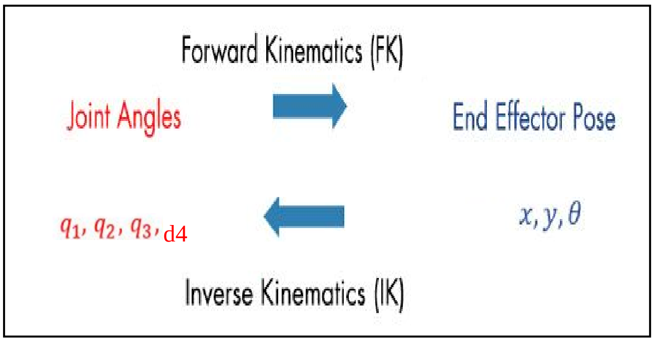
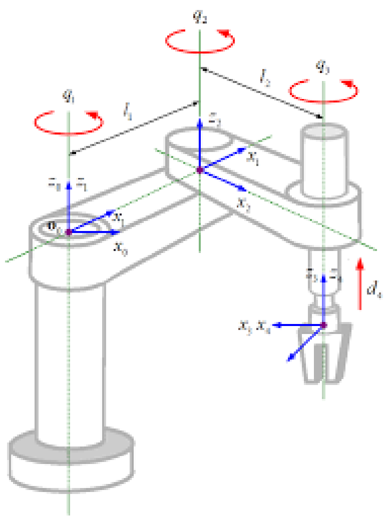
Project 2: Kinematic and Structural Analysis of SCARA Robot.

Abstract:

An **industrial robot** is comprised of a **robot manipulator**, power supply, and controllers. Robotic manipulators can be divided into two sections, each with a different function: **Robot Arm** and Body. Robot manipulators are created from a sequence of **link** and **joint** combinations. The links are the rigid members connecting the joints or axes. There are 2 types of joints: **Linear joint** and **Rotary joint**. The rotary joint consists of the **Revolute joint (R)** and a **Twisting joint (T)**. The linear joint consists of a **Prismatic joint (P)** and a **Sliding joint (S).** Each manipulator has its **DOF** (Degree of freedom), the number of degrees of freedom is equal to the **total number of independent movements** that an object can perform. Pick and place task is one of the most important tasks in the industrial field handled by the "**Selective Compliance Assembly Robot Arm**" (SCARA). Repeatability with high-speed movement in the horizontal plane is a remarkable feature of this type of manipulator. The challenge of designing SCARA is the difficulty of achieving stability of high-speed movement with a long length of links. Shorter links arm can move more stable. This condition made the links should be considered restricted then followed by restriction of operation area (workspace). And after loading the robot with its end effector there will be deformation and the end effector cannot reach the desired location because of a change in position and orientation due to deformation. So, we need to define the load within the limits. In this project, we are using a **SCARA ROBOT** whose **DOF** is **4** and consists of **3 Rotary joints and 1 Linear Joint**. We use the link and joint parameters to carry out the manipulator's kinematic analysis (Kinematics). We obtain the **position and orientation** information of the end-effector of a manipulator of known geometry for a given set of joint angles in **forward kinematics**. On the other hand, in **inverse kinematics**, we **determine the joint angles** of a manipulator of known geometry for a given position and the orientation information of its end-effector. The kinematic equations between the working space and the joint space are deduced by the homogeneous transformation principle. In the present project, kinematic analysis is done using **ROBOANALYZER** software to find out the position of the tool or end effector for different and joint variables and we use **ANSYS WORKBENCH** Software for Structural Analysis of SCARA Robot.



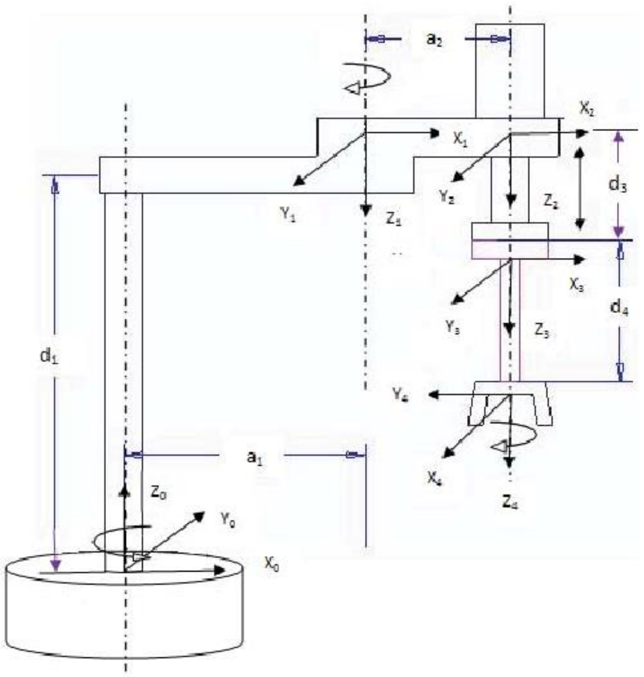


<u>Introduction to SCARA Robot</u>:

The **SCARA** acronym stands for **Selective Compliance Assembly Robot Arm** or Selective Compliance Articulated Robot Arm. SCARA is a type of **Industrial robot**.

The SCARA robot is most used for pick-and-place or assembly operations where high speed and high accuracy are required. Generally, a SCARA robot can operate at a higher speed and with optional cleanroom specifications.

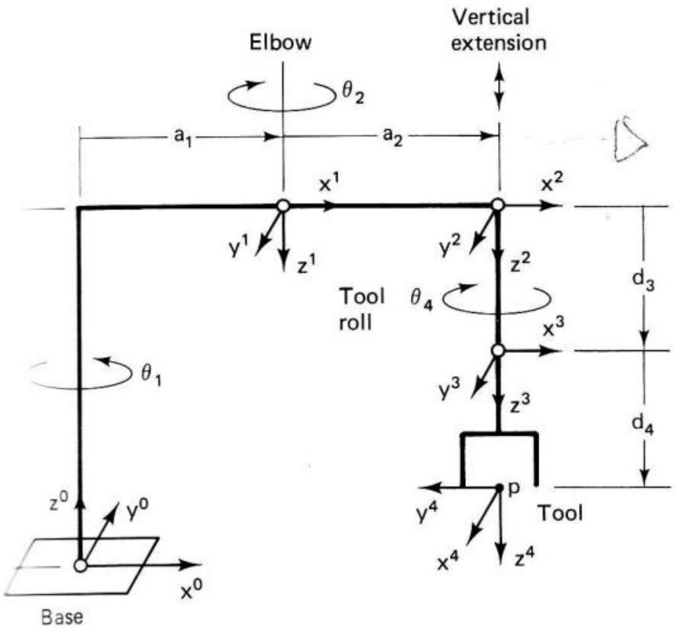
Industrial robots are defined as ‘multi-functional manipulators designed to move parts through various programmed motions. As such, robots provide **consistently reliable performance,** and **repetitive accuracy** and are able to handle heavy workloads perform in harsh environments. Additionally, robots can be **quickly reprogrammed** to reflect changes in production needs and cycles.



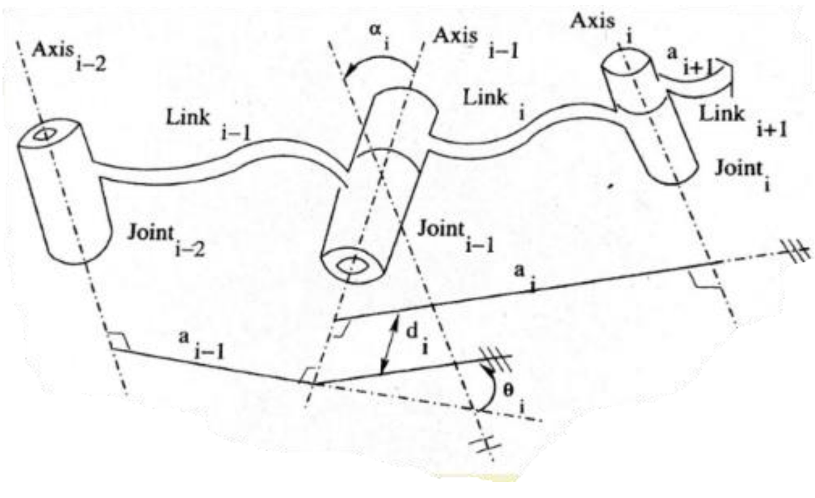
Four axis SCARA robot consists of an inner link that rotates about the World Z-axis, connected to an outer link that rotates about a Z elbow joint, which in turn is connected to a wrist axis that moves up and down and rotates about Z. An alternative configuration has the linear Z motion as the second axis. These are the most popular geometries for vertical assembly and small parts pick-and-place operations.

<u>Robot Kinematics</u>:

To specify the geometry of the planar 3R robot, we require **two parameters, a1 and a2.** These are the **two link lengths**. In the figure, the three **joint angles** are labeled **θ1, θ2, and θ4 and d3**. These are obviously **variable**.



The precise definitions for the link lengths and joint angles are as follows.



Length of link i (ai): It is the mutual perpendicular distance between Axisi-1 and Axisi

Angle of twist of link i (αi): It is defined as the angle between Axisi-1 and Axisi

Offset of link i (di): It is the distance measured from a point where ai-1 intersects the Axisi-1 to the point where ai intersects the Axisi-1 measured along the said axis

Joint Angle (θi): It is defined as the angle between the extension of ai-1 and ai measured about the Axis Notes: i-1

Another set of variables that is useful to define is the set of coordinates for the end effector. These coordinates define the position and orientation of the end effector. With a convenient choice of a reference point on the end effector, we can describe the position of the end effector using the coordinates of the reference point (x, y) and the orientation using the angle f. The three end effector coordinates (x, y, f) completely specify the position and orientation of the end effector.

Homogenous Transformation Matrix

To keep representation matrices square, if we represent both orientation and position in the same matrix, we will add the scale factors to the matrix to make it a 4 X 4 matrix. If we represent the orientation alone, we may either drop the scale factors and use 3x 3 matrices or add a 4th column with zeros for a position to keep the matrix square. Matrices of this form are called homogeneous matrices, and we write them as follows:

F is a 4x4 matrix that can describe a translation, rotation, or both in one matrix

$$F = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Could be rotation around the z-axis, x-axis, y-axis, or a combination of the three.

$$R_z = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) \\ 0 & sin(\theta) & cos(\theta) \end{bmatrix} \qquad R_y = \begin{bmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{bmatrix}$$

$$\mathrm{Trans}_{z_{n-1}}(d_n) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_n \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

$$\mathrm{Rot}_{z_{n-1}}(\theta_n) = \left[\begin{array}{ccc|c} \cos\theta_n & -\sin\theta_n & 0 & 0 \\ \sin\theta_n & \cos\theta_n & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

$$\mathrm{Trans}_{x_n}(r_n) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & r_n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

$$\mathrm{Rot}_{x_n}(\alpha_n) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_n & -\sin\alpha_n & 0 \\ 0 & \sin\alpha_n & \cos\alpha_n & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

$${}^{n-1}T_n = \mathrm{Screw}_Z \text{ X } \mathrm{Screw}_X$$

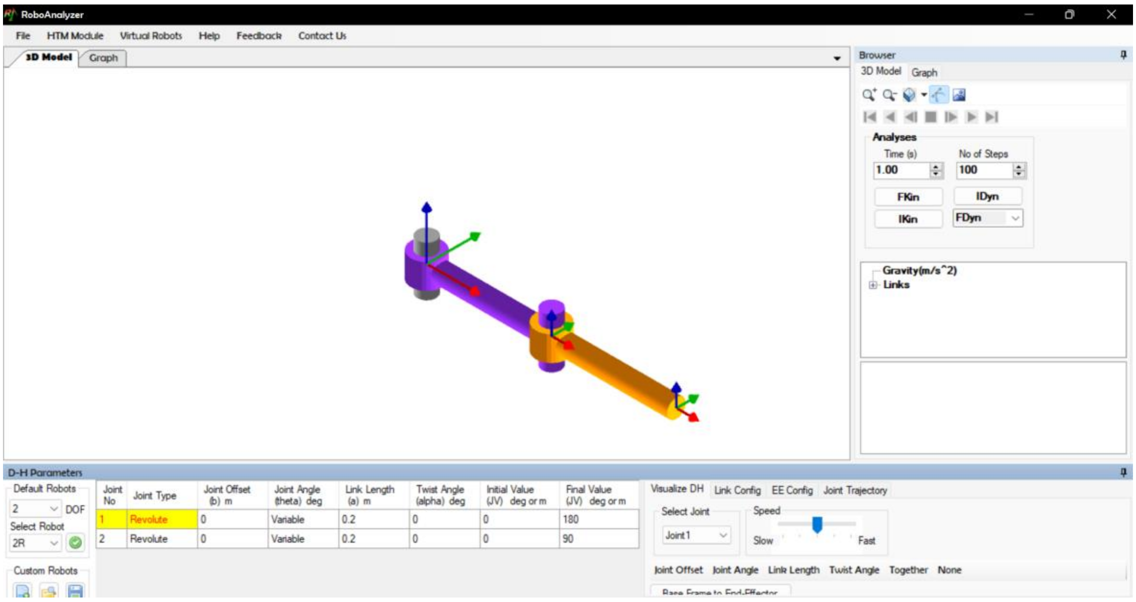
$${}^{n-1}T_n = \left[\begin{array}{ccc|c} \cos\theta_n & -\sin\theta_n\cos\alpha_n & \sin\theta_n\sin\alpha_n & r_n\cos\theta_n \\ \sin\theta_n & \cos\theta_n\cos\alpha_n & -\cos\theta_n\sin\alpha_n & r_n\sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc|c} & R & & T \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

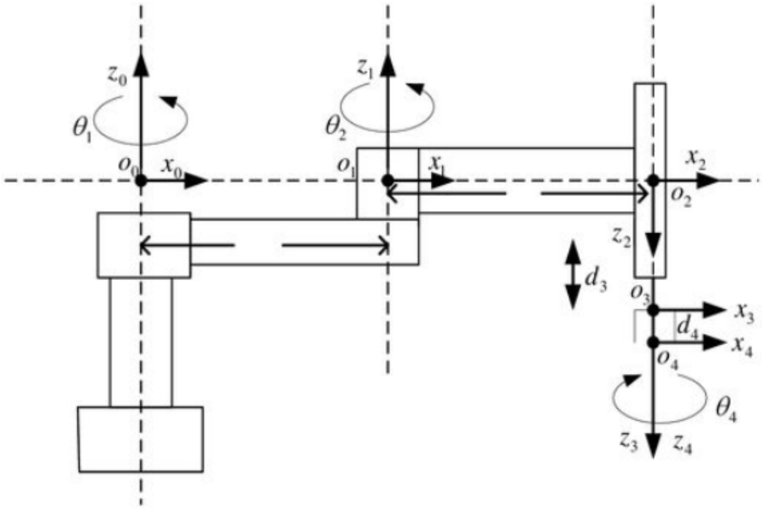
where R is the 3×3 submatrix describing rotation and T is the 3×1 submatrix describing translation.

Robo Analyzer:

RoboAnalyzer® is a 3D model-based software that can be used to teach and learn Robotics concepts.
It is an evolving product developed in the Mechatronics Lab, Department of Mechanical Engineering at IIT Delhi, New Delhi, India. Its development started under the guidance of Prof. S.K.Saha in order to support the learning/teaching of the topics covered in his book "Introduction to Robotics" published by Tata McGraw Hill, New Delhi.

Interface of Robo Analyzer Software





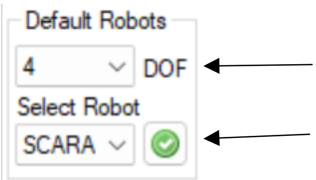
Joint Parameters

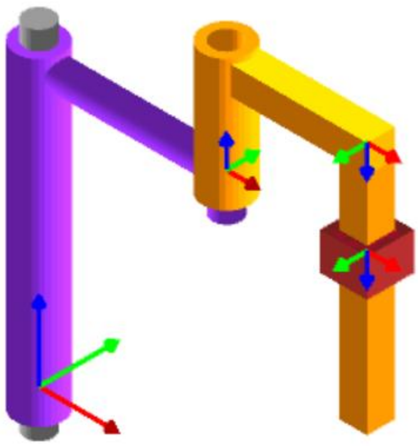
Joint(i)	Joint Type	Joint offset(b) meters	Joint angle(θ) degrees	Link length(a) meters	Link twist(α) degrees
1	Revolute	0.2	0 to 90	0.2	0
2	Revolute	0.1	0 to 90	0.1	180
3	Prismatic	0 to 0.1	0	0	0
4	Revolute	0	0 to 90	0	0

KINEMATIC ANALYSIS OF SCARA ROBOT using robo analyzer

3D Model of 3R robot

First, we must import the SCARA robot model into the scene so, we must change the Default option of Robot on the left side bottom-most bar. We must change the **DOF** of the robot to **4** and then we must select the **SCARA robot** then tap on the green tick mark to import our model.



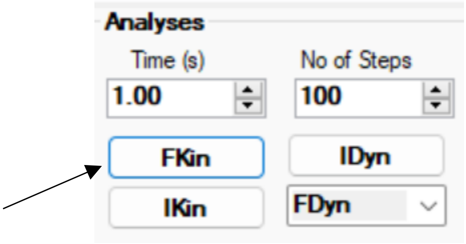


D-H Parameters

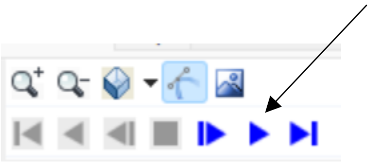
After importing the robot, we must change the joint parameters of our own values. Here I have measured the link lengths of the basic SCARA robot from the **Design Modeler IN ANSYS** and selected the parameters as follows.

Joint No	Joint Type	Joint Offset (b) m	Joint Angle (theta) deg	Link Length (a) m	Twist Angle (alpha) deg	Initial Value (JV) deg or m	Final Value (JV) deg or m
1	Revolute	0.2	Variable	0.2	0	0	90
2	Revolute	0.1	Variable	0.1	180	0	90
3	Prismatic	Variable	0	0	0	0	0.1
4	Revolute	0	Variable	0	0	0	90

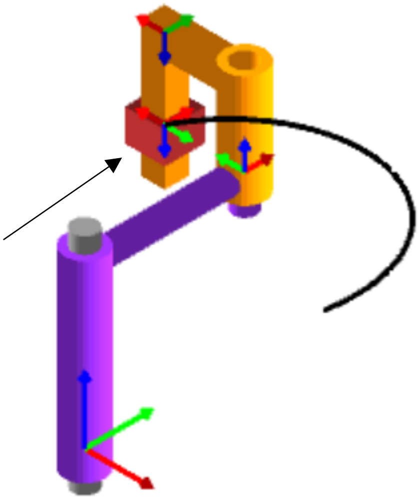
Analysis



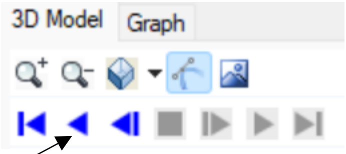
In the right-side bottom-most bar, we can see the analysis bar, we must tap on the **FKin (Forward Kinematics)** option to complete the analysis of Forward Kinematics.



After that, we must press the play button to analyze the Forward Kinematics of the robot and the robot will start moving and we can see in the below image that the end effector stops at the desired position of the given joint parameters data.

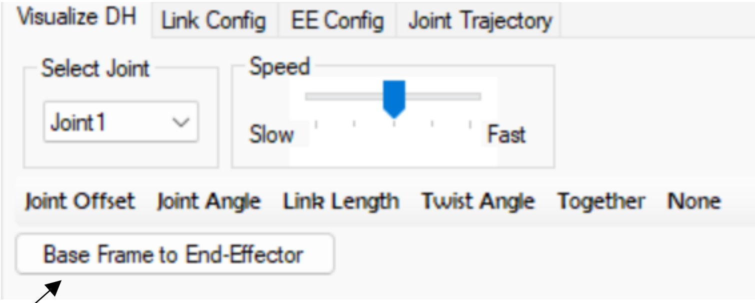


Isometric View of Robot (FKin)



Again, after pressing the rewind button, the robot comes to the initial position.

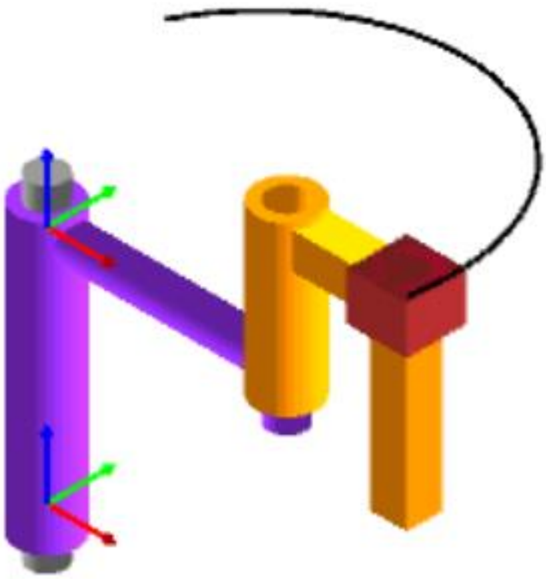
Frame Transformation



- **Joint 1**

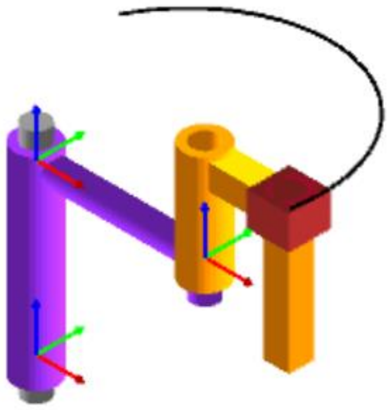
Joint offset = 0.2, so the frame of the axis move in Z-direction

Joint angle = 0 to 90 (variable)



Link length = 0.2m so the axis translates 0.2m along X-direction

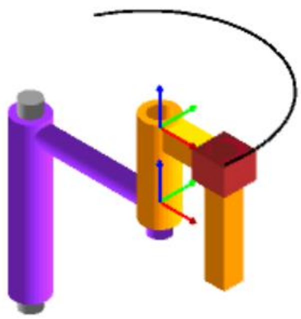
Link twist = 0, so the frame doesn't make an angle between the (Z)Axisi-1 and (Z)Axisi



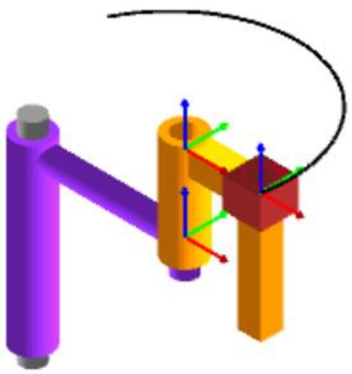
- **Joint 2**

Joint offset = 0.1, so the frame of the axis move in Z-direction

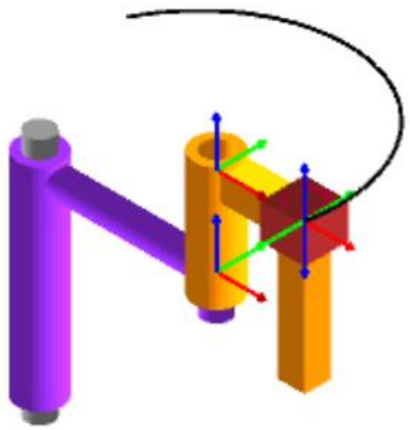
Joint angle = 0 to 90 (variable)



Link length = 0.1m so the axis translates 0.1m along X-direction



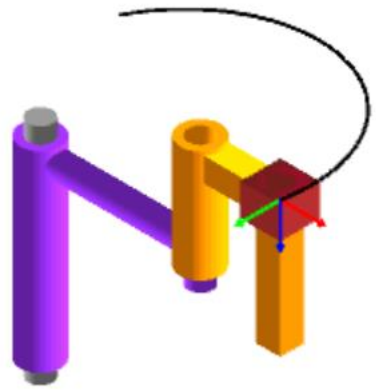
Link twist = 180, so the frame make an angle between the (Z)Axisi-1 and (Z)Axisi



- **Joint 3**

Joint offset = 0 to 0.1 (variable), so the frame of the axis moves in Z-direction

Joint angle = 0



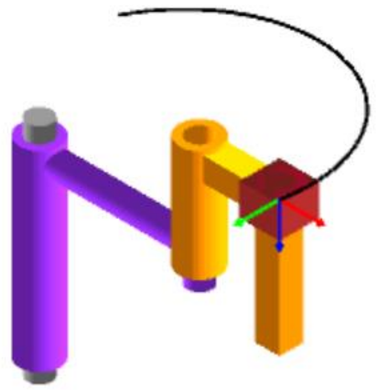
Link length = 0m so the axis doesn't translates along the X-direction

Link twist = 0, so the frame doesn't make an angle between the (Z)Axisi-1 and (Z)Axisi

- **Joint 4**

Joint offset = 0, so the frame of the axis doesn't move in the Z-direction

Joint angle = 0 to 90 (variable)



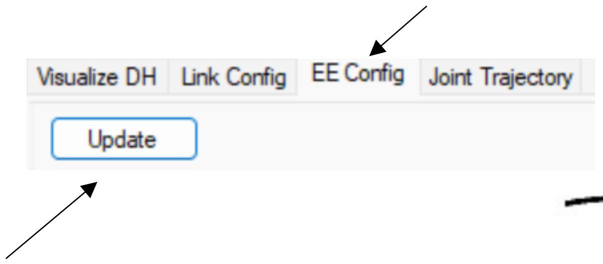
Link length = 0m so the axis doesn't translates along X-direction

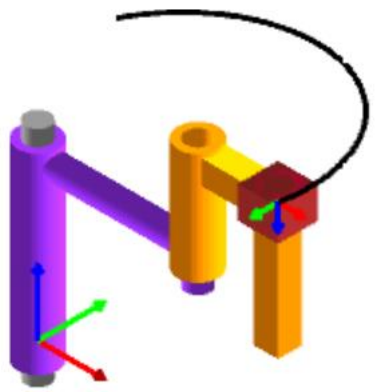
Link twist = 0, so the frame doesn't make an angle between the (Z)Axisi-1 and (Z)Axisi

End Effector Configuration

- **Initial Configuration**

To obtain the end effector configuration we need to tap on **EEConfig** which is in the bottom of the right side and then we need to tap on update.





After tapping on update we will get the initial end effector configuration or transformation matrix

Rotation Matrix (There is a rotation of the end effector)

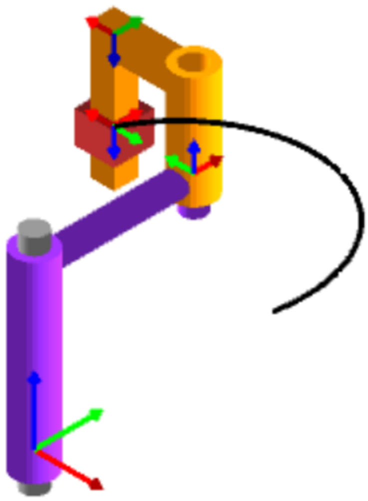
$$\begin{bmatrix} 1 & 0 & 0 & 0.3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Translation Matrix (End effector is (0.3m,0m,0.3m) distance away from the base frame at initial configuration)

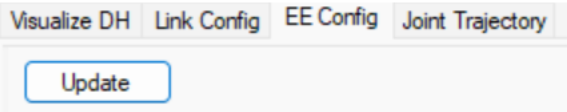
- **Final Configuration**

Now to obtain the final end effector configuration we need to move the robot's end effector to the final position by applying the forward kinematics of the given join parameters data.

After applying the Forward Kinematics, the robot comes into the desired end position.



To obtain the final end effector configuration we need to tap on **EEConfig** which is in the bottom of the right side and then we need to tap on update.



After tapping on the update, we will get the final end effector configuration or transformation matrix

Rotation Matrix (The final frame is oriented about the Z axis w.r.t base frame)

$$\begin{bmatrix} \boxed{\begin{matrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{matrix}} & \boxed{\begin{matrix} -0.1 \\ 0.2 \\ 0.2 \end{matrix}} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{bmatrix}$$

Translation Matrix (End effector is (-0.1m,0.2m,0.2m) distance away from the base frame at initial configuration)

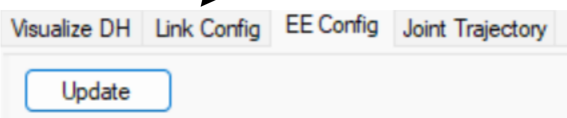
Link Configuration

We know that

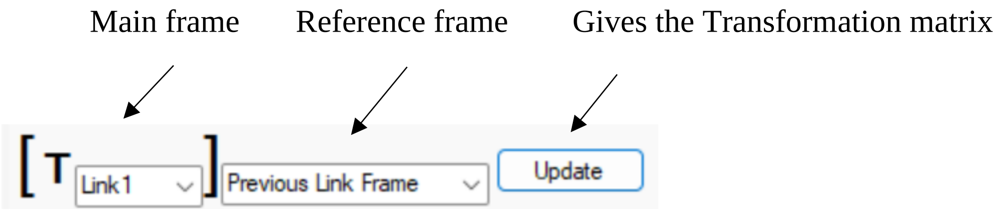
- ⇨ transformation matrix of link 3 w.r.t base frame = transformation matrix of link 1 w.r.t base frame X transformation matrix of link 2 w.r.t 1 X transformation matrix of link 3 w.r.t 2 X transformation matrix of link 4 w.r.t 3

- ⇨ ${}^0T_3 = {}^0T_1 x {}^1T_2 x {}^2T_3 x {}^3T_4$

To operate the link configuration, we need to tap on **Link Config**



After that, we must select the main frame and reference frame accordingly, and then we need to update them to get the Transformation matrix.



- **Link 1 w.r.t Base frame**

$$\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0.2 \\ 0 & 0 & 1 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **Link 2 w.r.t 1**

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0.1 \\ 0 & 0 & -1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **Link 3 w.r.t 2**

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **Link 4 w.r.t 3**

[T Link4] Previous Link Frame

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

- **Link 4 w.r.t base frame**

[T Link3] Base Frame Update

$$\begin{bmatrix} 0 & 1 & 0 & -0.1 \\ 1 & 0 & 0 & 0.2 \\ 0 & 0 & -1 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Hence,

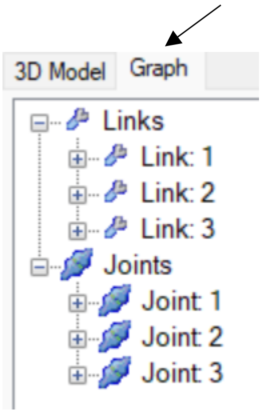
$$\begin{bmatrix} 0 & 1 & 0 & -0.1 \\ 1 & 0 & 0 & 0.2 \\ 0 & 0 & -1 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ({}^{0}T_{4})$$

$$\Updownarrow$$

$$\underset{({}^{0}T_{1})}{\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0.2 \\ 0 & 0 & 1 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}} \times \underset{({}^{1}T_{2})}{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0.1 \\ 0 & 0 & -1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}} \times \underset{({}^{2}T_{3})}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}} \times \underset{({}^{3}T_{4})}{\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

Graphs

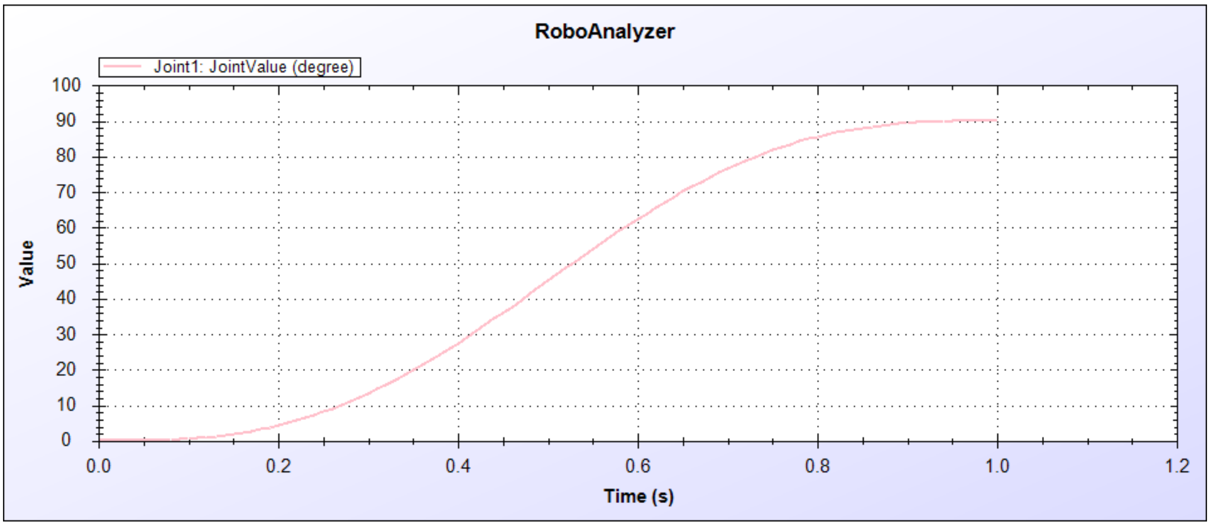
To get the different graphs of joints (joint angle, joint velocity, etc..) and links in robo analyzer 1st we must tap on the **FKin** option, and then we need to tap on the **Graph** option which is on the top right side. Then we need to select the different options which we see in the below image to access different graphs



Joint Angle Graphs

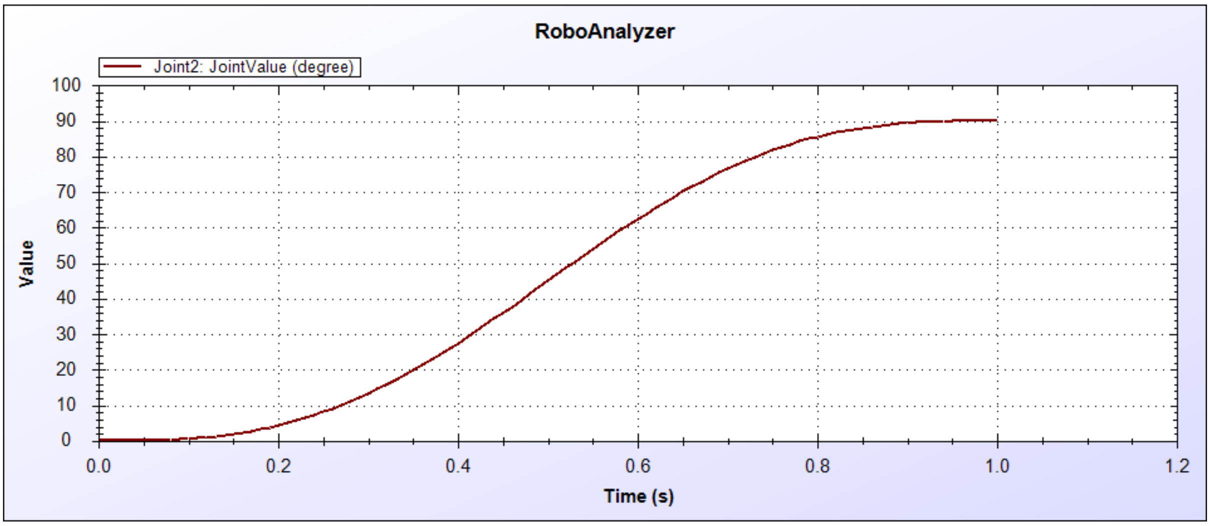
- **Joint 1**

  The joint angle of joint 1 is variable as time increases the movement of the joint starts from 0 degrees and ends at 90 degrees within 1sec and the graph will be a nonlinear curve.



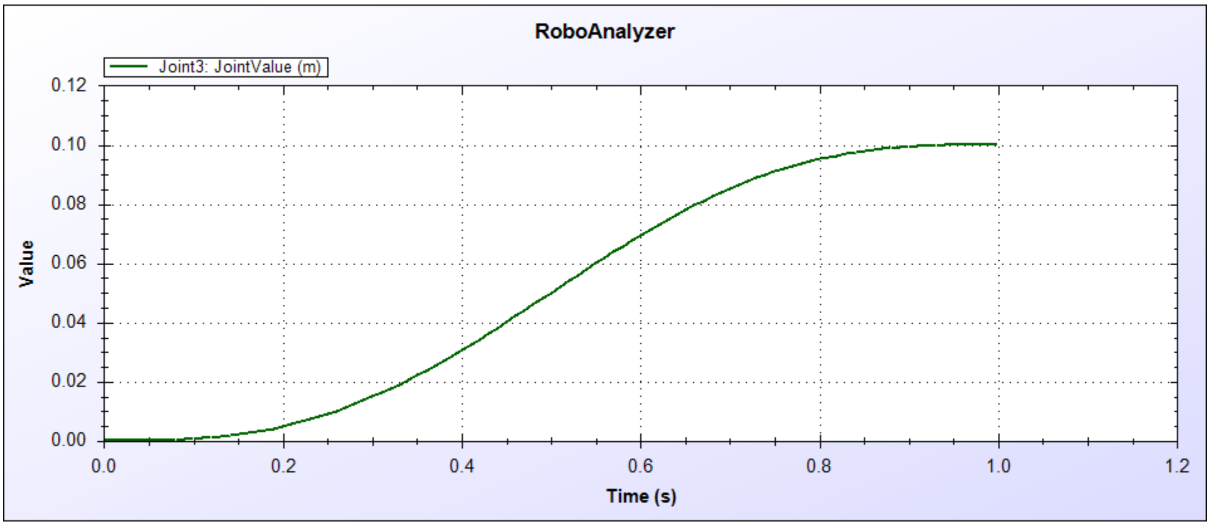
- **Joint 2**

The joint angle of joint 2 is variable as time increases the movement of the joint starts from 0 degrees and ends at 90 degrees within 1sec and the graph will be a nonlinear curve.



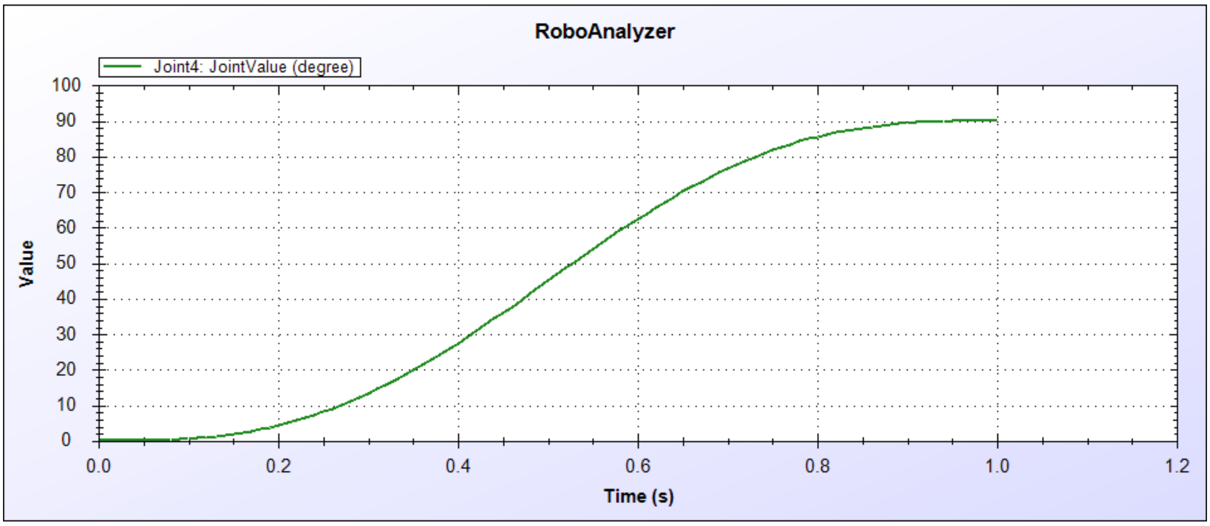
- **Joint 3**

The joint offset of joint 3 is variable as time increases the movement of the joint offset starts from 0m and ends at 0.1m within 1sec and the graph will be a nonlinear curve.



- ### Joint 4

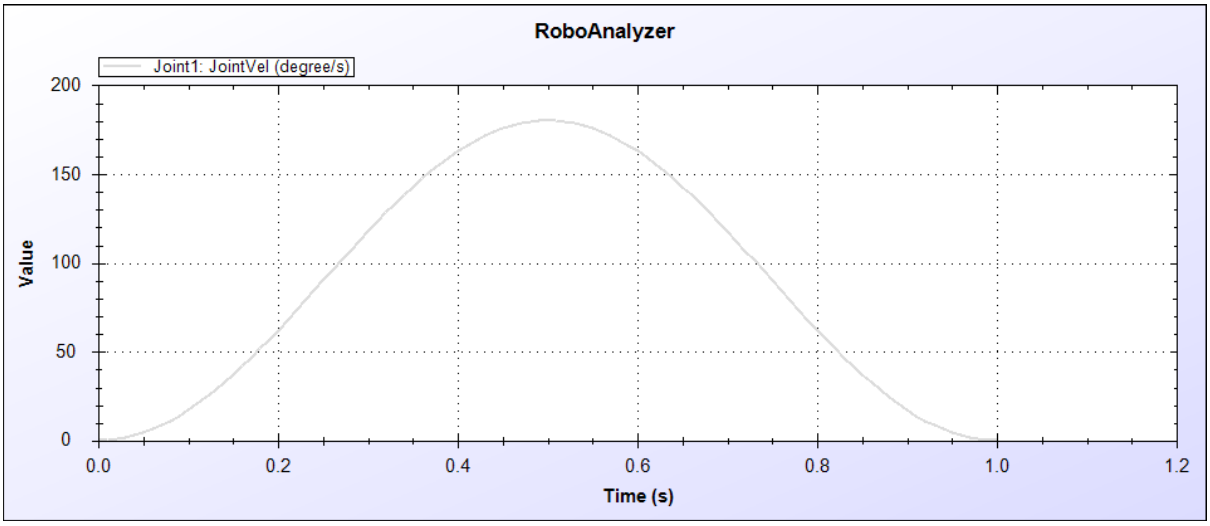
The joint angle of joint 4 is variable as time increases the movement of the joint starts from 0 degrees and ends at 90 degrees within 1sec and the graph will be a nonlinear curve.



Joint Velocity Graphs

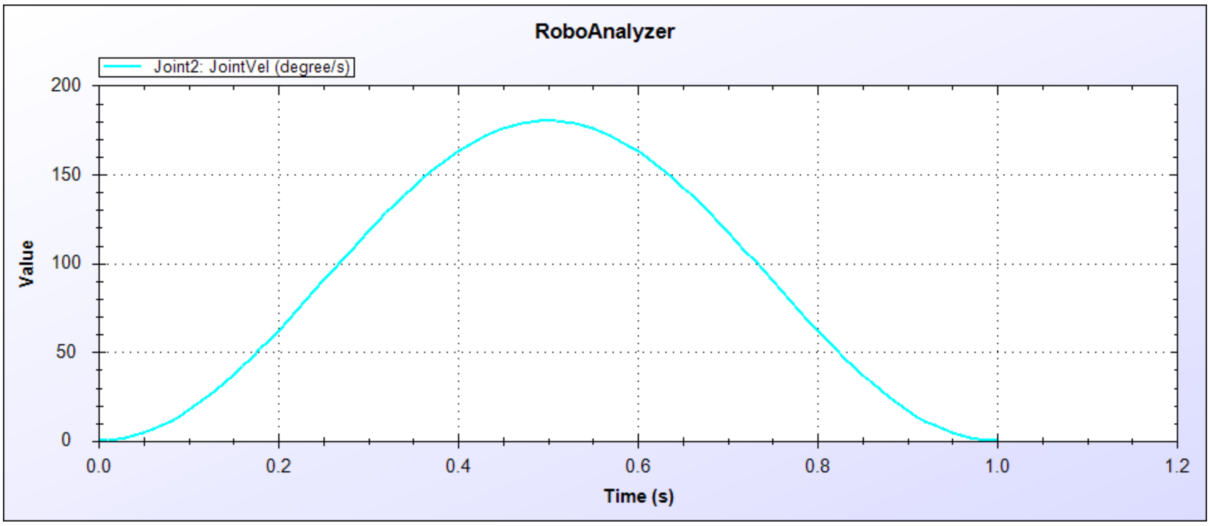
- ### Joint 1

The joint velocity of joint 1 is increasing as time increases up to half of the total period of settling time and then the velocity of the joint starts decreasing gradually within 1sec, and the graph will be an exponential curve.



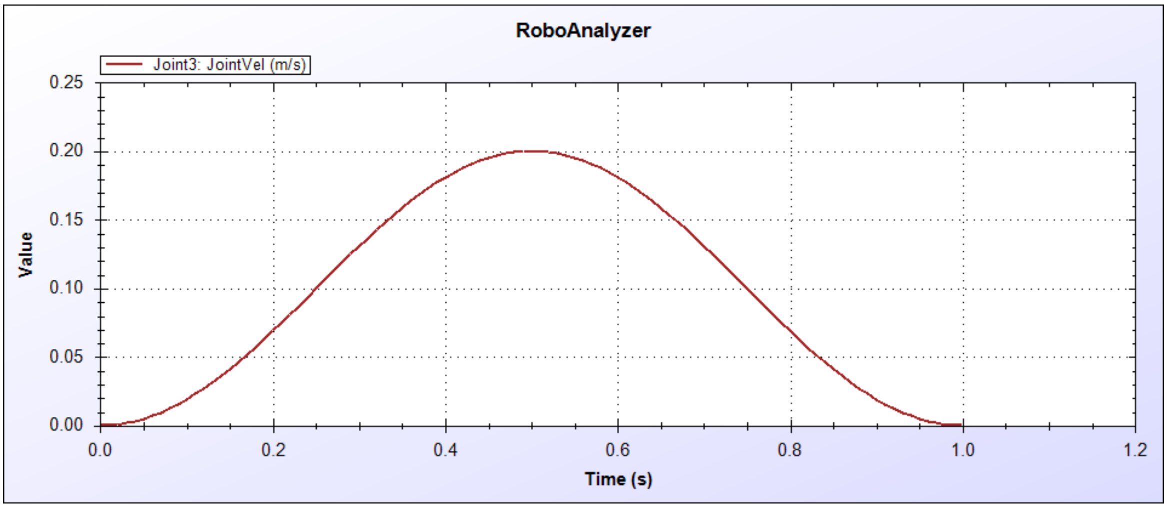
- ## Joint 2

The joint velocity of joint 2 is increasing as time increases up to half of the total period of settling time and then the velocity of the joint starts decreasing gradually within 1sec, and the graph will be an exponential curve.



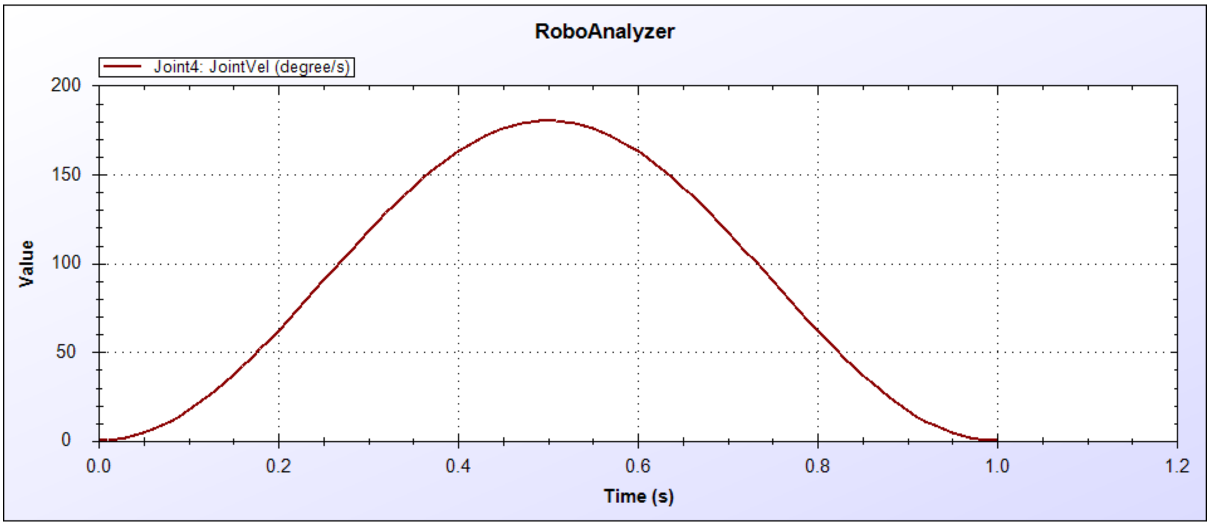
- ## Joint 3

The joint velocity of joint 3 is increasing as time increases up to half of the total period of settling time and then the velocity of the joint starts decreasing gradually within 1sec, and the graph will be an exponential curve.



- **Joint 4**

  The joint velocity of joint 4 is increasing as time increases up to half of the total period of settling time and then the velocity of the joint starts decreasing gradually within 1sec, and the graph will be an exponential curve.

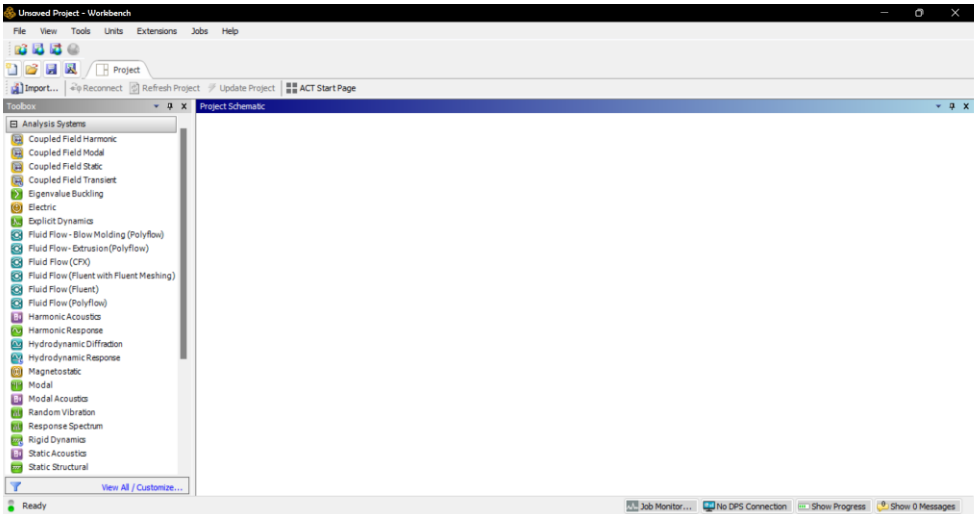


Ansys Workbench

ANSYS is a general-purpose, **finite-element modeling** package for numerically solving a wide variety of mechanical problems. These problems include static/dynamic, **structural analysis**, heat transfer, and fluid problems, as well as acoustic and electromagnetic problems.

Ansys has all the tools you need for complete, system-level modeling of a robot. The **mechanism's structure** can be modeled using Ansys Mechanical and Ansys Motion.

Interface of Ansys Software



<u>Structural analysis using Ansys</u>

If we apply a load to the end effector there will be a small deformation due to the **load increase** in a particular component. If the **deformation increases** gradually then the position and orientation of the end effector of the robot will not match with the desired position i.e., the **accuracy** of the robot becomes **lower**. So that we need to ensure that the deformation is very small in that robot during the loading of the end effector and based on that we need to check whether the robot is reaching the desired position or not. So, we need to define the **load** within the **limits**.

During **structural analysis**, we need to **calculate the deformation** in a particular component due to the presence of load.

Analysis is divided into Pre-processing stage, Processing Stage, and Postprocessing Stage.

Pre-processing Stage

We need to **assign the material properties** i.e., we need to consider the materials to particular components. We need to perform the **meshing** i.e., we need to divide the total component into small elements. We need to apply the **boundary conditions** i.e., which type of loads applying on the particular component and also which type of supports are present on the component.

Processing stage

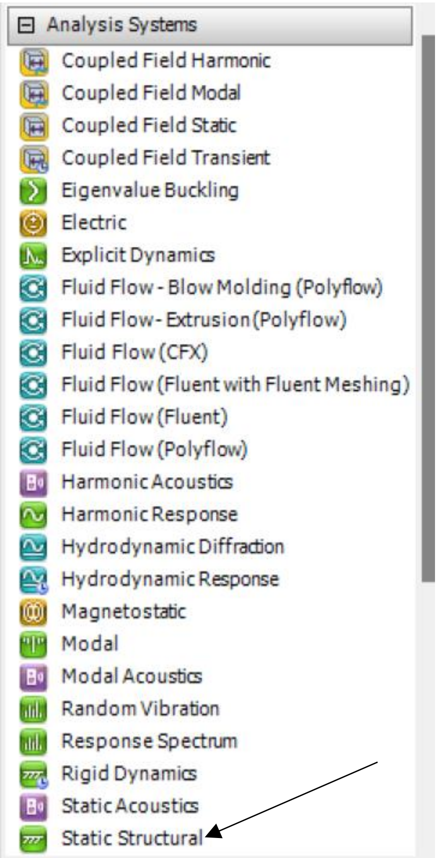
All the particular **solutions of equilibrium** and matrices which are needed were calculated internally.

Postprocessing stage

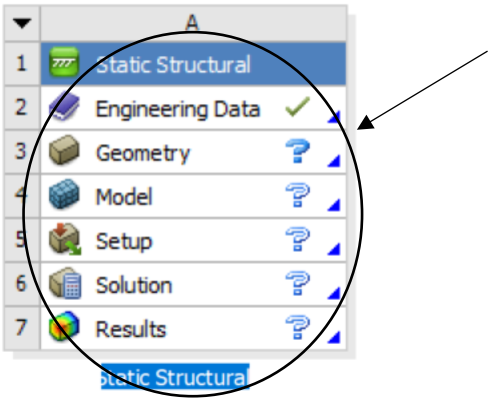
After completion of the calculation part, we will view the **result** in the post-processing stage.

Due to the presence of load if the component has very **small deformation**, then it will be considered **normal deformation** and if it has **large deformation** then we need to **reduce the load** on a particular component. We have to check for the given load the SCARA robot is withstanding or not at a particular configuration.

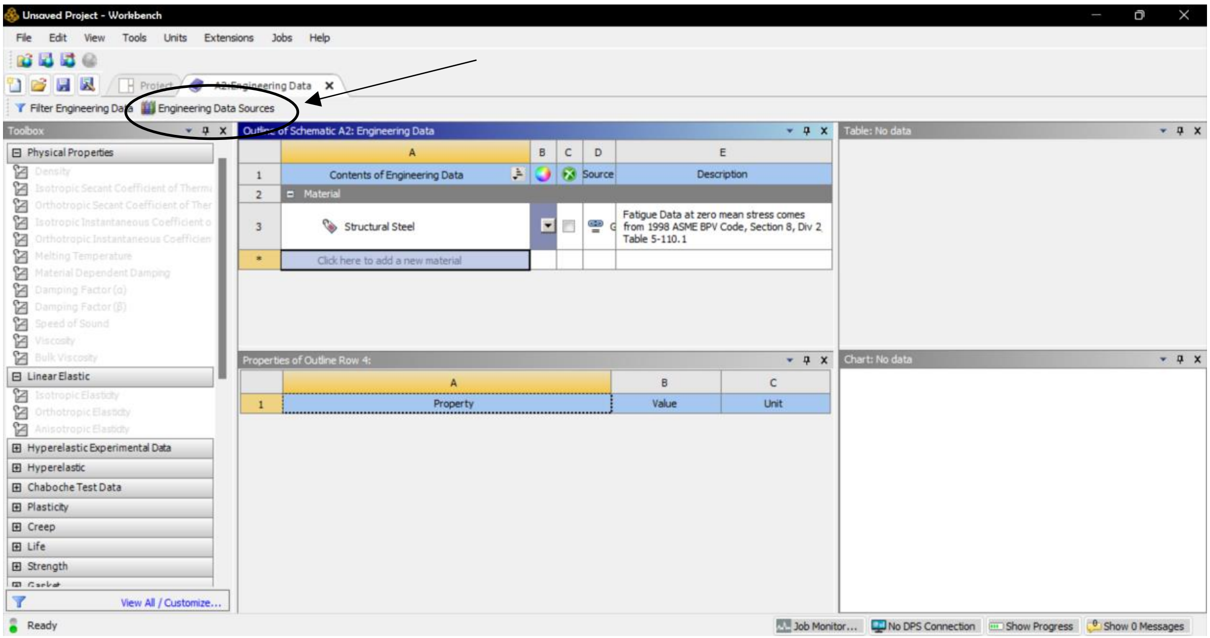
To begin Structural Analysis, we need to tap on the **Static Structural** option which is on the bottommost left side of the interface.



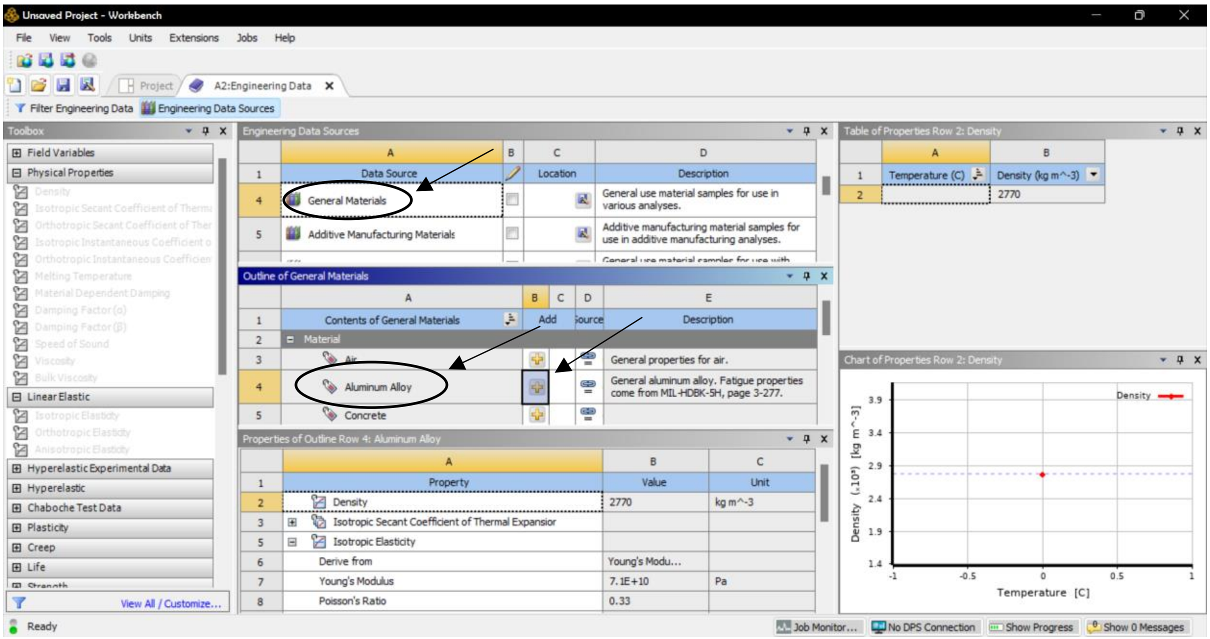
Then we will see these options on the screen



First, we need to add the **Material Properties** in the components so, we need to tap on the **Engineering Data** option. Now we will see this page, then we have to tap on **Engineering Data Sources.**

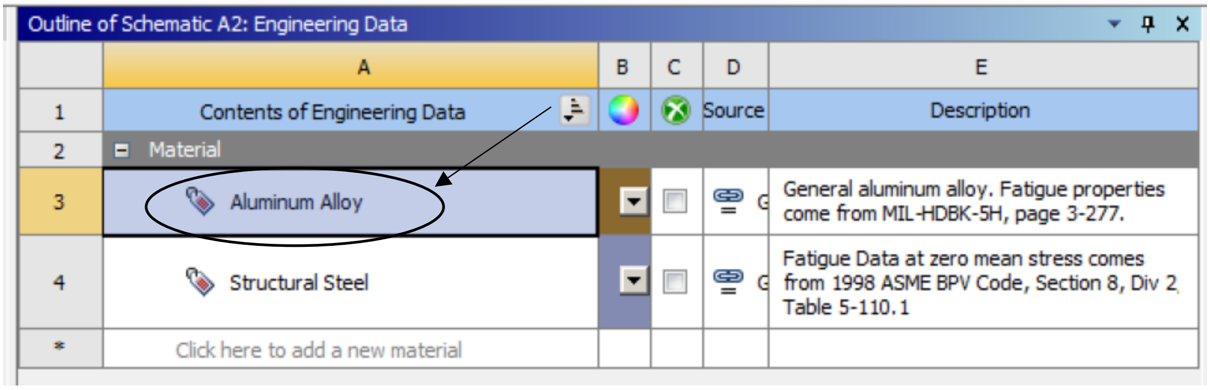


Now we must select the Aluminium Alloy which is present in the library. So, now we must select the **General Materials** from the Data Source, and after selecting general materials next, we must select **Aluminium Alloy** from the **Contents of General Materials**. Then we must click on the button then it will show as bookmarked .

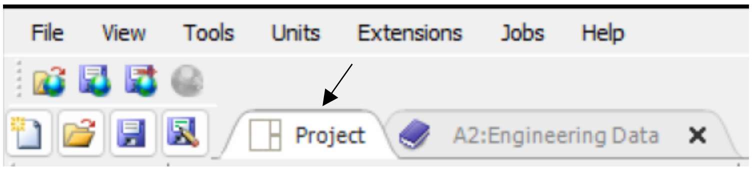


By default, we have Structural Steel in components now we included Aluminium Alloy (Low density and Lightweight).

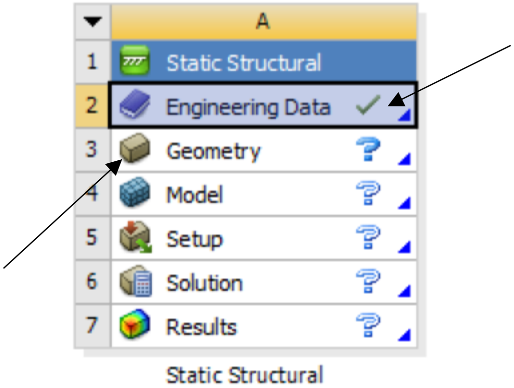
Now we must tap on **Engineering Data Sources** so we can confirm that we have added the aluminium alloy,



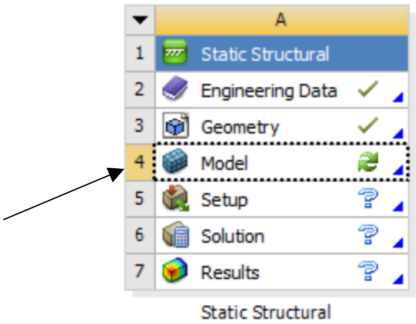
and then we should tap on the **Project** option which is on top.



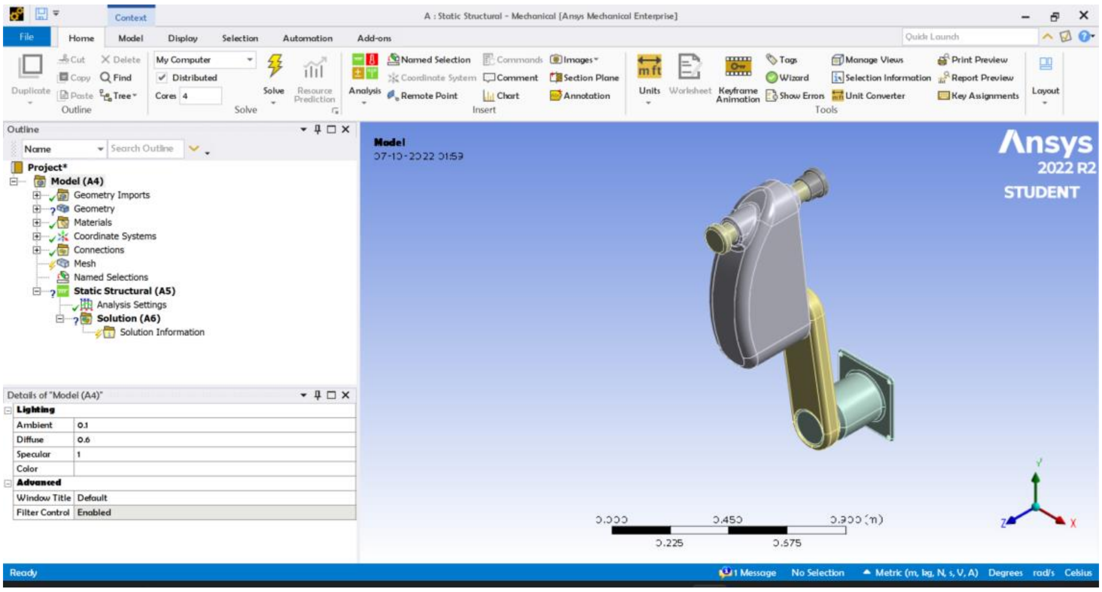
So finally, we have assigned the Engineering Data. So, after completion of the process, there will be a tick mark generated automatically. Now we have to move to **Geometry** and we have to right-click on it to **import** the **SCARA Robot model**.



Now we have to double-click on **Model**

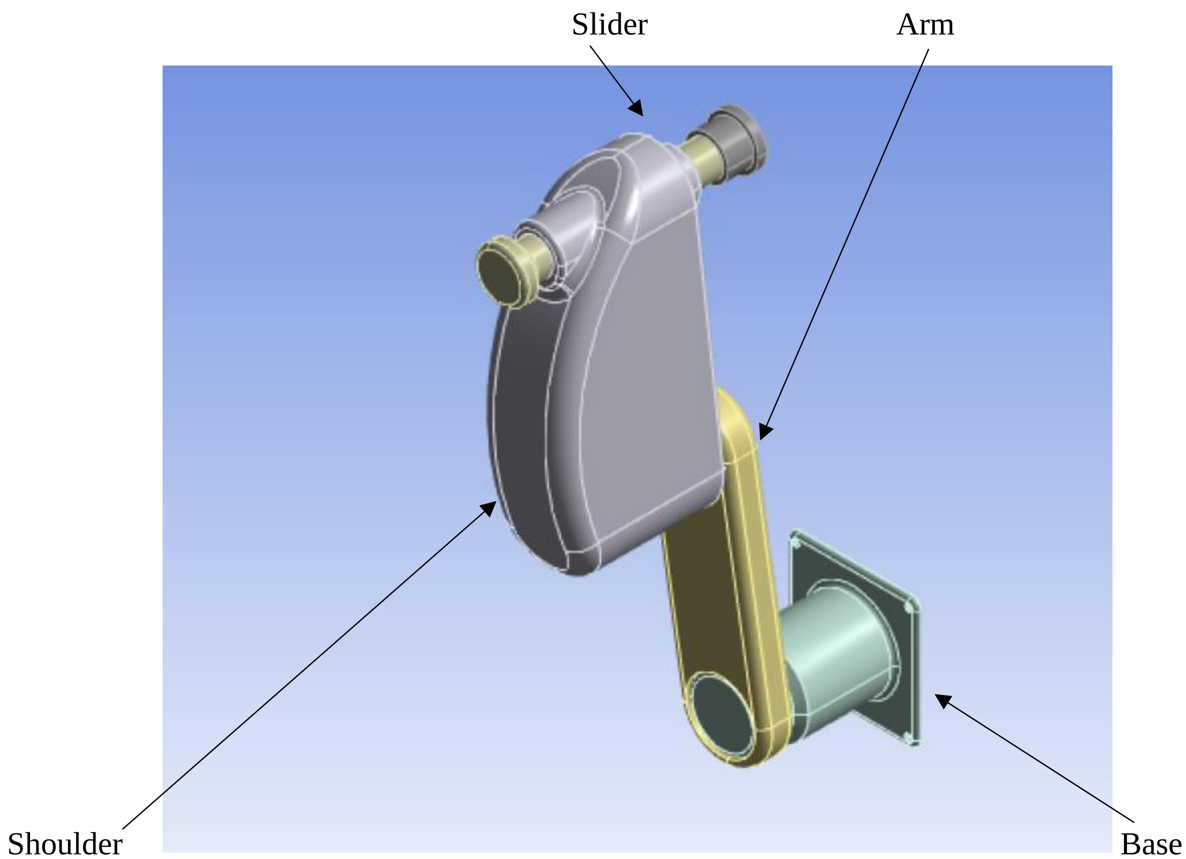


This **layout** will be displayed

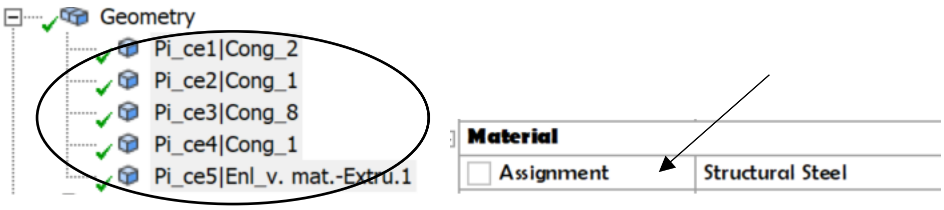


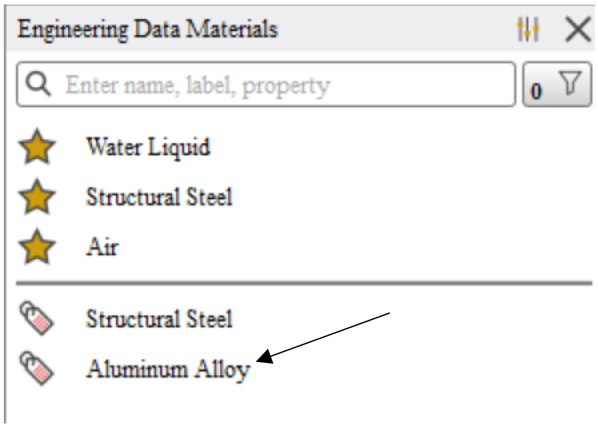
Now let us look into the Model Geometry there are four parts in it **BASE, ARM, SHOULDER, and SLIDER.**

There are **4DOF** in SCARA ROBOT.



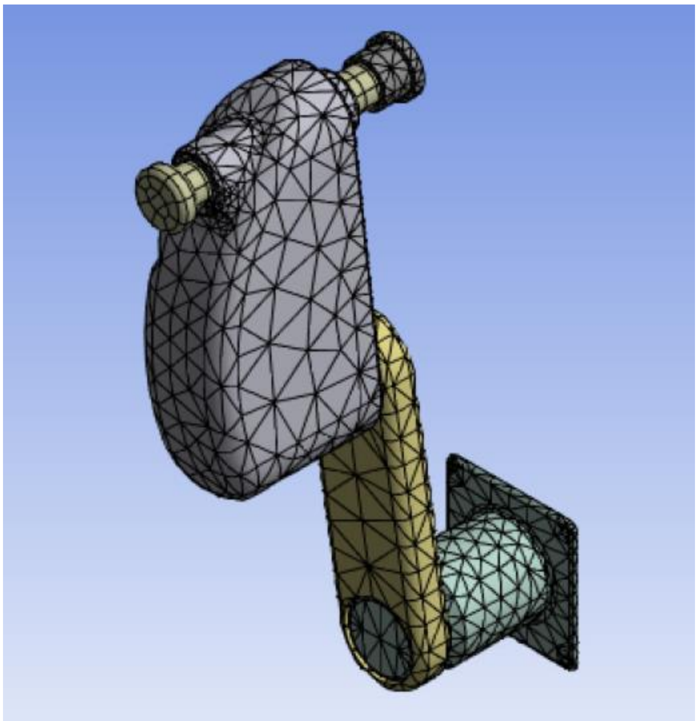
Now we must select the whole geometry and then tap on material Assignment to change the material from **structural steel to aluminium.**





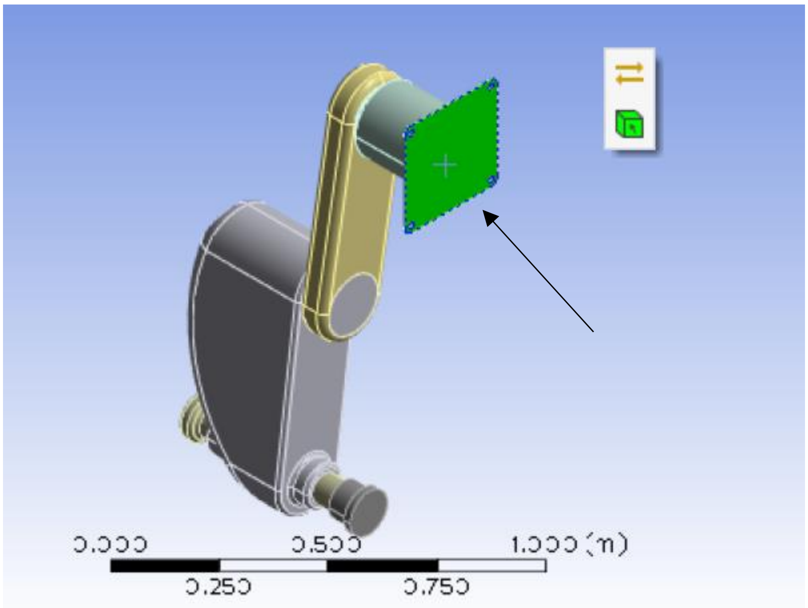
Now we have to select the meshing. It is important in the analysis part. To get the accurate solution we need to tap on **Find mesh** of the particular component instead of normal mesh.

Time taken by Find mesh is more when compared to normal mesh. At the **joint area, more elements** were generated when compared to other areas of robot.



So, after completing the meshing we need to give the boundary conditions in static structural which are **Fixed support boundary** conditions and the **load subjected** to the particular component.

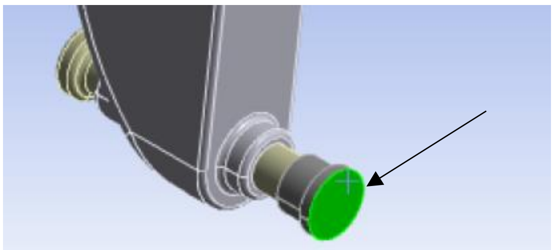
So, we need to right-click the **Static Structural** and we need to insert **fixed support** and rotate the base and select base to make it fixed support.



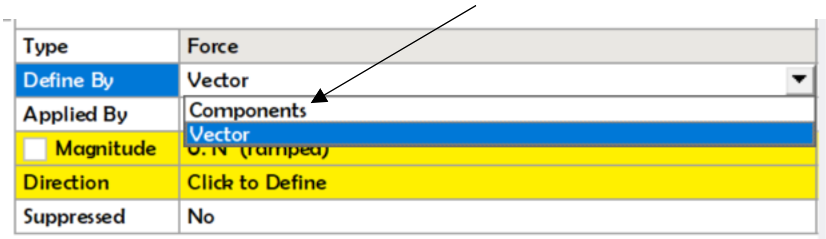
Now we must click on the **apply**

Scope	
Scoping Method	Geometry Selection
Geometry	Apply / Cancel
Definition	
Type	Fixed Support
Suppressed	No

Now we need to apply load on the end effector. So, we need to right-click the **Static Structural** and we need to insert **force** and rotate the end effector and select the end effector to make apply force. And we need to click on apply button.

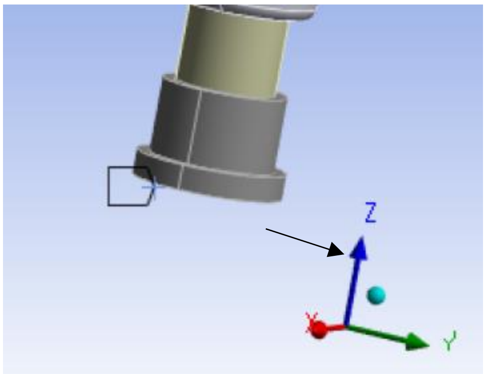


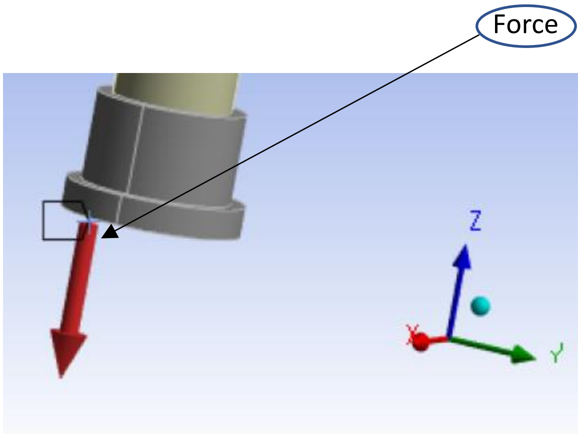
For an easy understanding of applying force let us click on the **components** instead of the vector.



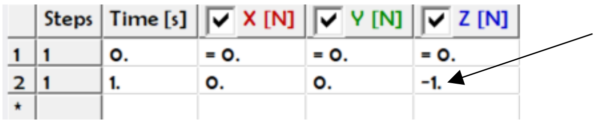
Type	Force
Define By	Vector
Applied By	Components
Magnitude	Vector
Direction	Click to Define
Suppressed	No

We can see that Z is the upward force direction opposite to the end effector, but we need to apply force vertically downward. So, we need to add a **negative force** to the Z direction.



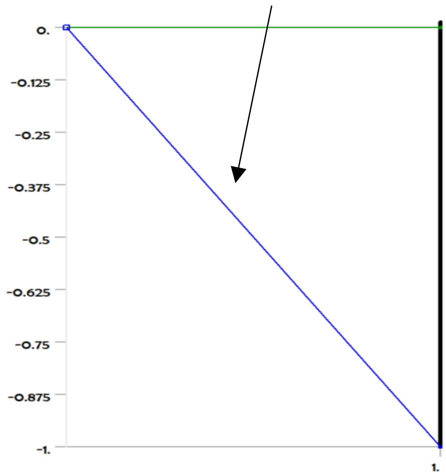


-1N(0.1Kg) force added to Z direction. Now let us check the deformation values.

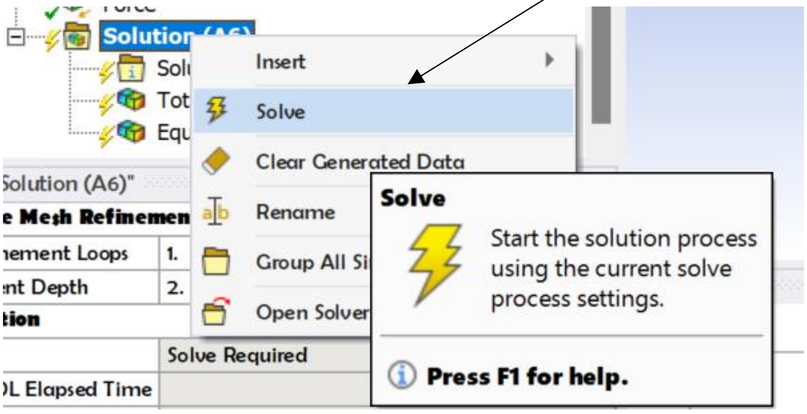


	Steps	Time [s]	X [N]	Y [N]	Z [N]
1	1	0.	= 0.	= 0.	= 0.
2	1	1.	0.	0.	-1.
*					

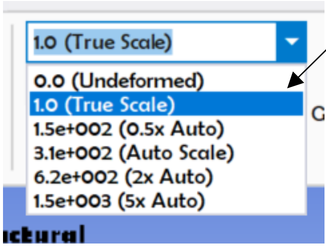
Graph (1N force in the negative Z direction linearly increases as time increases up to 1sec**)**



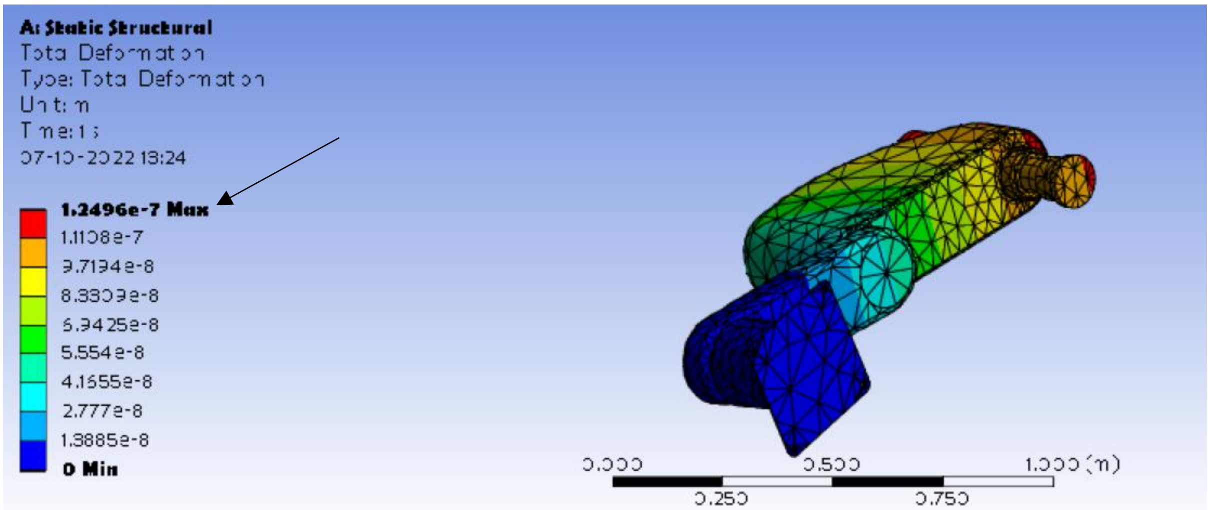
Now we must click on right click on **Solution** and then insert the **Total Deformation** and also insert the **Equivalent Stress.** And now right click the **Solution** and tap on **Solve** button**.**



Now select the **true scale** in the result section to find the actual deformation in the particular component.



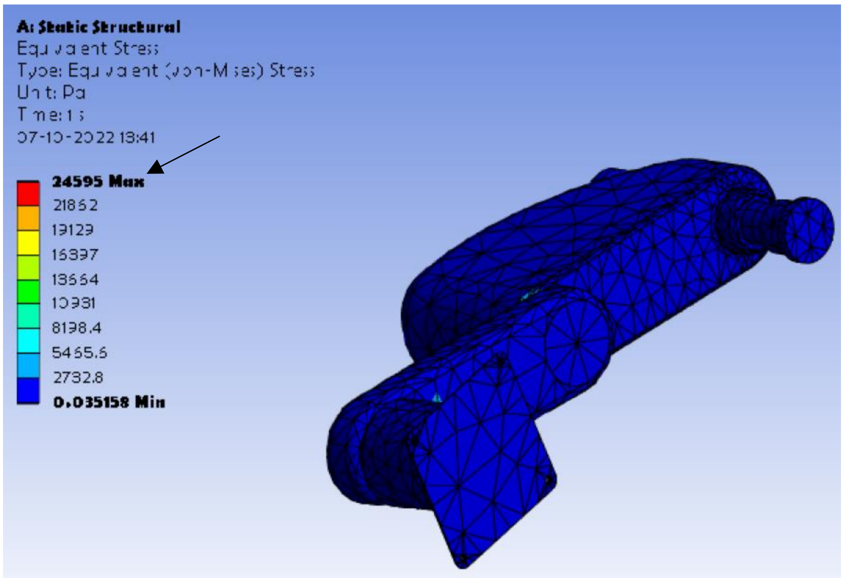
Now let us check the deformation by clicking on the **Total deformation** option. Now we can see the deformation values as follows.



Max deformation is 0.00012mm. It is negligible deformation. A small amount of deformation is considered as negligible deformation of the particular load. We are not considering the strength of the component we need to find the deformation present on the end effector. Complete deformation should be very less so that we can get the desired position and orientation of the end effector.

Now we need to look into **Equivalent Stress**, which has certain limits.

For aluminum Tensile strength is **75MPa to 135MPa.** Beyond this stress increases the material fails: the absorption of forces decreases until the material specimen ultimately tears. So, for the given 1N Load the **Max stress is 24595Pa** = 24kPa. So the End effector can withstand the load without any error in reaching the desired coordinates.



<u>Conclusion</u>:

Kinematic analysis of the 4DOF SCARA robot was simulated using Robo Analyzer software.

At the end of this project, we came to a clear understanding of Robo Analyzer Software.

By using Robo Analyzer we obtained forward kinematics of the end effector of the robot for the given joint parameters.

By using graphs in Robo analyzer software we analyzed the change in joint angles with respect to time and the change in joint velocities with respect to time.

By inputting the Material used in the component and the Load value we got desired Deformation Value and Equivalent Stress of the end effector we concluded that the component should have a low deformation value to maintain accuracy in reaching the goal coordinates and the Tensile strength of the aluminium is in between 75 to 135 MPa and the loading stress should not cross beyond these values to maintain stability and accuracy.

Thus, using roboanalyzer SCARA robot end effectors configuration was determined for different joint variables.

For the given SCARA robot, the payload is determined using ANSYS WORKBENCH.

--------------------------xxxxxxxxx------------------------