint

	Prismatic	Revolute
Linear (for finding the linear part of the velocity)	$R^0_{i-1}\begin{bmatrix}0\\0\\1\end{bmatrix}$	$R^0_{i-1}\begin{bmatrix}0\\0\\0\end{bmatrix} \times (d^0_n - d^0_{i-1})$
Rotational (for finding the rotational part of the velocity)	$\begin{bmatrix}0\\0\\0\end{bmatrix}$	$R^0_{i-1}\begin{bmatrix}0\\0\\1\end{bmatrix}$

Example (Manipulator has all prismatic joints)



for 1st joint of our manipulator

for 2nd joint

for 3rd joint

J

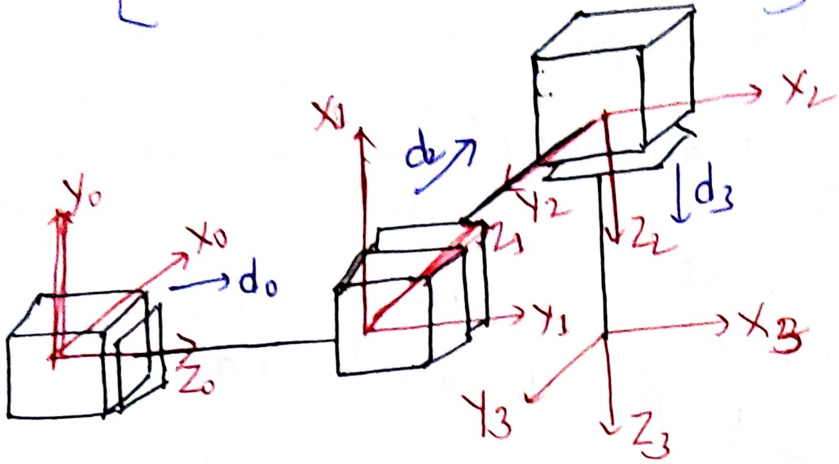
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{bmatrix} \begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix}$$

$$J = \begin{bmatrix} R_0^0\begin{bmatrix}0\\0\\1\end{bmatrix} & R_1^0\begin{bmatrix}0\\0\\1\end{bmatrix} & R_2^0\begin{bmatrix}0\\0\\1\end{bmatrix} \\ \begin{matrix}0\\0\\0\end{matrix} & \begin{matrix}0\\0\\0\end{matrix} & \begin{matrix}0\\0\\0\end{matrix} \end{bmatrix}$$

$R_0^0 \rightarrow$ Rotation from frame 0 to frame 0

$R_1^0 \rightarrow$ Rotation from frame 0 to frame 1.

$R_2^0 \rightarrow$ Rotation from frame 0 to frame 2.



$R_0^0 = I_{3\times3}$

$$R_1^0 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$R_2^1 = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

$$R_2^0 = R_1^0 R_2^1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$J = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

So,
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix}$$

$\dot{x} = \dot{d}_2$ $w_x = 0$

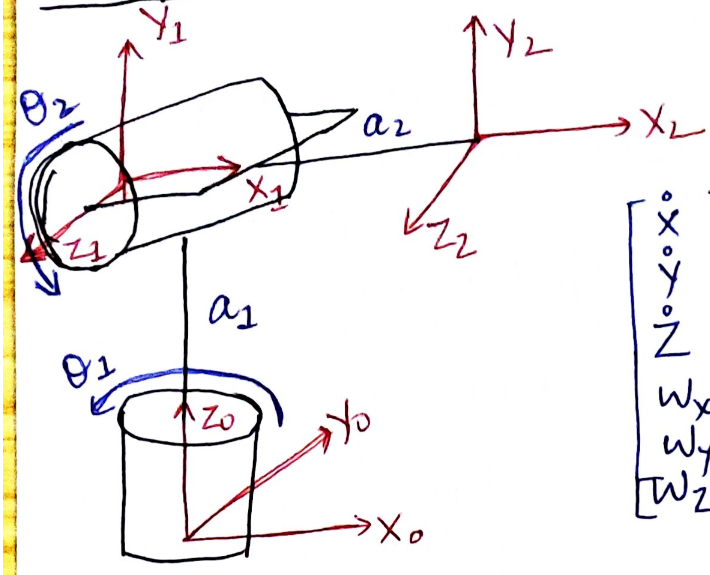
$\dot{y} = -\dot{d}_3$ $w_y = 0$

$\dot{z} = \dot{d}_1$ $w_z = 0$

No matter what we do with our three joints we can't get our end effector to rotate around x, y or z.

Imagine we have a gripper attached to above end effector, could we get the gripper to point into a different direction? No matter what we do with above three prismatic joints, the gripper will always be pointed in the same direction. There is no way to rotate the gripper.

Example



$n = 2$ (total number of joints = 2)

$i=1 \qquad\qquad\qquad\qquad i=2$

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\\ w_x\\ w_y\\ w_z\end{bmatrix} = \begin{bmatrix} R_0^0\begin{bmatrix}0\\0\\1\end{bmatrix}\underset{\uparrow\ \text{crossproduct}}{\times}(d_2^0-d_0^0) & R_1^0\begin{bmatrix}0\\0\\1\end{bmatrix}\times(d_2^0-d_1^0)\\ R_0\begin{bmatrix}0\\0\\1\end{bmatrix} & R_1^0\begin{bmatrix}0\\0\\1\end{bmatrix}\end{bmatrix}\begin{bmatrix}\dot\theta_1\\ \dot\theta_2\end{bmatrix}$$

$$R_0^0 = I_{3\times3}$$

$$H_2^0 = \begin{bmatrix} R_2^0 & d_2^0 \\ 0\ 0\ 0 & 1\end{bmatrix} = \begin{bmatrix} C\theta_1C\theta_2 & -C\theta_1S\theta_2 & S\theta_1 & a_2C\theta_1C\theta_2\\ S\theta_1C\theta_2 & -S\theta_1S\theta_2 & -C\theta_1 & a_2S\theta_1C\theta_2\\ 0 & C\theta_2 & 0 & a_2S\theta_2+a_1\\ 0 & 0 & 0 & 1\end{bmatrix}$$

(d_2^0 = last column, $\boxed{\begin{matrix}a_2C\theta_1C\theta_2\\ a_2S\theta_1C\theta_2\\ a_2S\theta_2+a_1\end{matrix}}$)

$$d_0^0 = \begin{bmatrix}0\\0\\0\end{bmatrix} \qquad R_1^0 = \begin{bmatrix} C\theta_1 & 0 & S\theta_1\\ S\theta_1 & 0 & -C\theta_1\\ 0 & 1 & 0\end{bmatrix} \qquad d_1^0 = \begin{bmatrix}0\\0\\a_1\end{bmatrix}$$

So,

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\\ w_x\\ w_y\\ w_z\end{bmatrix} = \begin{bmatrix} -a_2S\theta_1C\theta_2 & -a_2C\theta_1S\theta_2\\ a_2C\theta_1C\theta_2 & -a_2S\theta_1S\theta_2\\ 0 & 2a_2C\theta_2\\ 0 & S\theta_1\\ 0 & -C\theta_1\\ 1 & 0\end{bmatrix}\begin{bmatrix}\dot\theta_1\\ \dot\theta_2\end{bmatrix}$$

$$\dot{x} = -a_2S\theta_1C\theta_2\dot\theta_1 - a_2C\theta_1S\theta_2\dot\theta_2$$
$$\dot{y} = a_2C\theta_1C\theta_2\dot\theta_1 - a_2S\theta_1S\theta_2\dot\theta_2$$
$$\dot{z} = a_2C\theta_2\dot\theta_2$$ (depends on speed of joint 2)

$$\left.\begin{matrix} w_x = S\theta_1\dot\theta_2\\ w_y = -C\theta_1\dot\theta_2\end{matrix}\right]$$ Speed of rot. of end effector around X & Y axis is dependent only on speed of θ_2 (joint 2)

$$\left.w_z = \dot\theta_1\right]$$ Speed of rot. of the end effector around Z_0 axis is dependent only on speed of θ_1 (joint 1)

Path Planning and Trajectory Generation using the Jacobian Matrix

figure out the points in space through which the end effector will pass.

↓

we might define the path as an equation

figure out the velocity components of the end-effector motion along the path.

speed + direction of the end effector

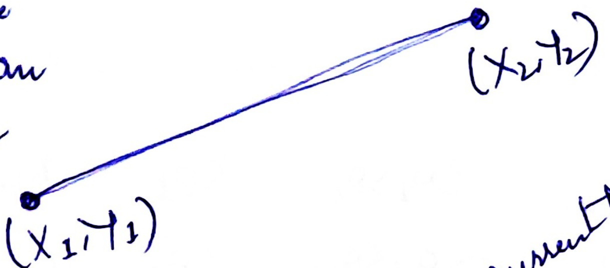
when we are doing path planing in robotics, there is a particular kind of path equation that is most useful to us, it's called a **Parametric Equation**.

defines the points on a path relative to a parameter that varies from one value at one end of the path to another value at the other end of the path.

$X(t) = \ldots$

$Y(t) = \ldots$

Here, t is a parameter.



Parametric eqns of above line

current time

$$x = \frac{v(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}\, t + x_1$$

(x-position of end effector at any point in time)

$$y = \frac{v(y_2 - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}\, t + y_1$$

velocity of the end effector (speed at which we want the end effector to be moving through space in a given direction)

Trajectory generation step

$$\dot{x} = \frac{v(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \qquad \dot{y} = \frac{v(y_2 - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ w_x \\ w_y \\ w_z \end{bmatrix} = J \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

This arrangement calculates the end effector velocities given the joint velocities

Calculate the joint velocities given the end-effector velocities

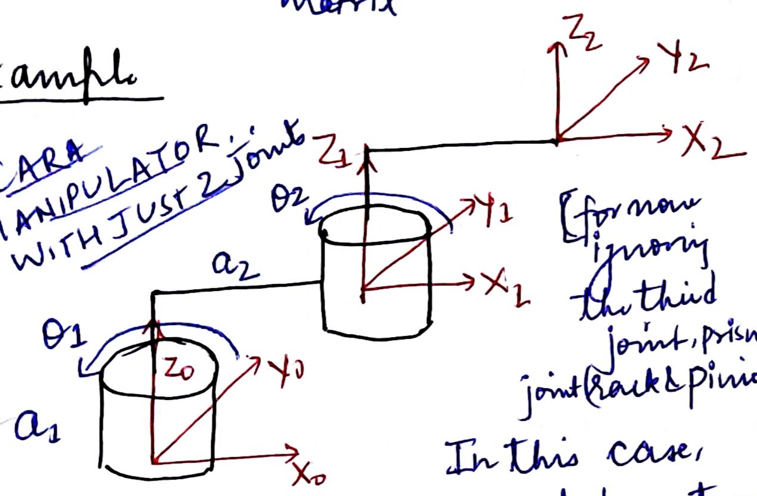
$(J^{-1}J = I)$

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} = J^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ w_x \\ w_y \\ w_z \end{bmatrix}$$

inverse of Jacobian matrix

Example

SCARA MANIPULATOR WITH JUST 2 Joints



[for now ignoring the third joint, prismatic joint (rack & pinion)]

$n = 2 \quad (i = 1, 2)$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ w_x \\ w_y \\ w_z \end{bmatrix} = \begin{array}{c} \begin{array}{cc} i=1 & i=2 \end{array} \\ \begin{bmatrix} \square & \square \\ \square & \square \\ \square & \square \\ \square & \square \\ \square & \square \\ \square & \square \end{bmatrix} \end{array} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

In this case, we only have two joints, so we can only control two of the ~~joints~~ velocities (along x & y direction)

Linear velocities along x & y

So,
$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \square & \square \\ \square & \square \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

2 revolute joints

Refer the below table

	Prismatic	Revolute
Linear	$R^0_{i-1}\begin{bmatrix}0\\0\\1\end{bmatrix}$	$R^0_{i-1}\begin{bmatrix}0\\0\\1\end{bmatrix} \times (d^0_n - d^0_{i-1})$
Rotational	$\begin{bmatrix}0\\0\\0\end{bmatrix}$	$R^0_{i-1}\begin{bmatrix}0\\0\\1\end{bmatrix}$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} R^0_0\begin{bmatrix}0\\0\\1\end{bmatrix} \times (d^0_2 - d^0_0) & R^0_1\begin{bmatrix}0\\0\\1\end{bmatrix} \times (d^0_2 - d^0_1) \\ R^0_0\begin{bmatrix}0\\0\\1\end{bmatrix} & R^0_1\begin{bmatrix}0\\0\\1\end{bmatrix} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$R^0_0 = I$

$$H_2^0 = \begin{bmatrix} c\theta_1 c\theta_2 - s\theta_1 s\theta_2 & -c\theta_1 s\theta_2 - s\theta_1 c\theta_2 & 0 & a_4 c\theta_1 c\theta_2 - a_4 s\theta_1 s\theta_2 + a_2 c\theta_1 \\ s\theta_1 c\theta_2 + c\theta_1 s\theta_2 & -s\theta_1 s\theta_2 + c\theta_1 c\theta_2 & 0 & a_4 s\theta_1 c\theta_2 + a_4 c\theta_1 s\theta_2 + a_2 s\theta_1 \\ 0 & 0 & 1 & a_3 + a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad d_2^0$$

$$H_1^0 = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & a_2 c\theta_1 \\ s\theta_1 & c\theta_1 & 0 & a_2 s\theta_1 \\ 0 & 0 & 1 & a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad d_1^0$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} \underbrace{-a_4 s\theta_1 c\theta_2 - a_4 c\theta_1 s\theta_2 - a_2 s\theta_1}_{J_{11}} & \underbrace{-a_4 s\theta_1 c\theta_2 - a_4 c\theta_1 s\theta_2}_{J_{12}} \\ \underbrace{a_4 c\theta_1 c\theta_2 - a_4 s\theta_1 s\theta_2 + a_2 c\theta_1}_{J_{21}} & \underbrace{a_4 c\theta_1 c\theta_2 - a_4 s\theta_1 s\theta_2}_{J_{22}} \\ 0 & 0 \\ R_0^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & R_1^0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

Since we are only going to control x & y linear velocities, we are only going to take the first two rows of above matrix.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \Rightarrow \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = J^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$

$$J^{-1} = \frac{1}{J_{11}J_{22} - J_{12}J_{21}} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix}$$

$$\boxed{\begin{aligned} \dot{\theta}_1 &= J_{11}^{-1}\dot{x} + J_{12}^{-1}\dot{y} \\ \dot{\theta}_2 &= J_{21}^{-1}\dot{x} + J_{22}^{-1}\dot{y} \end{aligned}}$$

Note :-

If $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ then

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Scanned with ACE Scanner

PM1

Parallel Manipulators

→ Each actuated joint is attached both to the ground and to the end-effector.

→ Two common types of parallel manipulators:

1. Steward/Gough Platform

(Used for flight Simulators or other kind of Simulators)

→ defined by having all of its joints which are actuated be prismatic joints.

2. Delta robot

→ characterized by having each of ~~[illegible]~~ its actuated joints be revolute joints.

→ ~~[illegible]~~ Like the Steward/Gough Platform, the Delta robot is also capable of a range of motions that includes all three degrees of positioning and also all three degree of orientation.

Serial Manipulator

→ Each actuated joint is attached to the previous actuated joint, or to the ground (if it's the first joint).

can be ceiling as well.

→ Each actuated joint is attached to the next actuated joint or the end-effector if it's the last actuated joint.

Parallel Manipulator Kinematics

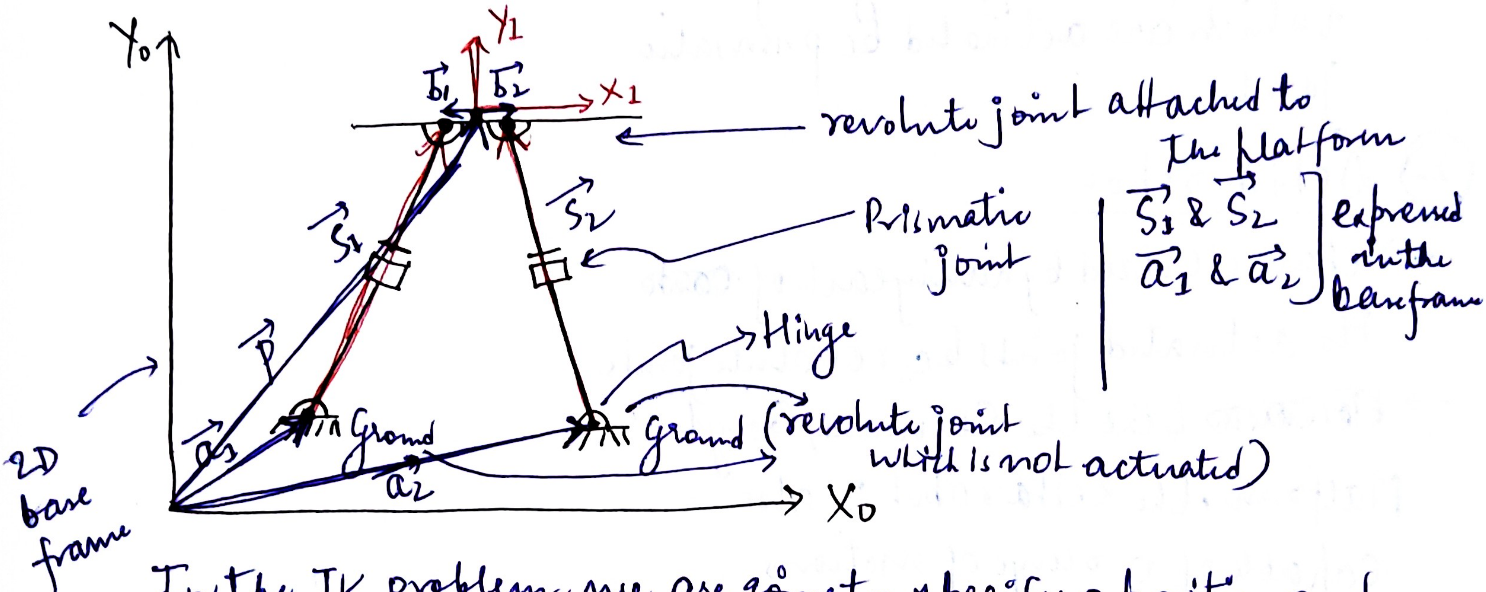
→ Forward Kinematics (difficult)
→ Inverse Kinematics (easy)

With Serial manipulator, FK is easy while IK is difficult
With parallel manipulator, FK is difficult while IK is easy.

⇓

If we ~~know~~ know the kind of motion we want the end-effector of our parallel manipulator to have then it's easy for us to find either the joint angles or the joint displacements that produce that kind of a motion for our manipulator.

Inverse Kinematics for the Stewart platform
(derived in 2D with 2DOF manipulator)



In the IK problem, we are going to specify a position and orientation for the end-effector (i.e., frame 1). We want to find the joint variables in this case the lengths of prismatic joints to get that pos & orient that we specified. This problem becomes much easier to solve if ~~we vector~~ try to solve it as a vector addition problem.

$|\vec{S_1}|$ & $|\vec{S_2}|$

$\vec{a_1}$ & $\vec{a_2}$ are vectors from frame 0 to the base location to the two joints

$\vec{P}$ ⇒ vector from center of frame 0 to centre of frame 1.

$\vec{P}$ is the position that we want the upper frame/platform to have. That's part of the input our IK problem, so we know what $\vec{p}$ is. We also know $\vec{a_1}$ & $\vec{a_2}$ since they are part of the robot design.

PM3

we also know the location of the attachment of both of the upper joints (attached to the ceiling) relative to the center of the frame 1.

i.e., $\vec{b_1}$ & $\vec{b_2}$ are also know since they are part of our robot design.

These two vectors state where on the platform we've attached the joints.

Note :- $\vec{b_1}$ & $\vec{b_2}$ are not expressed in the base frame, they are expressed in frame 1.

So, we know where relative to the center of the platform/frame 1 we have attached the joints but we don't necessarily know where those joint attachments are relative to the ground.

$$\vec{a_2} + \vec{S_2} = \vec{p} + \vec{b_2}$$

expressed in frame 1 while others are expressed in frame 0.

$$\boxed{\vec{S_2} = \vec{p} + R\vec{b_2} - \vec{a_2}}$$

Output — tells us the length that we should set for leg 2 in order to achieve the desired position & orientation of the platform

desired position of the platform

Rotation of the platform

Here,
$\vec{p}$ & R are the inputs
$\vec{b_2}$ & $\vec{a_2}$ are the parameters we set in building this robot

Expressing vectors in different frame

$$R_1^0 \vec{b_2} = \vec{b_2}$$

expressed in frame 1 — expressed in frame 0

Generalize Equation

$$\vec{S_i} = \vec{p} + R\vec{b_i} - \vec{a_i}$$

(leg i)

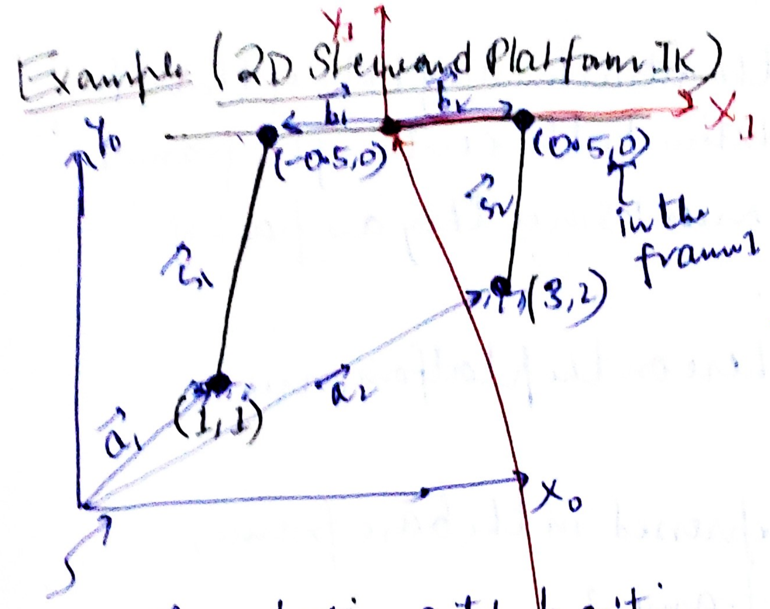
(position vector of the center of the platform)

Rotation matrix that we set for the platform

the vector betn the center of the platform and position where leg is attached to the platform for leg i

(vector betn the center of the base frame and position where the leg i joint is ~~attached to the ground~~)

Example (2D ~~Steward~~ Platform IK)



Let suppose first joint is connected to the base at (1,1) and second joint is connected to the base at (3,2). ~~Apart from these~~ The other two design variables that we have to specify is the location where the two joints ~~are~~ are attached to the platform.

The rotation & the position that we want the platform to be in:-

(desired position of the platform)

$$\vec{p} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

In other words, we want above location/position (marked with red) in the base frame (X_0, Y_0) to be (2,4).

desired rotation matrix for the platform relative to the base frame

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$ (no rotation, hence Identity) We want no rotation.

The lengths are what we send to our controllers to command the joint in order to expand or contract the amount we want it to.

$$|\vec{S_1}| = \sqrt{(0.5)^2 + (3)^2} = \sqrt{9.25} \approx 3.04$$

$$|\vec{S_2}| = \sqrt{(-0.5)^2 + 2^2} = \sqrt{4.25} \approx 2.06$$

Equation for IK

$$\vec{S_i} = \vec{p} + R\vec{b_i} - \vec{a_i}$$

$$\vec{S_1} = \begin{bmatrix} 2 \\ 4 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} -0.5 \\ 0 \end{bmatrix}}_{\vec{b_1}} - \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{\vec{a_1}}$$

$$= \begin{bmatrix} 2 \\ 4 \end{bmatrix} + \begin{bmatrix} -0.5 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5 \\ 3 \end{bmatrix}$$

$$\vec{S_2} = \vec{p} + R\vec{b_2} - \vec{a_2}$$

$$= \begin{bmatrix} 2 \\ 4 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} - \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \\ 4 \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} - \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} -0.5 \\ 2 \end{bmatrix}$$

(so vector from the base of joint 2 to the platform is -ve 0.5 in the X direction & 2 in the Y direction)

Not part of same notes

EXTRA

How to assign Denavit–Hartenberg frames to Robotic Arms?

(Jacques Denavit & Richard S. Hartenberg) → In ME, DH parameters are the four parameters associated with a particular convention for attaching reference frames to the links of a spatial kinematic chain or robotic manipulator.

→ In D-H convention, coordinate frames are attached to the joints between two links such that one transformation is associated with the joint, [Z] and the second is associated with the link [X].

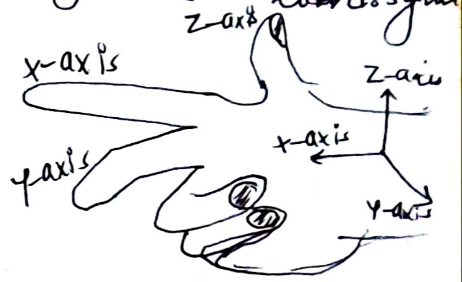
4 rules of the D-H Convention

(I) Z-axis is the axis of rotation for a revolute joint.

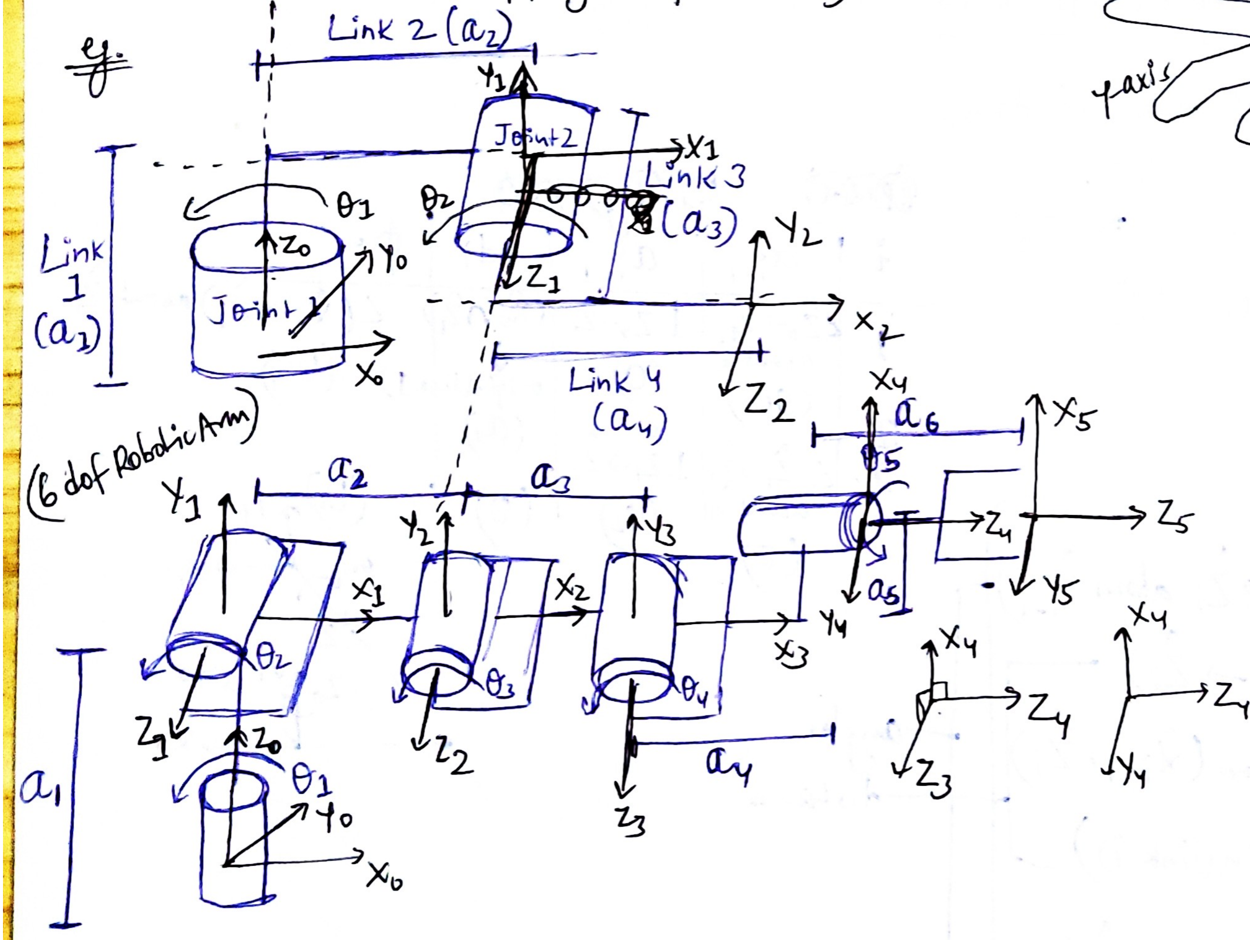
(II) X-axis must be ⊥ to both the current Z-axis & prev Z-axis.

(III) Y-axis is determined from X-axis & Z-axis by using the right-hand coord. system

(IV) X-axis must intersect the prev Z-axis (rule does not apply to frame 0)



e.g.



How to find DH parameter tables?

3 steps for finding the DH parameter table and the homogeneous transformation matrices for a robotic manipulator —

Homogeneous transformation matrices enable us to express the position and orientation of the end effector frame (eg. robotic gripper, hand, vacuum suction cup etc.) in terms of the base frame.

(I) Draw the kinematic diagram a/c to the 4 DH rules.

(II) Create the DH parameter table

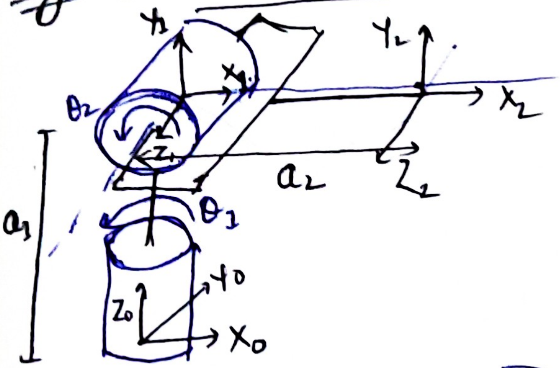
Number of rows = number of frames − 1

Number of columns = 4 ⇒ two cols for rotation & two columns for displacement.

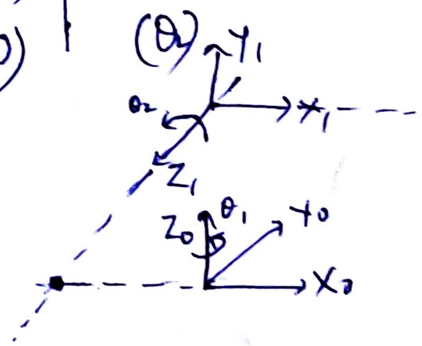
(III) Find the homogeneous transformation matrix.

Note – for more, refer Modern Robotics – Appendix C

Eg (1) 2 DOF Robotic Arm



i	α_{i-1}	a_{i-1}	d_i	ϕ_i
1	$\angle \hat{Z}_0 \to \hat{Z}_1$ about $\hat{X}_0$ (90°)	$\perp \hat{Z}_0 \to \hat{Z}_1$ (0)	$(\hat{X}_0 \cap \hat{Z}_1)$ ↓ origin(link 1) (a_1)	$\angle(\hat{X}_0 \to \hat{X}_1)$ about $\hat{Z}_1$ (θ_1)
2	$\angle \hat{Z}_1 \to \hat{Z}_2$ about $\hat{X}_1$ (0)	$\perp \hat{Z}_1 \to \hat{Z}_2$ (a_2)	(0)	(θ_2)



α_{i-1} $(\hat{Z}_{i-1} \to \hat{Z}_i$ about $\hat{X}_{i-1})$

a_{i-1} $(\perp \hat{Z}_{i-1} \to \hat{Z}_i)$

d_i (Intersection $(\hat{X}_{i-1}, \hat{Z}_i)$ ↓ origin(link i))

ϕ_i $(\hat{X}_{i-1} \to \hat{X}_i$ about $\hat{Z}_i)$

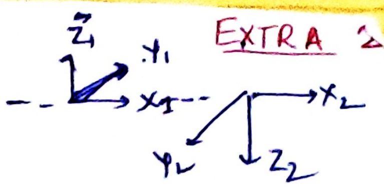
— angles

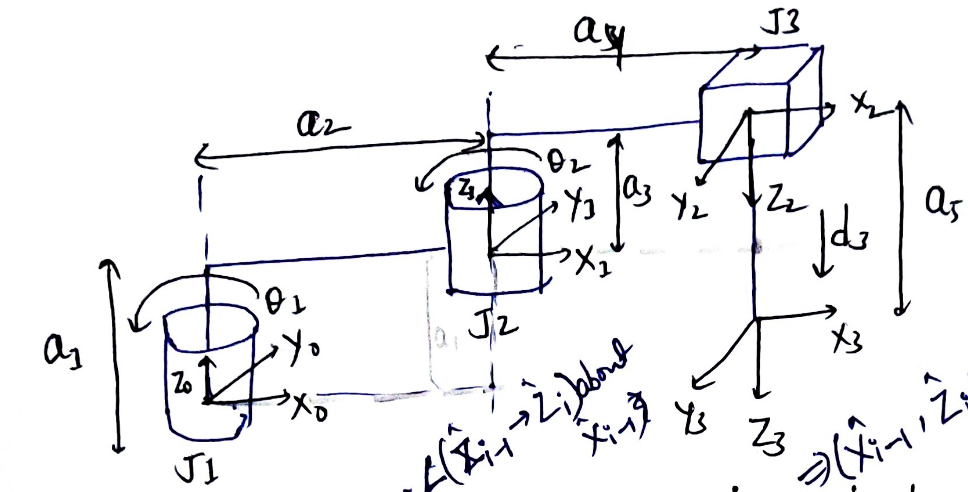
— distances

Not part of same notes

EXTRA 2

Eg ② SCARA Robot



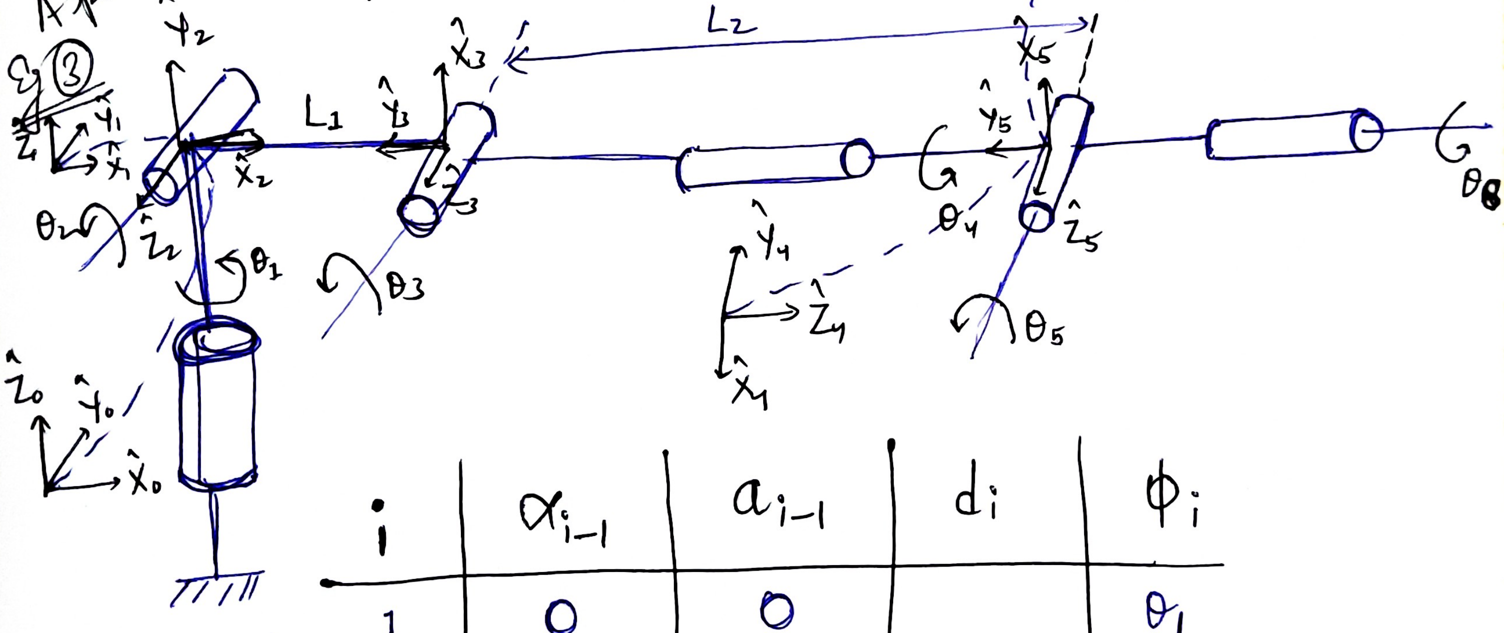


$\angle(\hat{Z}_{i-1} \to \hat{Z}_i)$ about $\hat{X}_{i-1}$

$\Rightarrow(\hat{X}_{i-1}, \hat{Z}_i) \to$ origin (link i)

i	α_{i-1}	a_{i-1} ($\perp \hat{Z}_{i-1} \to \hat{Z}_i$)	d_i	ϕ_i ($\hat{X}_{i-1} \to \hat{X}_i$ about $\hat{Z}_i$)
1	0	a_2	a_1	$0+\theta_1=\theta_1$
2	180	a_4	a_3	$0+\theta_2=\theta_2$
3	0	0	a_5+d_3	0

Eg ③ A Spatial 6R open Chain



i	α_{i-1}	a_{i-1}	d_i	ϕ_i
1	0	0		θ_1
2				
3				
4				
5				
6				

$\alpha_{i-1} \to \angle(\hat{Z}_{i-1} - \hat{Z}_i)$ about X_{i-1}

$a_{i-1} \to (\perp \hat{Z}_{i-1} \to \hat{Z}_i)$

$d_i \to \cap(\hat{X}_{i-1}, \hat{Z}_i) \to$ origin (link i)

ϕ_i ($\hat{X}_{i-1} \to \hat{X}_i$ about $\hat{Z}_i$)