SONA COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Salem-5

Department of Mechatronics Engineering

U15MC703R – Robotics Laboratory

Academic year 2021-22

Name :

Reg. No :

Year :

Semester :

EXP.: 1

FORWARD KINEMATICS OF MOVEMASTER RM-501 – HORIZONTAL POSITION

Aim:

To create and simulate horizontal position of Movemaster RM-501 and finding the joint angular values.

Objective:

- To identify the geometric relationship between input and output motion parameters of Movemaster RM-501 robotic arm.
- Formation of the transformation matrix though which a relationship is established between different links of the manipulator.
- To have a brief idea about the workspace through a 3D graph plot of manipulator position for various inputs.
- Simulate the robot motion for various inputs of the joint angular value.
- Robot Trajectory Visualization while moving from one position to another.

Theory:

Movemaster RM-501:

- The Mitsubishi Movemaster RM-501 robot is a robotic arm having 5 degree of freedom with 5 rotational joints.
- The robot unit consists of the waist, shoulder, elbow, wrist pitch and wrist roll.
- These are actuated with DC motors to each of which two encoders are associated, which allow differential directional information to be read for each joint.

Overall Size:

- The robot unit, including the base weighs 27 kg.

- The overall size of the robot is comparatively small with a circular work envelope of 445 mm maximum horizontal reach and a maximum payload of 1.2 kg.

Components:

- Computer RS232
- Kinematic Controller
- Drive Unit
- Teaching Box
- Robot Unit
- Gripper

Range of Movement of each Joint:

Joint	Range (theta)
Waist Joint	-150 to +150
Shoulder Joint	-100 to +30
Elbow Joint	-45 to +45
Wrist Bend	0 to -180
Wrist Roll	-180 to +180

Procedure:

1. Insert different values of `theta` within the joint range as prescribed in theory part and then click ok to get the output orientation and position of the end effector.
2. To see the individual movements of the robot links, drag the sliders on the controller panel.
3. Manipulator position is shown in a 3D graph for every submission of joint values.
4. The view can be rotated about a point by keeping the *left mouse button* pressed and rotating the mouse.
5. The view can be translated by keeping the *right mouse button pressed* and translating the mouse in the desired direction.
6. The *scroll button* or *middle mouse button* can be used for zooming.

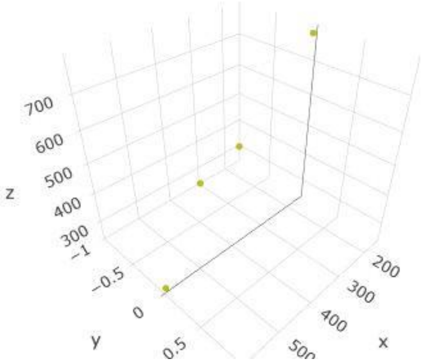


Fig. Final Plot

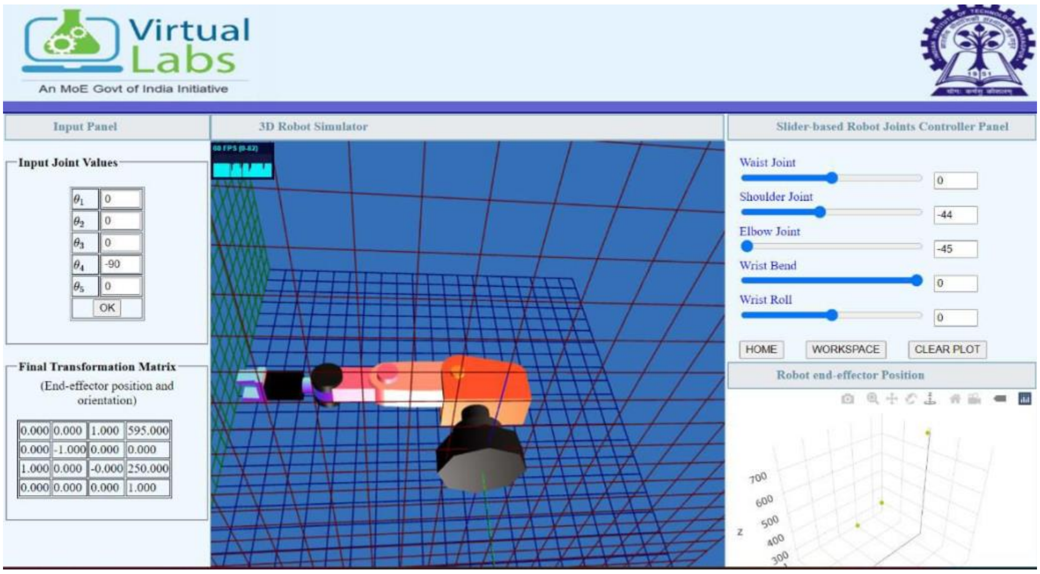


Fig. Horizontal Position of Movemaster RM-501

Result:

- The horizontal position is achieved by joint angle using Movemaster RM-501.
- The horizontal position is achieved by assigning the theta values of joints as follows:

Joint	Angle (theta)
Waist Joint	
Shoulder Joint	
Elbow Joint	
Wrist Bend	
Wrist Roll	

EXP.: 2

FORWARD KINEMATICS OF MOVEMASTER RM-501 – VERTICAL POSITION

Aim:

To create and simulate vertical position of Movemaster RM-501 and finding the joint angular values.

Objective:

- To identify the geometric relationship between input and output motion parameters of Movemaster RM-501 robotic arm.
- Formation of the transformation matrix though which a relationship is established between different links of the manipulator.
- To have a brief idea about the workspace through a 3D graph plot of manipulator position for various inputs.
- Simulate the robot motion for various inputs of the joint angular value.
- Robot Trajectory Visualization while moving from one position to another.

Theory:

Movemaster RM-501:

- The Mitsubishi Movemaster RM-501 robot is a robotic arm having 5 degree of freedom with 5 rotational joints.
- The robot unit consists of the waist, shoulder, elbow, wrist pitch and wrist roll.
- These are actuated with DC motors to each of which two encoders are associated, which allow differential directional information to be read for each joint.

Overall Size:

- The robot unit, including the base weighs 27 kg.

- The overall size of the robot is comparatively small with a circular work envelope of 445 mm maximum horizontal reach and a maximum payload of 1.2 kg.

Components:

- Computer RS232
- Kinematic Controller
- Drive Unit
- Teaching Box
- Robot Unit
- Gripper

Range of Movement of each Joint:

Joint	Range (theta)
Waist Joint	-150 to +150
Shoulder Joint	-100 to +30
Elbow Joint	-45 to +45
Wrist Bend	0 to -180
Wrist Roll	-180 to +180

Procedure:

- Insert different values of `theta` within the joint range as prescribed in theory part and then click ok to get the output orientation and position of the end effector.
- To see the individual movements of the robot links, drag the sliders on the controller panel.
- Manipulator position is shown in a 3D graph for every submission of joint values.
- The view can be rotated about a point by keeping the *left mouse button* pressed and rotating the mouse.
- The view can be translated by keeping the *right mouse button pressed* and translating the mouse in the desired direction.
- The *scroll button* or *middle mouse button* can be used for zooming.

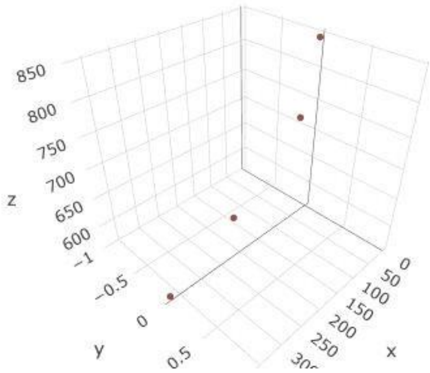


Fig. Final Plot

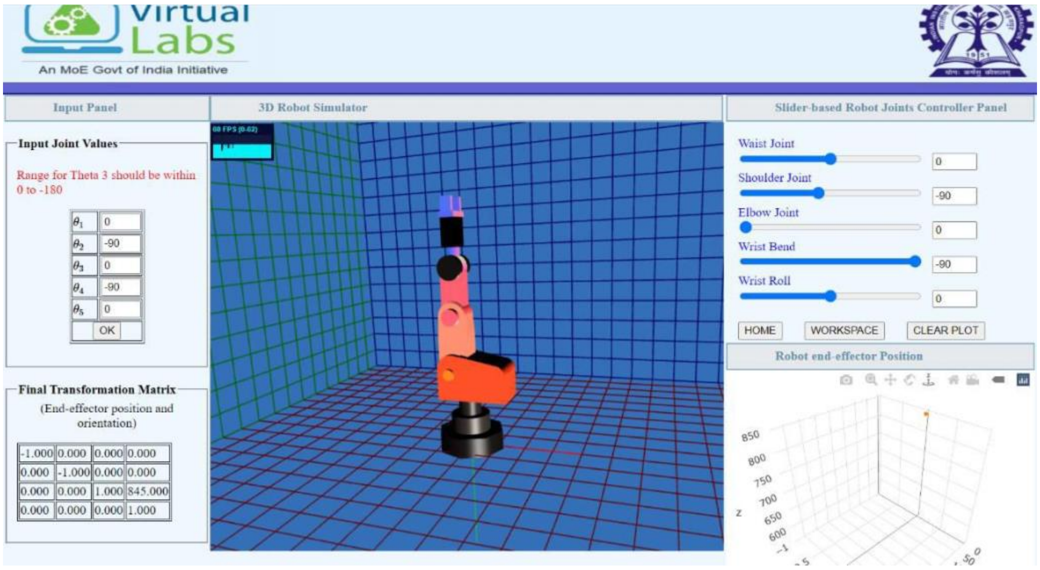


Fig. Vertical Position of Movemaster RM-501

Result:

- The vertical position is achieved by joint angle using Movemaster RM-501.
- The vertical position is achieved by assigning the theta values of joints as follows:

Joint	Angle (theta)
Waist Joint	
Shoulder Joint	
Elbow Joint	
Wrist Bend	
Wrist Roll	

EXP.: 3

FORWARD KINEMATICS OF PUMA 560 – HORIZONTAL POSITION

Aim:

To create and simulate horizontal position of PUMA 560.

Objective:

- To identify the geometric relationship between input and output motion parameters of PUMA 560 robotic manipulator.
- Formation of the transformation matrix though which a relationship is established between different links of the manipulator.
- To have a brief idea about the workspace through a 3D graph plot of manipulator position for various inputs.
- Simulate the robot motion for various inputs of the joint angular value.
- Robot Trajectory Visualization while moving from one position to another.

Theory:

PUMA 560:

- Programmable Universal Machine for Assembly, popularly known as PUMA is an industrial robot arm developed by Victor Scheinman at Unimation in 1978.
- PUMA comes in various makes PUMA 260, PUMA 560, PUMA 761, etc.
- Forward kinematics problem is concerned with the relationship between the individual joints of the robot manipulator, the position and orientation of the tool or end effector.

Components:

- Computer RS232
- Kinematic Controller
- Drive Unit

- Teaching Box
- Robot Unit
- Gripper

Range of Movement of each Joint:

Joint	Range (theta)
Waist Joint	-160 to +160
Shoulder Joint	-225 to +45
Elbow Joint	-225 to +45
Wrist Roll	-110 to +170
Wrist Bend	-100 to +100
Wrist Swivel	-266 to +266

Procedure:

- Insert different values of `theta` within the joint range as prescribed in theory part and then click ok to get the output orientation and position of the end effector.
- To see the individual movements of the robot links, drag the sliders on the controller panel.
- The Transformation Matrix for a particular position and orientation can be obtained either through input panel or via the controller.
- Manipulator position is shown in a 3D graph for every submission of joint values.
- The view can be rotated about a point by keeping the *left mouse button* pressed and rotating the mouse.
- The view can be translated by keeping the *right mouse button pressed* and translating the mouse in the desired direction.
- The *scroll button* or *middle mouse button* can be used for zooming.

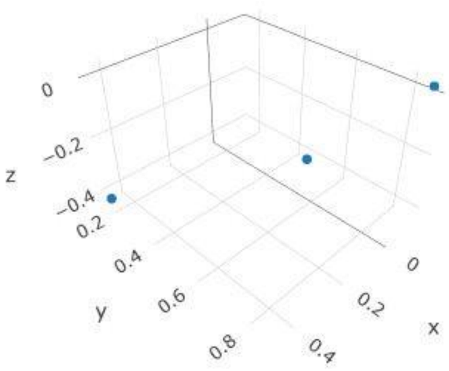


Fig. Final Plot

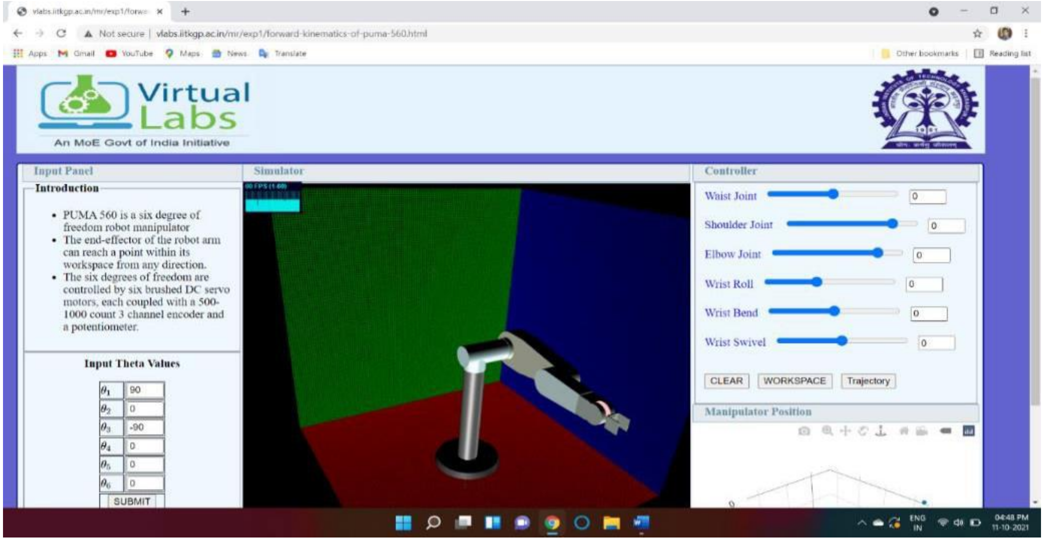


Fig. Horizontal Position of PUMA 560

Result:

- The horizontal position is achieved by joint angle using PUMA 560.
- The horizontal position is achieved by assigning the theta values of joints as follows:

Joint	Angle (theta)
Waist Joint	
Shoulder Joint	
Elbow Joint	
Wrist Roll	
Wrist Bend	
Wrist Swivel	

EXP.: 4

FORWARD KINEMATICS OF PUMA 560 – VERTICAL POSITION

Aim:

To create and simulate vertical position of PUMA 560

Objective:

- To identify the geometric relationship between input and output motion parameters of PUMA 560 robotic manipulator.
- Formation of the transformation matrix though which a relationship is established between different links of the manipulator.
- To have a brief idea about the workspace through a 3D graph plot of manipulator position for various inputs.
- Simulate the robot motion for various inputs of the joint angular value.
- Robot Trajectory Visualization while moving from one position to another.

Theory:

PUMA 560:

- Programmable Universal Machine for Assembly, popularly known as PUMA is an industrial robot arm developed by Victor Scheinman at Unimation in 1978.
- PUMA comes in various makes PUMA 260, PUMA 560, PUMA 761, etc.
- Forward kinematics problem is concerned with the relationship between the individual joints of the robot manipulator, the position and orientation of the tool or end effector.

Components:

- Computer RS232
- Kinematic Controller
- Drive Unit

- Teaching Box
- Robot Unit
- Gripper

Range of Movement of each Joint:

Joint	Range (theta)
Waist Joint	-160 to +160
Shoulder Joint	-225 to +45
Elbow Joint	-225 to +45
Wrist Roll	-110 to +170
Wrist Bend	-100 to +100
Wrist Swivel	-266 to +266

Procedure:

- Insert different values of `theta` within the joint range as prescribed in theory part and then click ok to get the output orientation and position of the end effector.
- To see the individual movements of the robot links, drag the sliders on the controller panel.
- The Transformation Matrix for a particular position and orientation can be obtained either through input panel or via the controller.
- Manipulator position is shown in a 3D graph for every submission of joint values.
- The view can be rotated about a point by keeping the *left mouse button* pressed and rotating the mouse.
- The view can be translated by keeping the *right mouse button pressed* and translating the mouse in the desired direction.
- The *scroll button* or *middle mouse button* can be used for zooming.

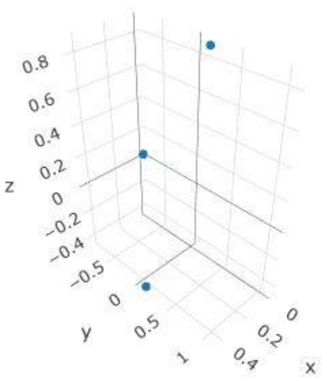


Fig. Final Plot

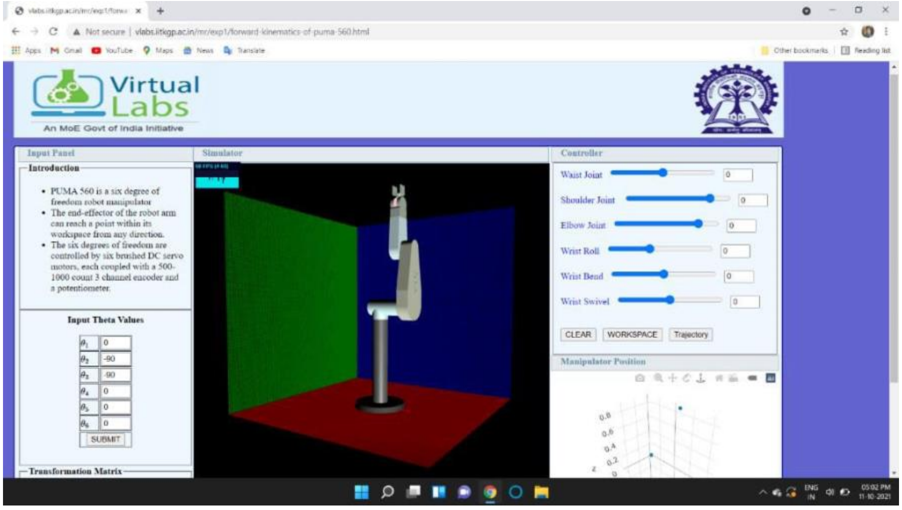


Fig. Vertical Position of PUMA 560

Result:

- The vertical position is achieved by joint angle using PUMA 560.
- The vertical position is achieved by assigning the theta values of joints as follows:

Joint	Angle (theta)
Waist Joint	
Shoulder Joint	
Elbow Joint	
Wrist Roll	
Wrist Bend	
Wrist Swivel	

EXP.: 5

Maximum Horizontal Reach of KGP 50

Aim:

To create maximum horizontal reach of KGP 50 Using Vlabs.

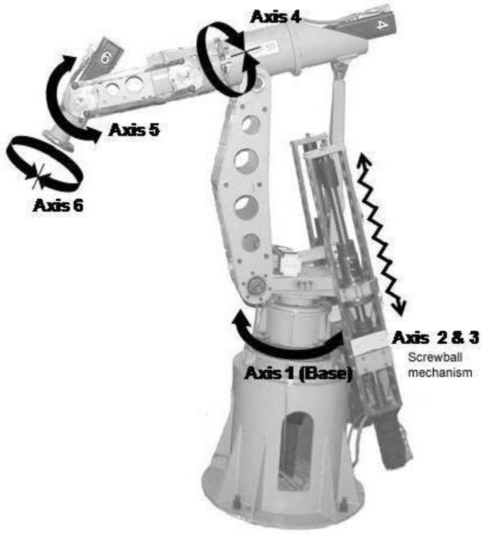
Objective:

- To identify the geometric relationship between input and output motion parameters of KGP 50 robotic manipulator.
- Formation of the transformation matrix though which a relationship is established between different links of the manipulator.
- To have a brief idea about the workspace through a 3D graph plot of manipulator position for various inputs.
- Simulate the robot motion for various inputs of the joint angular value.
- Robot Trajectory Visualization while moving from one position to another.

Theory:

KGP 50 :

- The "KGP 50" is a state-of-the-art industrial prototype robot, positioned in the latest generation of industrial robot category.
- Currently, it is being used as a test bed for cutting edge technologies of human computer interaction and intelligent systems. Robotic technology, harnessed in the laboratory, has spawned and supported many growing areas of intelligent systems research. Prominent among these were the development of an intelligent driver vehicle interaction simulator. The system is a vehicle-driving simulator for a passenger car, where the driver's responses and automated vehicle control strategies under different driving conditions are generated and designs tested. A novel techniques of using fast learning neural networks for crisis management and control is being evolved on this test bed. It is expected to evolve a man machine collaborative learning system for intelligent systems in the new generation auto-mated vehicular systems.



Specifications:

- 6-Axis, Continuous path control
- 50kg Payload
- 1.5m reach
- 1.5m/s maximum speed
- 1mm repeatability

Characteristics:

Full 6-Axis, Continuous servo implementation :

This robot is designed for industrial specification of all 6 axis motion specification with a considerable workspace for manipulation and dexterity. The axis are in continuous servo mode and fine positioning and velocity control for continuous paths can thus be easily achieved in industrial applications with this model. This feature is comparable to top of the line models in all international robot models.

3-Phase AC digital servo technology implemented and tested:

The robot is powered by the state-of-the art digital servo control of AC motors. The drive technology is the muscle of the robot, which drivers high torque even at a low speed maneuver operations, as well as rapid acceleration and decelerations required in high production rate manufacturing operations. The choice of technology has also been influenced by the international trend by which servo drive technology is moving towards model-based control and direct torque control. This technology is comparable and equals that of the top of the line industrial robot models used for a variety of high production rate processes and heavy-duty assembly line operations. The robot has its six degrees of freedom and motors synchronized and precision-controlled for coordinated motions using a real-time digital control station working on a Digital Signal Processing-based motion controller. The real-time operating system in the robot enables it to interact well with its environment

DSP-based motion control implemented:

The skills of the robot are derived from the state-of-the-art motion control technology implemented in each of drivers and co-ordinate systems of the robot. A highly complex computational scheme in providing spline and continuous motion and trajectory interpolation in industry-desired co-ordinate frames has been adopted in the core design of the robot controller. This provides the computational capacity to operate the robot in the continuous path mode with extremely high repeatability and low error in all the six joint axis as well as cartesian or factory co-ordinate systems. It also provides the core for the extremely fast response of the robot to its own sensor systems as well as to interlock with other systems in any real application.

Model for online and offline programming based on industry standard robotics software developed:

Robot of such complex order (six axis in continuous path mode) are extremely difficult to program by operators on the shop floor without a considerable technological training. In order to reduce the complexity of the training and operation phase in use, as well as make the system more usable by intuitive operators, the design incorporates a novel 3-D geometrical programming environment. This is based on industry standard CAD systems that provide an interface for easy operability and verification of the robot program. this unique feature can help in drastically reducing the operational costs related to operate training, downtime due to programming errors, rapid reprogramming of robot and testing of synchronization with any other robots or devices in real shop floor.

Capability to fine-tune and implement motion control strategies incorporated in the design:

The robot has an open architecture heart built-in into the system, that enables robot developers to diagnose and improvise the design and/or performance of the robot, even after deployment. This is an important feature for technology developers that is not available to consumers of robot technology as technology adopters. The robot system has been specifically designed with this features, which is no available from any competitors (as it's the core technology of the system) and is very much required for a development industry as well as systems which can be adapted to any high-end applications.

Procedure:

- Insert different values of θ within the joint range as prescribed in theory part and then click ok to get the output orientation and position of the end effector.
- To see the individual movements of the links drag the sliders on the controller panel.
- The view can be rotated about a point by keeping the left mouse button pressed and rotating the mouse.
- The view can be translated by keeping the right mouse button pressed and translating the mouse in the desired direction.
- The scroll button or middle mouse button can be used for zooming.

Range of Movement of each Joint:

Theta	**Range**
Theta 1	-160 to 160
Theta 2	-40 to 15
Theta 3	-15 to 40
Theta 4	-180 to 180
Theta 5	-100 to 100
Theta 6	-226 to 226

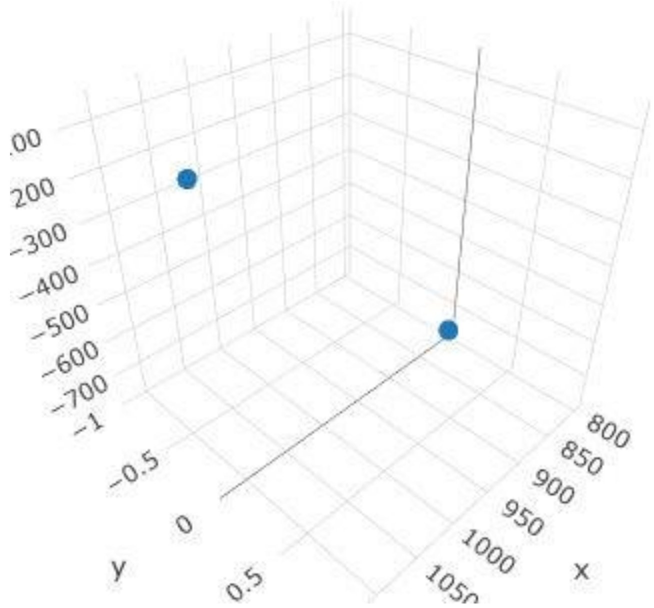


Fig. Final Plot

Fig. Horizontal Position of KGP 50

Result:

- The maximum reach of the horizontal position.
- The horizontal position is achieved by assigning the theta values of joints as follows:

Maximum Horizontal Reach

Theta	Range
Theta 1	
Theta 2	
Theta 3	
Theta 4	
Theta 5	
Theta 6	

EXP.: 6

Maximum Vertical Reach of KGP50

Aim:

To create maximum vertical reach of KGP50 Using Vlabs

Objective:

- To identify the geometric relationship between input and output motion parameters of KGP 50 robotic manipulator.
- Formation of the transformation matrix though which a relationship is established between different links of the manipulator.
- To have a brief idea about the workspace through a 3D graph plot of manipulator position for various inputs.
- Simulate the robot motion for various inputs of the joint angular value.
- Robot Trajectory Visualization while moving from one position to another.

Theory:

KGP 50 :

- The "KGP 50" is a state-of-the-art industrial prototype robot,positioned in the latest generation of industrial robot category.

Specifications:

- 6-Axis, Continuous path control
- 50kg Payload
- 1.5m reach
- 1.5m/s maximum speed
- 1mm repeatability

Characteristics:

Full 6-Axis, Continuous servo implementation :

This robot is designed for industrial specification of all 6 axis motion specification with a considerable workspace for manipulation and dexterity. The axis are in continuous servo mode and fine positioning and velocity control for continuous paths can thus be easily achieved in industrial applications with this model. This feature is comparable to top of the line models in all international robot models.

3-Phase AC digital servo technology implemented and tested:

The robot is powered by the state-of-the art digital servo control of AC motors. The drive technology is the muscle of the robot, which drivers high torque even at a low speed maneuver operations, as well as rapid acceleration and decelerations required in high production rate manufacturing operations.

DSP-based motion control implemented:

The skills of the robot are derived from the state-of-the-art motion control technology implemented in each of drivers and co-ordinate systems of the robot. A highly complex computational scheme in providing spline and continuous motion and trajectory interpolation in industry-desired co-ordinate frames has been adopted in the core design of the robot controller.

Model for online and offline programming based on industry standard robotics software developed:

Robot of such complex order (six axis in continuous path mode) are extremely difficult to program by operators on the shop floor without a considerable technological training. In order to reduce the complexity of the training and operation phase in use, as well as make the system more usable by intuitive operators, the design incorporates a novel 3-D geometrical programming environment.

Capability to fine-tune and implement motion control strategies incorporated in the design:

The robot has an open architecture heart built-in into the system, that enables robot developers to diagnose and improvise the design and/or performance of the robot, even after deployment. This is an important feature for technology developers that is not available to consumers of robot technology as technology adopters.

Procedure:

- Insert different values of θ within the joint range as prescribed in theory part and then click ok to get the output orientation and position of the end effector.
- To see the individual movements of the links drag the sliders on the controller panel.
- The view can be rotated about a point by keeping the left mouse button pressed and rotating the mouse.
- The view can be translated by keeping the right mouse button pressed and translating the mouse in the desired direction.
- The scroll button or middle mouse button can be used for zooming.

Range of Movement of each Joint:

Theta	Range
Theta 1	-160 to 160
Theta 2	-40 to 15
Theta 3	-15 to 40
Theta 4	-180 to 180
Theta 5	-100 to 100
Theta 6	-226 to 226

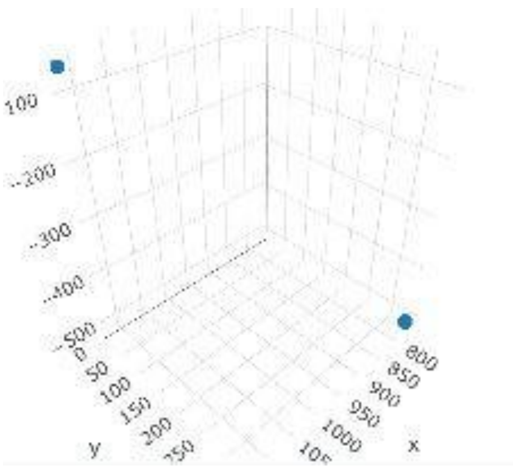


Fig. Final Plot

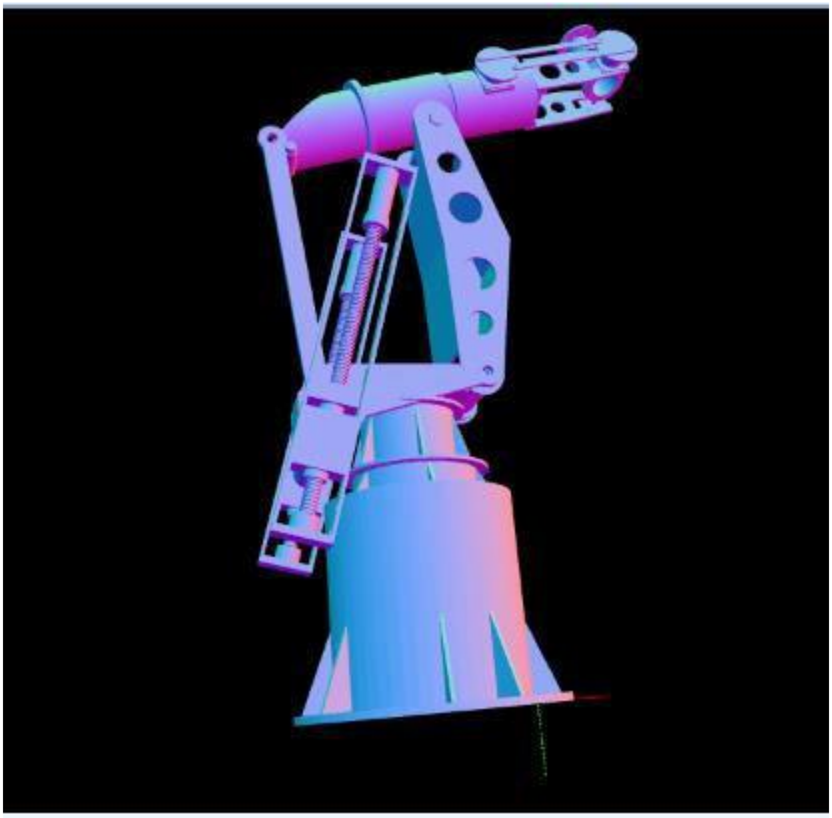


Fig. Vertical Position of KGP 50

Result:

- The maximum reach of the horizontal position.
- The horizontal position is achieved by assigning the theta values of joints as follows:

Maximum Horizontal Reach

Theta	Range
Theta 1	
Theta 2	
Theta 3	
Theta 4	
Theta 5	
Theta 6	

EXP: 7

MOVEMASTER RM-501 – PICK AND PLACE.

Aim:

To create and simulate pick and place an object of Movemaster RM-501.

Objective:

- To identify the geometric relationship between input and output motion parameters of Movemaster RM-501 robotic arm.
- Formation of the transformation matrix though which a relationship is established between different links of the manipulator.
- To have a brief idea about the workspace through a 3D graph plot of manipulator position for various inputs.
- Simulate the robot motion for various inputs of the joint angular value.
- Robot Trajectory Visualization while moving from one position to another.

Theory:

Movemaster RM-501:

- The Mitsubishi Movemaster RM-501 robot is a robotic arm having 5 degrees of freedom with 5 rotational joints.
- The robot unit consists of the waist, shoulder, elbow, wrist pitch and wrist roll.
- These are actuated with DC motors to each of which two encoders are associated, which allow differential directional information to be read for each joint.

Overall Size:

- The robot unit, including the base weighs 27 kg.
- The overall size of the robot is comparatively small with a circular work envelope of 445 mm maximum horizontal reach and a maximum payload of 1.2 kg.

Components:

- Computer RS232
- Kinematic Controller
- Drive Unit
- Teaching Box
- Robot Unit
- Gripper

Range of Movement of each Joint:

Joint	Range (theta)
Waist Joint	-150 to +150
Shoulder Joint	-100 to +30
Elbow Joint	-45 to +45
Wrist Bend	0 to -180
Wrist Roll	-180 to +180

Procedure:

1. Insert different values of `theta` within the joint range as prescribed in theory part and then click ok to get the output orientation and position of the end effector.
2. To see the individual movements of the robot links, drag the sliders on the controller panel.
3. Manipulator position is shown in a 3D graph for every submission of joint values.
4. The view can be rotated about a point by keeping the left mouse button pressed and rotating the mouse.
5. The view can be translated by keeping the right mouse button pressed and translating the mouse in the desired direction.
6. The scroll button or middle mouse button can be used for zooming.

Plot:

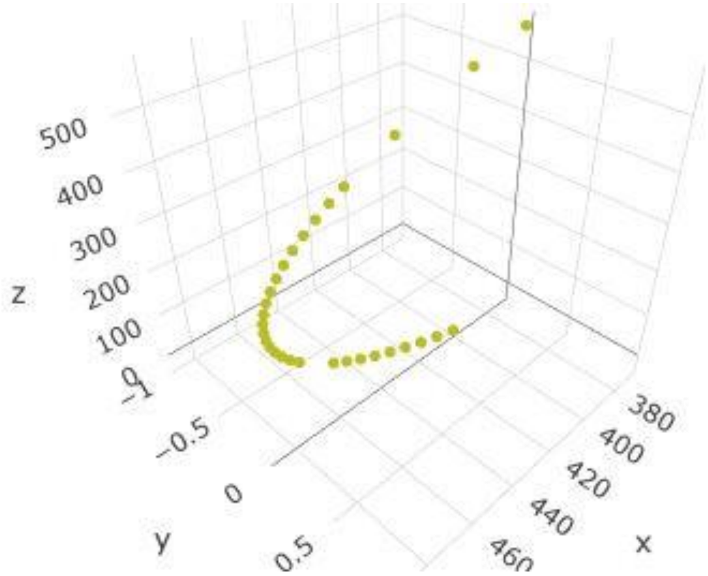


Fig 1. Initial Plot

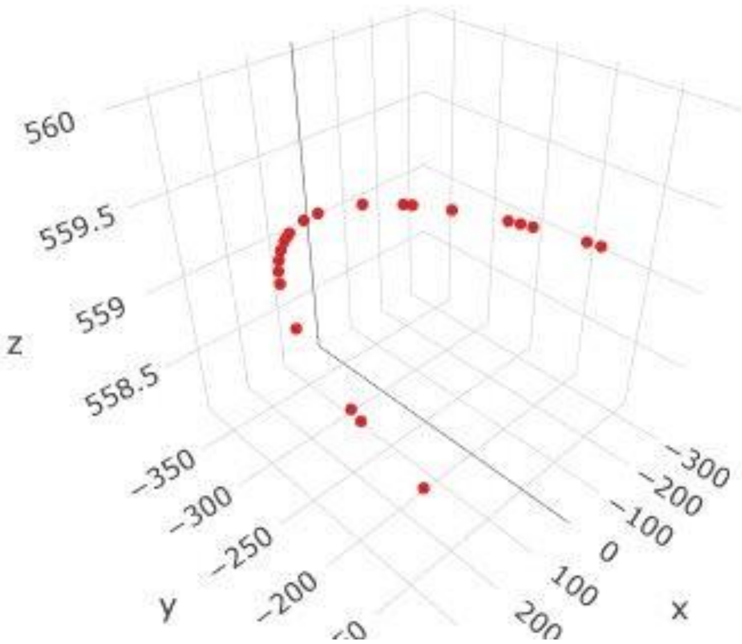


Fig2. Holding plot

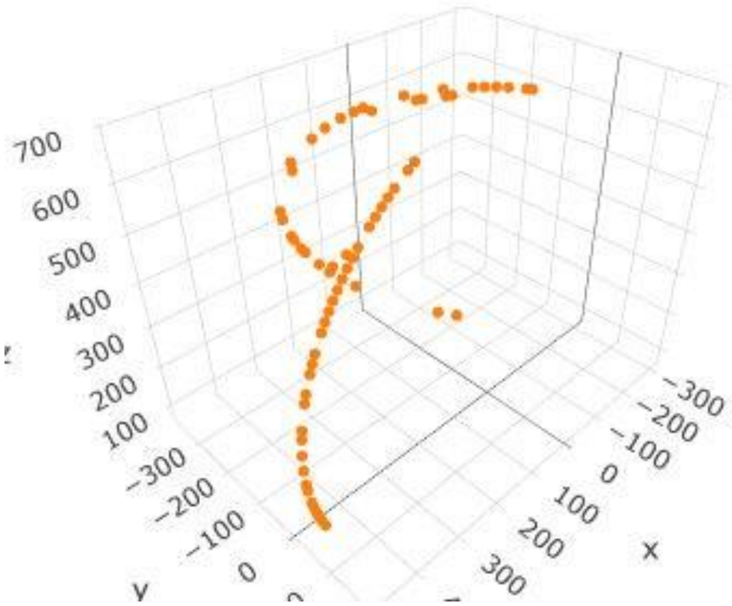


Fig3. Hold an object Plot

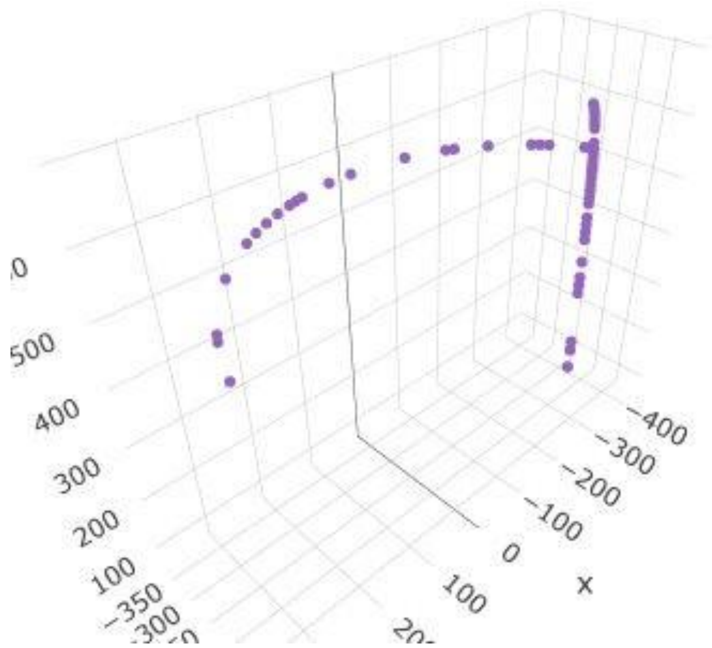


Fig4. Final Plot

Simulation:

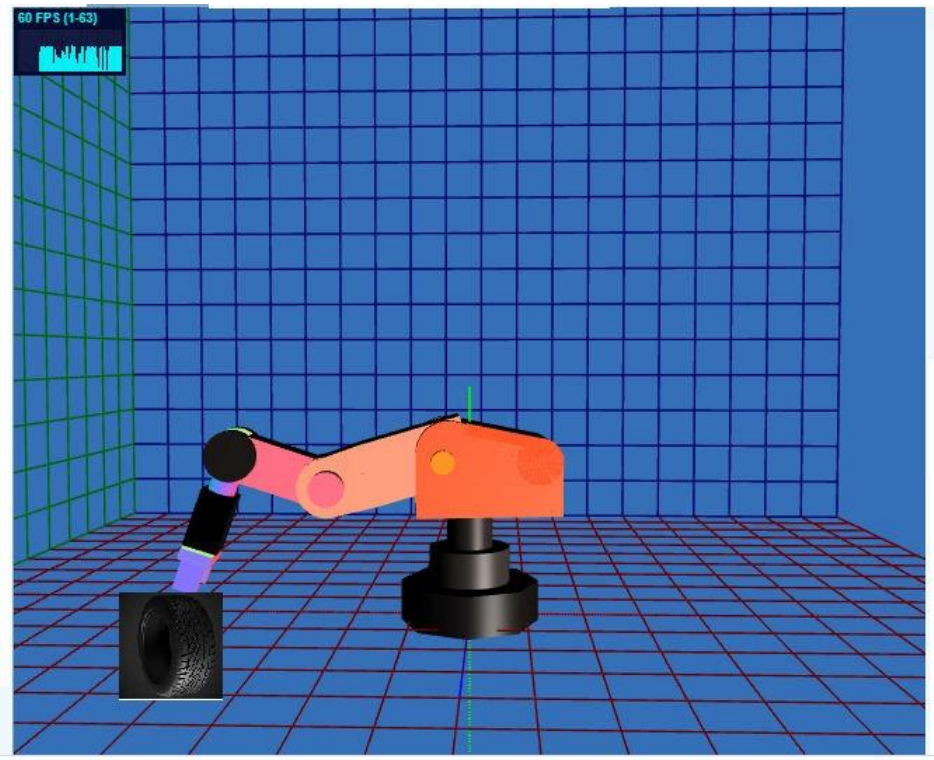


Fig5. Pick Position of Movemaster RM-501

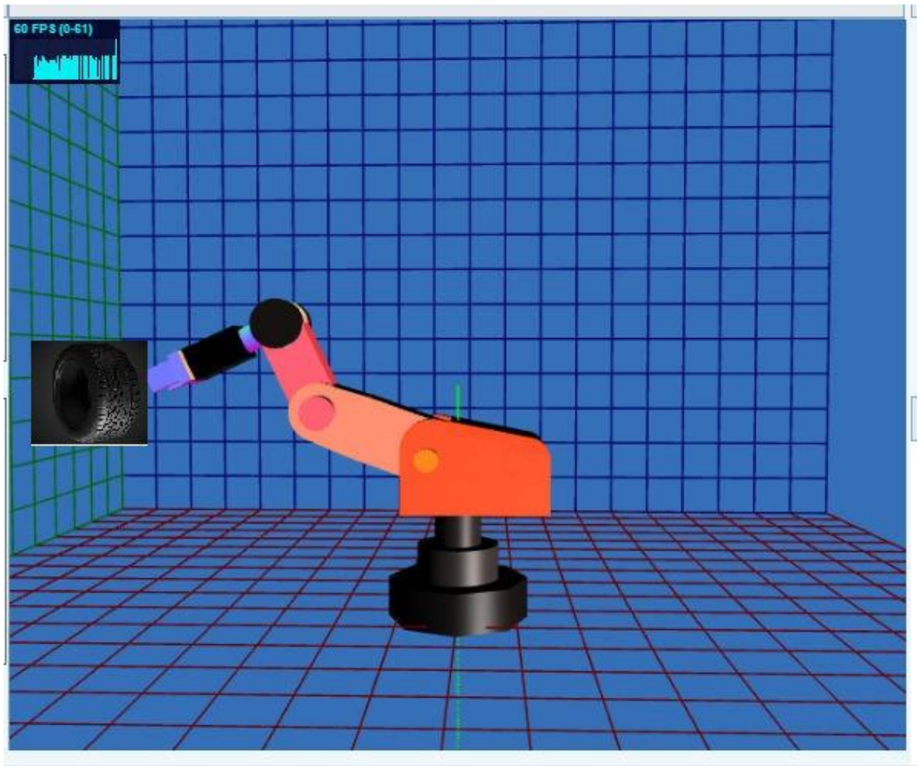


Fig6. Hold Position of Movemaster RM-501

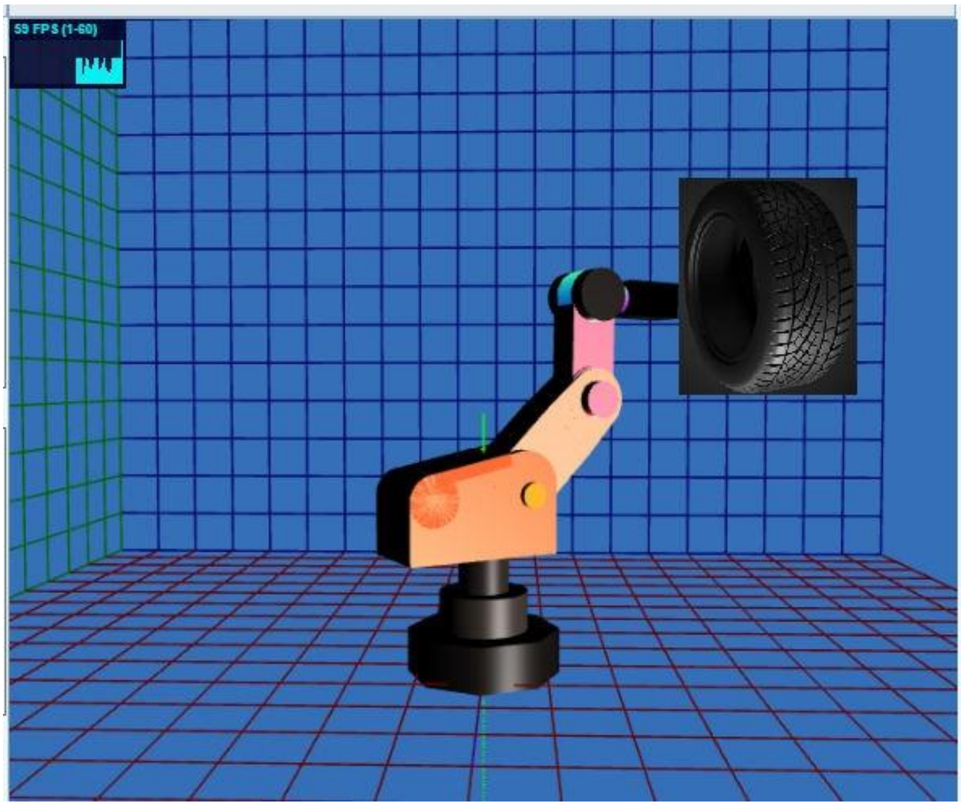


Fig7. Hold an object Simulation

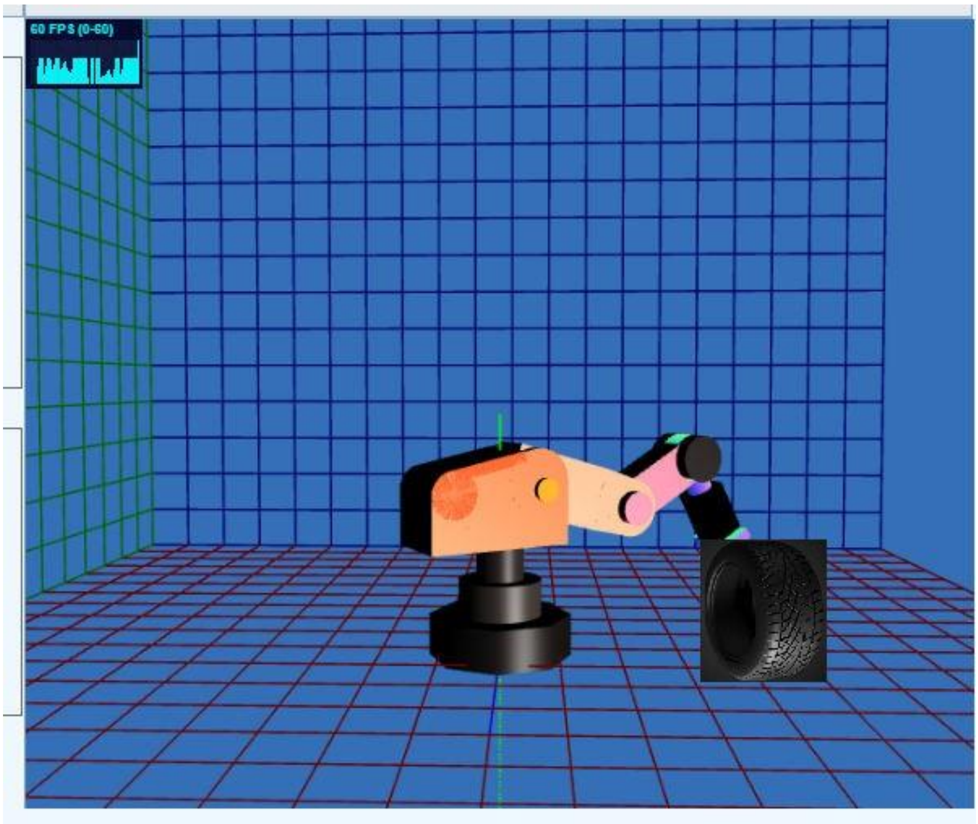


Fig8. Place Position of Movemaster RM-501

Result:

- The Pick and place are achieved by joint angle using Movemaster RM-501.
- The horizontal position is achieved by assigning the theta values of joints as follows:

Joint	**Angle (theta) Initial**	**Angle (theta) Pick**	**Angle (theta) Hold**	**Angle (theta) Place**
Waist Joint				
Shoulder Joint				
Elbow Joint				
Wrist Bend				
Wrist Roll				

EXP:8

PICK AND PLACE PUMA 560.

Aim:

To create and simulate PICK AND PLACE PUMA 560.

Objective:

- To identify the geometric relationship between input and output motion parameters of PUMA 560 robot manipulator.
- Formation of the transformation matrix though which a relationship is established between different links of the manipulator.
- Simulate the robot motion for various inputs of the joint angular value.
- To have a brief idea about the workspace through a 3D graph plot of manipulator position for various inputs.

Theory:

PUMA 560:

- The PUMA 560 robot is a robotic arm having 5 degrees of freedom with 5 rotational joints.
- The robot unit consists of the waist, shoulder, elbow, wrist pitch and wrist roll.
- These are actuated with DC motors to each of which two encoders are associated, which allow differential directional information to be read for each joint.

Overall Size:

- The robot unit, including the base weighs 27 kg.
- The overall size of the robot is comparatively small with a circular work envelope of 445 mm maximum horizontal reach and a maximum payload of 1.2 kg.

Components:

- Computer RS232
- Kinematic Controller
- Drive Unit
- Teaching Box
- Robot Unit
- Gripper

Range of Movement of each Joint:

Joint	Range (theta)
Waist Joint	-160 to +160
Shoulder Joint	-225 to +45
Elbow Joint	-225 to +45
Wrist Bend	-110 to -170
Wrist Roll	-100 to +100
Wrist Swivel	-266 to 266

Procedure:

1. Insert different values of `theta` within the joint range as prescribed in theory part and then click ok to get the output orientation and position of the end effector.
2. To see the individual movements of the robot links, drag the sliders on the controller panel.
3. Manipulator position is shown in a 3D graph for every submission of joint values.
4. The view can be rotated about a point by keeping the left mouse button pressed and rotating the mouse.
5. The view can be translated by keeping the right mouse button pressed and translating the mouse in the desired direction.
6. The scroll button or middle mouse button can be used for zooming.

Plot:

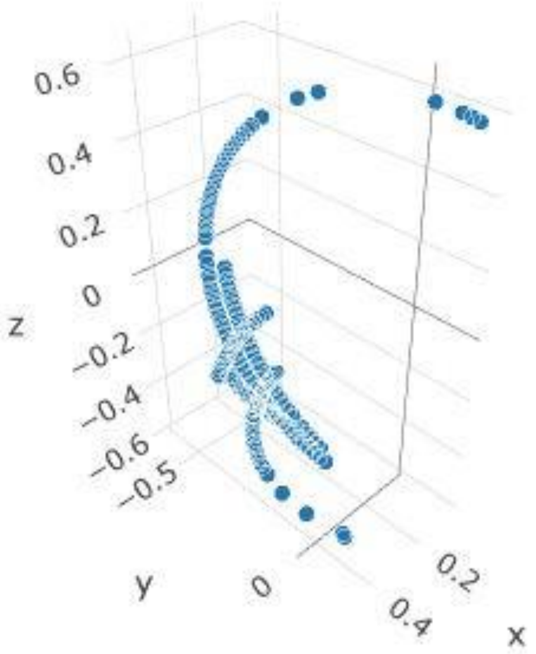


Fig1. Pick Plot

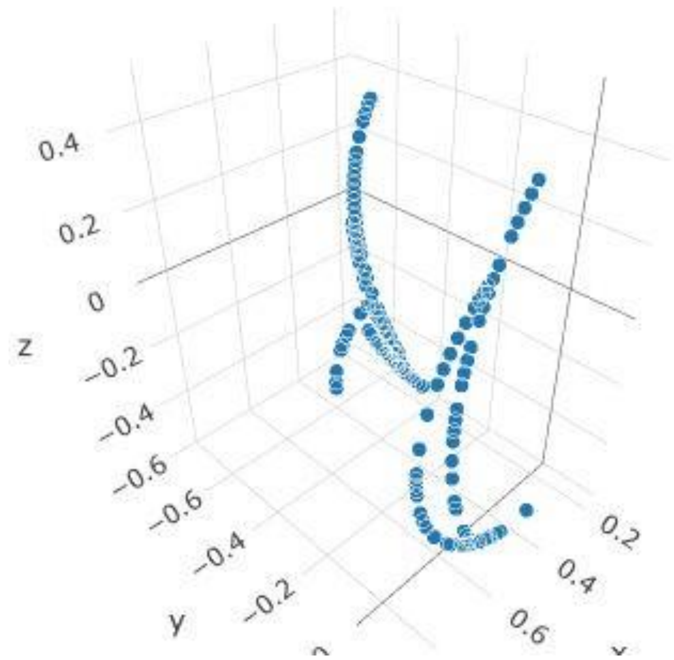


Fig2. Hold Plot

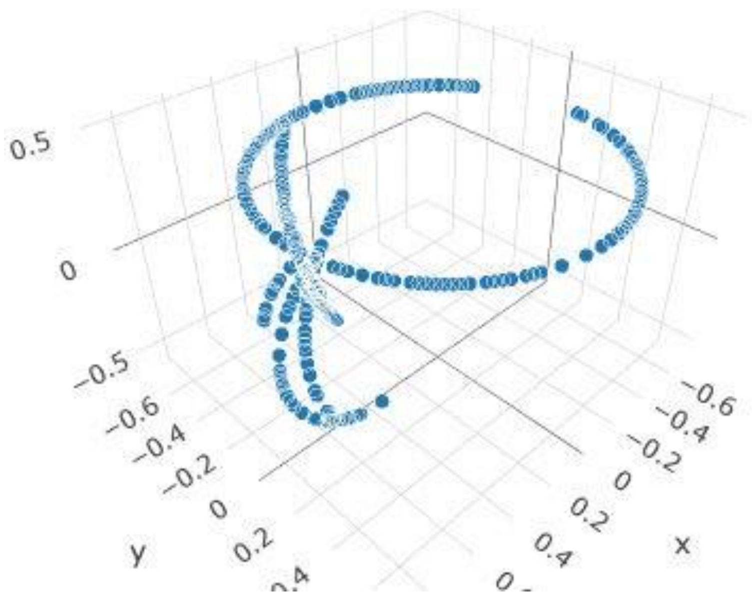


Fig3. Place Plot

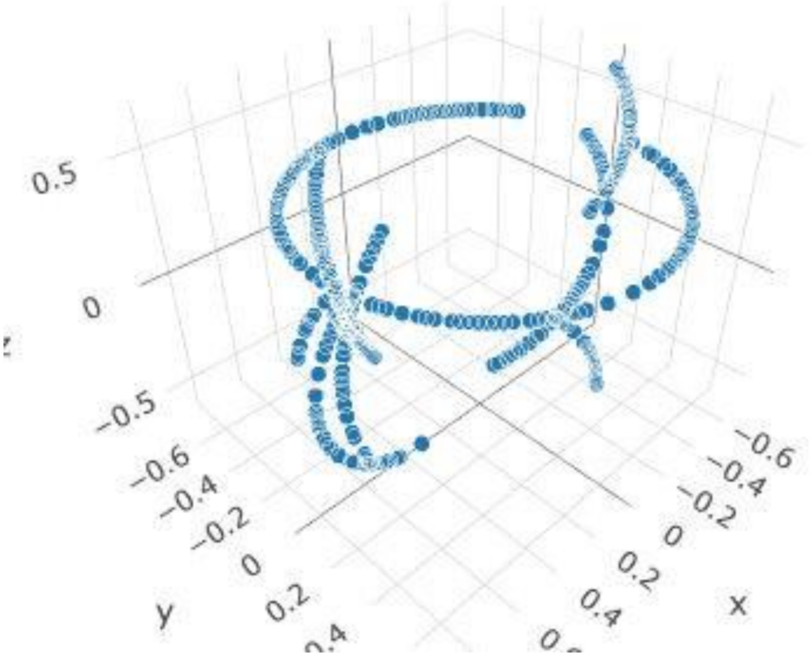


Fig4. Final Plot

Simulation:

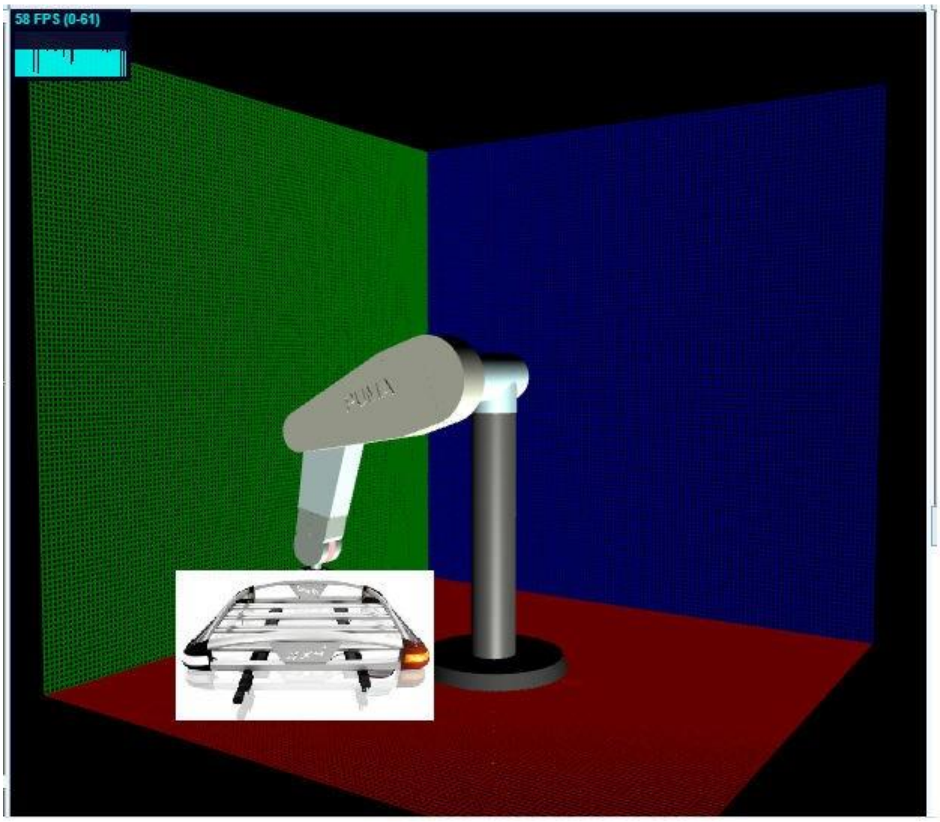


Fig5. Pick an object

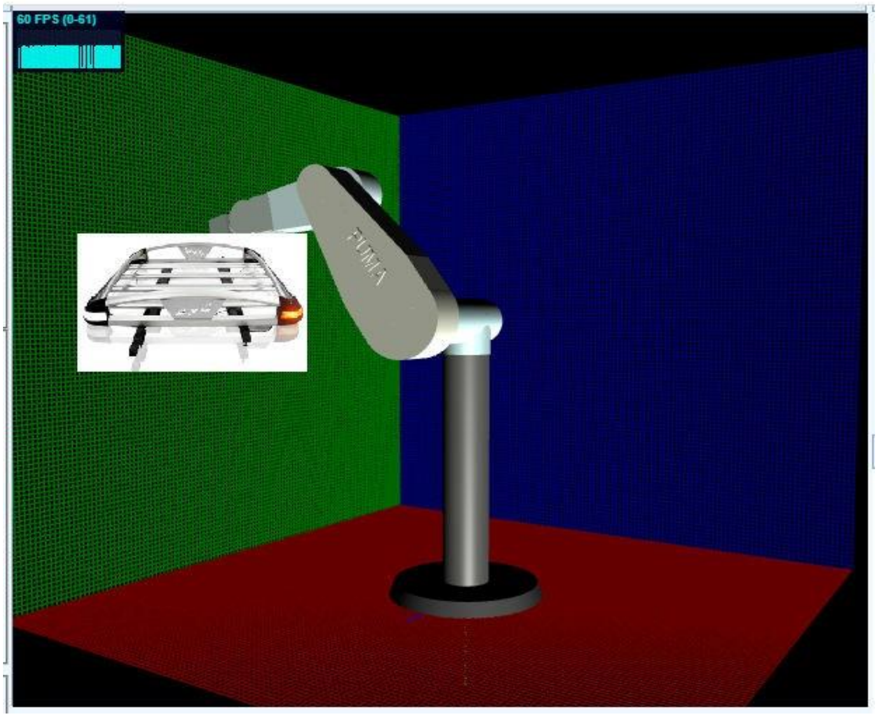


Fig6. move an object

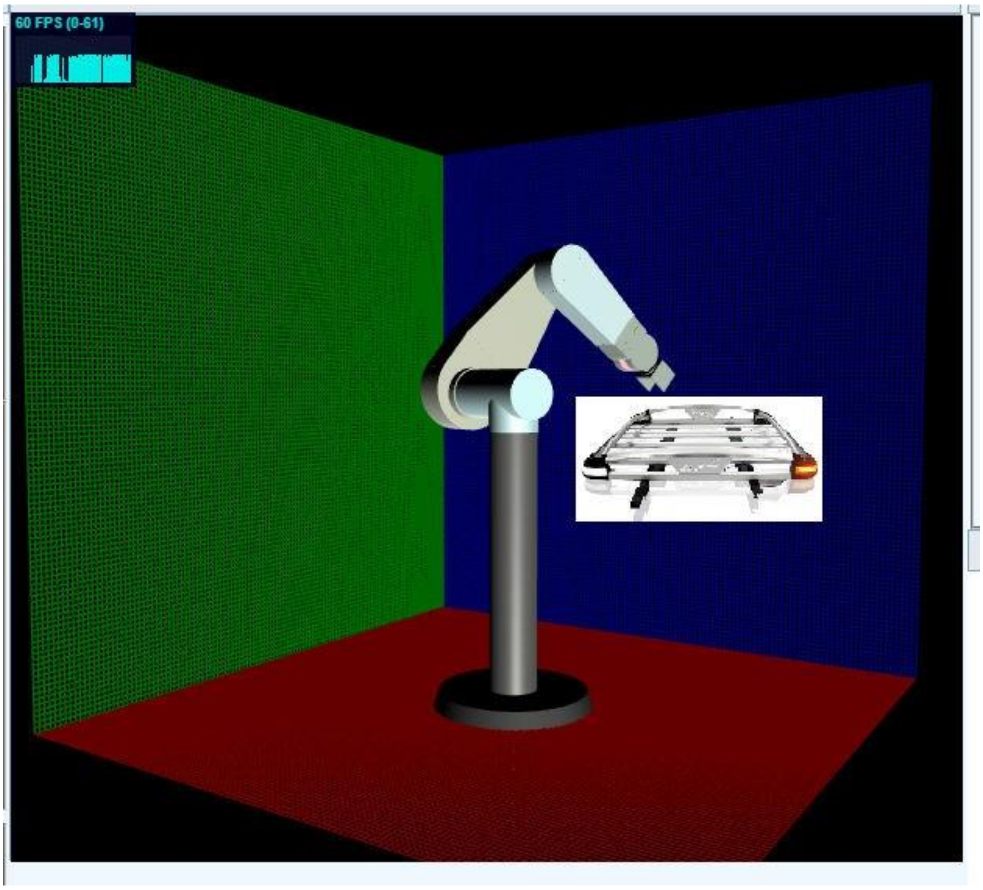


Fig7. Hold an object

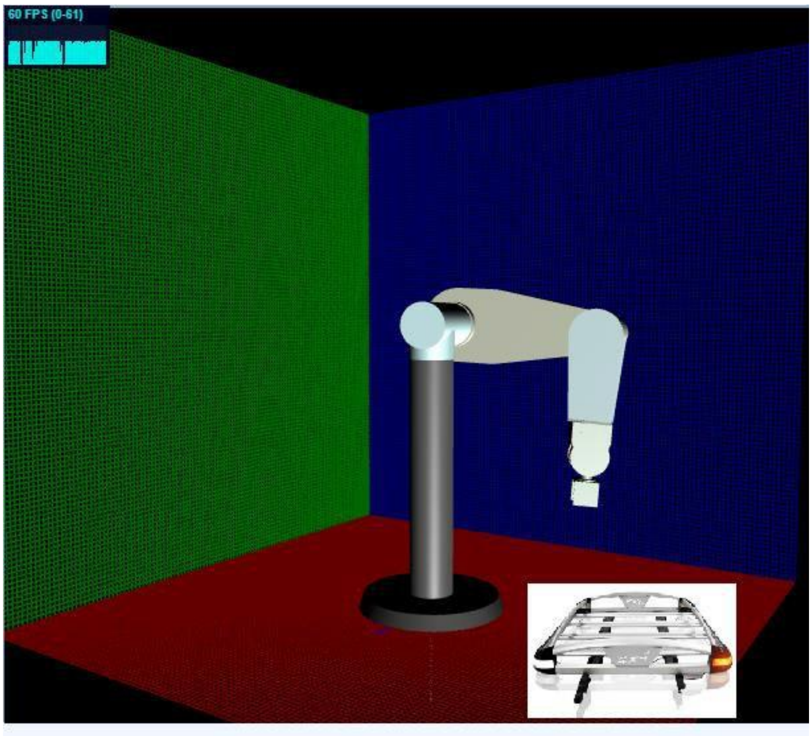


Fig8. Place an object

Result:

- The horizontal position is achieved by joint angle using PUMA 560 robot
- The horizontal position is achieved by assigning the theta values of joints as follows:

Joint	Angle (theta) Initial	Angle (theta) Pick	Angle (theta) Hold	Angle (theta) Place
Waist Joint				
Shoulder Joint				
Elbow Joint				
Wrist Bend				
Wrist Roll				
Wrist Swivel				

EXP.: 9

INVERSE KINEMATICS OF PUMA 560

Aim:

To find values of joint angle using inverse kinematics technique by using the coordinates of manipulator from the transformation matrix,

Objective:

To identify the geometric relationship between input and output motion parameters of PUMA 560 robot manipulator.

To verify the robot configuration for a particular set of joint solution.

Simulate the robot motion for various inputs of the manipulator position.

Theory:

Inverse Kinematics

Inverse kinematics problem deals with the determination of joint variables given a desired position and orientation for the tool. The inverse kinematics problem is significant because manipulation tasks are usually formulated in terms of the desired tool position and orientation. Analysis of Inverse Kinematics is in general more difficult and not straightforward like Forward Kinematics problem. Exact solution of Inverse Kinematics problem for complex configure manipulator is very difficult. The solution of Inverse Kinematics problem for control of a robot manipulator is attempted using various methods such as algebraic methods, geometric methods, and numerical methods. Algebraic methods are used to obtain closed-form solutions, but these methods, do not guarantee closed form solutions.

Given the end-effector position and orientation from Forward kinematics problem, the inverse kinematics approach is used to obtain the joint angles. But as stated in the introduction inverse kinematics is more difficult problem than forward kinematics as its include much complexity. The relationship between forward and inverse kinematics is shown in Figure 1. In general there are two main solution techniques for inverse kinematics problem one is analytic approach and other is numerical method. Analytic approach comprises of geometric and algebraic solutions in which joint variables are solved analytically according to given configuration data.

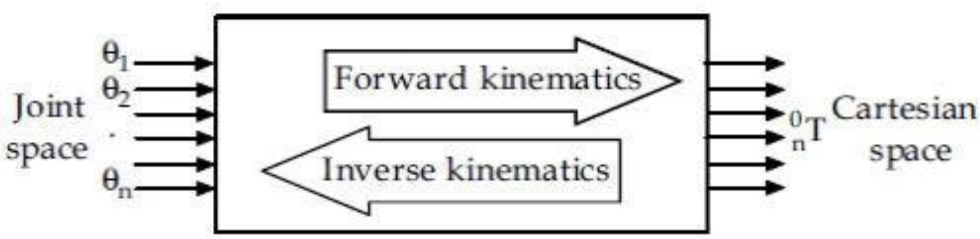
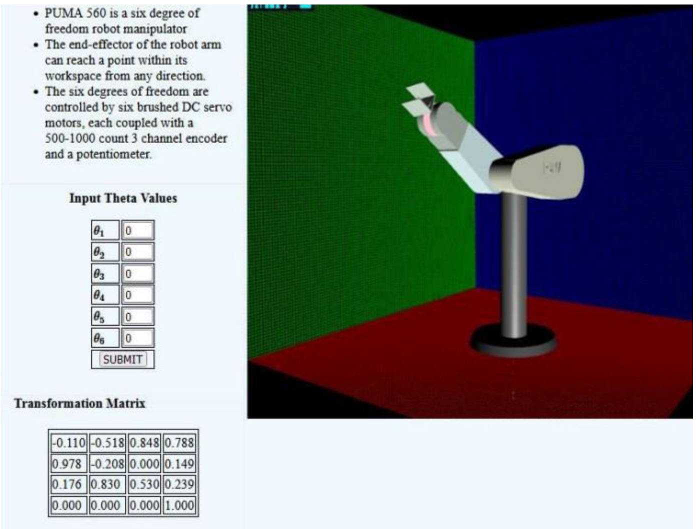


Figure 1. The schematic representation of forward and inverse kinematics.
Procedure:

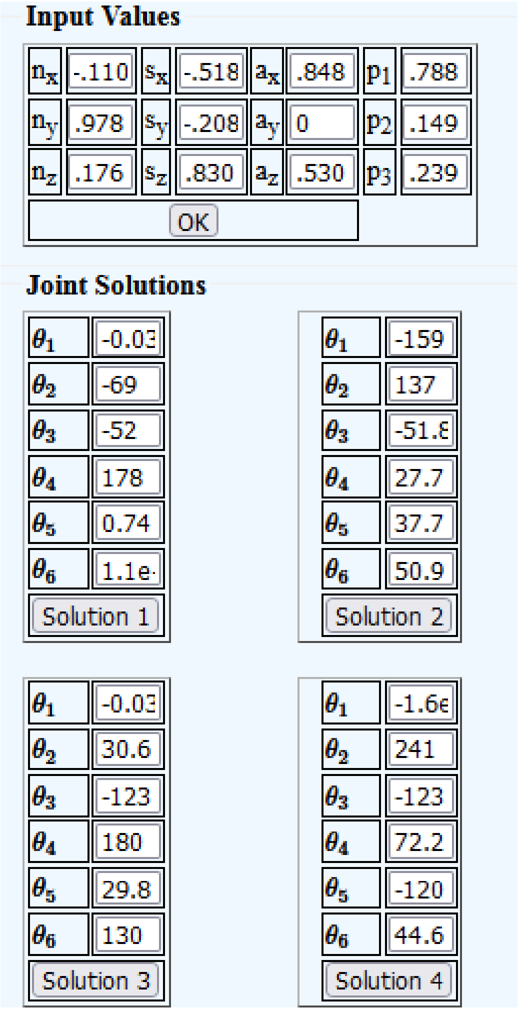
- Insert three position of the end-effector and click OK button to see the joint values.
- Orientation of the manipulator can also be specified.
- The view can be rotated about a point by keeping the left mouse button pressed and rotating the mouse.
- The view can be translated by keeping the right mouse button pressed and translating the mouse in the desired direction.
- The scroll button or middle mouse button can be used for zooming.

Input : Inserting the coordinates value (transformation matrix)



Result:

- Four set of solutions



Input Values

n_x	-.110	s_x	-.518	a_x	.848	p_1	.788
n_y	.978	s_y	-.208	a_y	0	p_2	.149
n_z	.176	s_z	.830	a_z	.530	p_3	.239

OK

Joint Solutions

	Solution 1	Solution 2	Solution 3	Solution 4
θ_1	-0.03	-159	-0.03	-1.6e
θ_2	-69	137	30.6	241
θ_3	-52	-51.8	-123	-123
θ_4	178	27.7	180	72.2
θ_5	0.74	37.7	29.8	-120
θ_6	1.1e-	50.9	130	44.6

EXP.: 10

ROBOT PROGRAMMING – VAL

Aim:

To learn the necessity of Robot Programming, Types, VAL Programming and its commands.

Objective:

The primary objective of robot programming is to make the robot understand its work cycle.

ROBOT PROGRAMMING

The program teaches the robot the following:

- The path it should take
- The points it should reach precisely How to interpret the sensor data
- How and when to actuate the end-effector
- How to move parts from one location to another, and

so forth Programming of conventional robots normally takes one of two forms:

(1) Teach-by-showing, which can be divided into:

- Powered leadthrough or discrete point programming
- Manual leadthrough or walk-through or continuous path programming

(2) Textual language programming

In teach-by-showing programming the programmer is required to move the robot arm through the desired motion path and the path is defined in the robot memory by the controller.

Control systems for this method operate in either:

Teach mode: is used to program the robot

Run mode: is used to run or execute the program

Powered lead through programming uses a teach pendant to instruct a robot to move in the working space. A teach pendant is a small handled control box equipped with toggle switches, dials, and buttons used to control the robot's movements to and from the desired points in the space.

These points are recorded in memory for subsequent playback. For playback robots, thisis the most common programming method used. However, it has its limitations:

- It is largely limited to point-to-point motions rather than continuous movement, because of the difficulty in using a teach pendant to regulate complex geometric paths in space. In cases such as machine loading and unloading, transfer tasks, and spot welding, the movements of the manipulator are basically of a point-to-point nature and hence this programming method is suitable.

Manual lead through programming is for continuous-path playback robots. In walk-through programming, the programmer simply moves the robot physically through the

required motion cycle. The robot controller records the position and speed as the programmer leads the robot through the operation.

If the robot is too big to handle physically, a replica of the robot that has basically the same geometry is substituted for the actual robot. It is easier to manipulate the replica during programming. A teach button connected to the wrist of the robot or replica acts as a special programming apparatus. When the button is pressed, the movements of the manipulator become part of the program. This permits the programmer to make moves of the arm that will not be part of the program. The programmer is able to define movements that are not included in the final program with the help of a special programming apparatus.

Teach-by-showing methods have their limitations:

1. Teach-by-showing methods take time for programming.
2. These methods are not suitable for certain complex functions, whereas with textualmethods it is easy to accomplish the complex functions.
3. Teach-by-showing methods are not suitable for ongoing developments such ascomputer-integrated manufacturing (CIM) systems.

Thus, textual robot languages have found their way into robot technology.

Textual language programming methods use an English-like language to establish the logical sequence of a work cycle. A cathode ray tube (CRT) computer terminal is used to input the program instructions, and to augment this procedure a teach pendant might be used to define on line the location of various points in the workplace. Off-line programming is used when a textual language program is entered without a teach pendant defining locations in the program.

Programming Languages

Different languages can be used for robot programming, and their purpose is to instruct the robot in how to perform these actions. Most robot languages implemented today are a combination of textual and teach-pendant programming.

Some of the languages that have been developed are:

WAVE	VAL
AML	RAIL
MCL	TL- 10
IRL	PLAW
SINGLA	VAL II

VAL II

- It is one of the most commonly used and easily learned languages.
- It is a computer-based control system and language designed for the industrial robots at Unimation, Inc.
- The VAL II instructions are clear, concise, and generally self explanatory.
- The language is easily learned.
- VAL II computes a continuous trajectory that permits complex motions to be executed quickly, with efficient use of system memory and reduction in overall system complexity.
- The VAL if system continuously generates robot commands and can simultaneously interact with a human operator, permitting on-line program generation and modification.
- A convenient feature of VAL If is the ability to use libraries of manipulation routines. Thus, complex operations can be easily and quickly programmed by combining predefined subtasks.

Rules for the location name are as follows:

1. It is any string of letters, numbers, and periods.
2. The first character must be alphabetic.
3. There must be no intervening blank.

4. Every location name must be unique.

5. There may be a limit on the maximum number of characters that can

be used.The following example illustrates the general command format for

VAL II:

100 APPRO P1 15

In this example, 100 is the label that refers to this instruction, APPRO is the instruction tothe robot to approach the location named P1 by a distance of 15 mm.

In the following, we describe the most commonly used VAL II commands.

MOVE P1	This causes the robot to move in joint interpolation motion from its present location to location P1.
MOVES P1	Here, the suffix S stands for straight-line interpolation motion.
MOVE P1 VIA P2	This command instructs the robot to move from its present location to P1, passing through location P2.
APPRO P1 10	This command instructs the robot to move near to the location P1 but offset from the location along the tool z-axis in the negative direction (above the part) by a distance of 10
DEPART 15	Similar to APPRO, this instructs the robot to depart by a specified distance (15 mm) from its present position. The APPRO and DEPART commands can be modified to use straight-line interpolation by adding the suffix S.

DEFINE PATH 1= PATH(P1,P2,P3,P5) MOVE PATH 1	The first command (DEFTNE) defines a path that consists of series of locations P1, P2, P3, and P5 (all previously defined). The second command (MOVE) instructs the robot to move through these points in joint interpolation. A MOVES command can be used to get straight-line interpolation
ABOVE & BELOW	These commands instruct the elbow of the robot to point up and down, respectively.
SPEED 50 IPS	This indicates that the speed of the end- effector during program execution should be 50 inch per second (in./s).

SPEED 75	This instructs the robot to operate at 75% of normal speed.
OPEN	Instructs end effector to open during the execution of the next motion.
CLOSE	Instructs the end-effector to close during the execution of the next motion.
OPENI	Causes the action to occur immediately.
CLOSEI	Causes the action to occur immediately

If a gripper is controlled using a servo-mechanism, the following commands may also be available.	
CLOSE 40 MM	The width of finger opening should be 40 mm.
CLOSE 3.0 LB	This causes 3 lb of gripping force to be applied against the part.
GRASP 10, 100	This statement causes the gripper to close immediately and checks whether the final opening is less than the specified amount of 10 mm. If it is, the program branches to statement 100 in the program
SIGNAL 4 ON	This allows the signal from output port 4 to be turned on at one point in the program and
SIGNAL 4 OFF	Turned off at another point in the program.
WAIT10 ON	This command makes the robot wait to get the signal on line 10 so that the device is on there.

Logarithmic, exponential, and similar functions:

The following relational and logical operators are also available.

EQ Equal to

NE Not equal to
GT Greater than
GE Greater than or equal to
LT Less than

LE Less than or equal

to **AND** Logical AND
operator **OR** Logical
OR
NOT Logical complement

IF (Logical expression) THEN
(Group of instructions)
ELSE
(Group of instructions)
END
	If the logical expression is true, the group of statements between THEN and ELSE is executed. If the logical expression is false, the group of statements between ELSE and END is executed. The program continues after the END statement.
	The group of instructions after the DO statement makes a logical set whose variable value would affect the logical expression with the UNTIL

DO (Group of instructions) UNTIL(Logical expression)	statement. After every execution of the group of instructions, the logical expression is valuated. If the result is false, the DO loop is executed again; if the result is true, the program continues.

TYPE "text“ This statement displays the message given in the quotation marks. The statement is also used to display output information on the terminal.

PROMPT "text", INDEX This statement displays the message given in the quotation marks on the terminal. Then the system waits for the input value, which is to be assigned to the variable INDEX.

In most real-life problems, program sequence control is required. The following statements are used to control logic flow in the program.

GOTO 10This command causes an unconditional branch to statement 10.

SUBROUTINES can also be written and called in VAL II programs. Monitor mode commands are used for functions such as entering locations and systems supervision, data processing, and communications. Some of the commonly used monitor mode commands are as follows:

EDIT (Program name) This makes it possible to edit the existing program or to create a new program by the specified program name.

EXIT This command stores the program in controller memory and quits the edit mode.

STORE (Program name) This allows the program to be stored on a specified device.

READ (Program name) Reads a file from storage memory to robot controller.

LIST (Program name) Displays program on monitor.

PRINT (Program name) Provides hard copy.

DIRECTORY Provides a listing of the program names that are stored either in the controller memory or on the disk.

ERASE (Program name) Deletes the specified program from memory or storage.

EXECUTE (Program name) Makes the robot execute the specified program. It may be abbreviated as EX or EXEC.

ABORT Stops the robot motion during execution.

STOP The same as abort.

required motion cycle. The robot controller records the position and speed as the programmer leads the robot through the operation.

If the robot is too big to handle physically, a replica of the robot that has basically the same geometry is substituted for the actual robot. It is easier to manipulate the replica during programming. A teach button connected to the wrist of the robot or replica acts as a special programming apparatus. When the button is pressed, the movements of the manipulator become part of the program. This permits the programmer to make moves of the arm that will not be part of the program. The programmer is able to define movements that are not included in the final program with the help of a special programming apparatus.

Teach-by-showing methods have their limitations:

4. Teach-by-showing methods take time for programming.
5. These methods are not suitable for certain complex functions, whereas with textualmethods it is easy to accomplish the complex functions.
6. Teach-by-showing methods are not suitable for ongoing developments such ascomputer-integrated manufacturing (CIM) systems.

Programming Languages

Different languages can be used for robot programming, and their purpose is to instruct the robot in how to perform these actions. Most robot languages implemented today are a combination of textual and teach-pendant programming.

Some of the languages that have been developed are:

WAVE	VAL
AML	RAIL
MCL	TL- 10
IRL	PLAW
SINGLA	VAL II

Result

Various commands of VAL II programming studied.

EXP.: 11

1.ROBOT PROGRAMMING – VAL

Exercise 1

Develop a program in VAL II to command a PUMA robot to unload a cylindrical part of 10 mm diameter from machine 1 positioned at point P1 and load the part on machine 2 positioned at P2. The speed of robot motion is 40 in./s. However, because of safety precautions, the speed is reduced to 10 in./s while moving to a machine for an unloading or loading operation.

Solution

SIGNAL 5

SPEED 40 IPS

OPEN 100

APPRO PI, 50

SPEED 10 IPS

MOVE PI

GRASP 10, 100

DEPART P1, 50

SPEED 40 IPS

APPRO P2, 50

SPEED 10 IPS

MOVEP2

BELOW

OPENI 100

ABOVE

DEPART P2, 50

STOP

Sequence:

P1 → Position of M_1 P_2 – Position of M_2

Signal From Port 5 initiated the Robot

↓

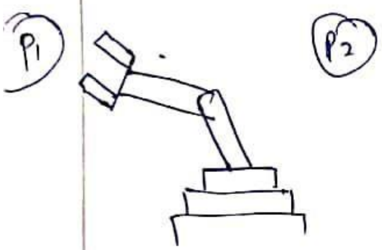
Sequence

Set Robot speed 40 Inch per sec.

↓

Open gripper, Move near to Position 1

↓



Lower speed, Move and grasp the workpiece from P1

↓

Depart slightly and set speed to normal,

↓

Move near to P_2, lower speed

↓

Open gripper, Place, depart to a little,

↓

Back to Home Position or stop.

Result:

VAL II Command Sequence Developed for the given Problem.

Exercise 2

Develop a program in VAL II to command a PUMA robot to insert a job stick in 3 different places P1,P2,P3. Robot has to pick job stick from automatic feeder in P0.

Solution

SIGNAL 5

OPEN 100

SPEED 40 IPS

MOVE P0

GRASP 10

DEPART P1, 50

SPEED 10 IPS

MOVE P1

OPENI 100

DEPART P1, 50

MOVE P0

GRASP 10

DEPART P2, 50

MOVE P2

OPENI 100

DEPART P2, 50

MOVE P0

GRASP 10

DEPART P3, 50

MOVE P3

OPENI 100

DEPART P3, 50

MOVE P0

GRASP 10

DEPART P4, 50

MOVE P4

OPENI 100

DEPART P4, 50

STOP

Sequence:

Signal from port 5 initiated
the robot

↓

Set speed 40 IPS.

↓

Open gripper, move near P1
& Alter speed to 10 IPS

↓

Move P0 and grasp the
Jobstick

↓

Depart a little, alter speed
move to Position 1

↓

Open & drop the Jobstic

↓

Depart little.

↓

Do this Process for Four
Positions.

↓

End.

Result:

VAL II Command Sequence Developed for the given Problem.