www.ijemr.net ISSN (ONLINE): 2250-0758, ISSN (PRINT): 2394-6962

Vandana Publications
IJEMR

Volume-5, Issue-2, April-2015
International Journal of Engineering and Management Research
Page Number: 136-141

Modeling, Simulation and Kinematics Calculations of Stanford Manipulator

M. Sailaja[1], S. Phani kumar[2], M. Raja roy[3]
[1,2]Assistant Professor, Department of Mechanical Engineering, ANITS, Visakhapatnam, Andhra Pradesh, INDIA
[3]Sr. Assistant Professor, Department of Mechanical Engineering, ANITS, Visakhapatnam, Andhra Pradesh, INDIA

ABSTRACT

Robots are very powerful elements of today's industry. They are capable of performing many different tasks and operations with precision and do not require common safety and comfort elements human needs. In designing a robot manipulator, kinematics plays a vital role. The kinematic problem of manipulator control is divided into two types, direct kinematics and inverse kinematics. The mathematical calculations of direct and inverse kinematics based on the theoretical procedure are time taking, difficult, less accurate, slow, and laborious. Robot inverse kinematics, which is important in robot path planning, is a fundamental problem in robotic control. Past solutions for this problem have been through the use of various algebraic or algorithmic procedures, which may be less accurate and time consuming. In the present work an attempt has been made to find direct kinematics and inverse kinematics of STANFORD, SCORBOT, SCARA, ELBOW serial manipulators using the software Visual Basic 6 by using mathematical equations of direct and inverse kinematics and also to create and simulate STANFORD manipulator using the software AutoCAD, AutoCAD ActiveX Automation and Visual Basic Applications (VBA). A program is developed using Visual Basic 6 for generating direct and inverse kinematics of four serial manipulators for the given input data. Using AutoCAD, AutoCAD ActiveX Automation, and VBA, a program is written for simulating STANFORD manipulator.

Keywords**--** AutoCAD ActiveX, Kinematics, Stanford Manipulator, VBA

I. INTRODUCTION

1.1 Autocad ActiveX and VBA

AutoCAD objects are exposed through an ActiveX interface and those objects are programmed using the Visual Basics for the applications programming environment. AutoCAD ActiveX provides a mechanism to manipulate AutoCAD programmatically from within or outside AutoCAD. A capability accompanying VBA is the ability to communicate with other applications such as visual basic using ActiveX automation. Microsoft VBA is an object-oriented programming environment designed to provide rich development capabilities similar to those of Visual Basic (VB). The main difference between VBA and VB is that VBA runs in the same process space as AutoCAD, providing an AutoCAD-intelligent and very fast programming environment. VBA also provides application integration with other VBA-enabled applications. This means that AutoCAD, using other application object libraries, can be an Automation controller for other applications such as Microsoft Word or Excel. VBA sends messages to AutoCAD by the AutoCAD ActiveX Automation interface. AutoCAD VBA permits the VBA environment to run simultaneously with AutoCAD and provides programmatic control of AutoCAD through the ActiveX Automation interface. This coupling of AutoCAD, ActiveX Automation, and VBA provides an extremely powerful interface not only for manipulating AutoCAD objects, but also for sending data to or retrieving data from other applications.

There are three fundamental elements that define ActiveX and VBA programming in AutoCAD. The first is AutoCAD itself, which has a rich set of objects that encapsulates AutoCAD entities, data, and commands. The second element is the AutoCAD ActiveX Automation interface, which establishes messages (communication) with AutoCAD objects. Programming in VBA requires a fundamental understanding of ActiveX Automation. A description of the AutoCAD ActiveX Automation interface can be found in the ActiveX and VBA Reference. Even the experienced VB programmer will find the AutoCAD ActiveX Automation interface invaluable for understanding and developing AutoCAD VBA applications. The third element is the VBA programming environment, which has its own set of objects, keywords, constants, and so forth that provides program flow, control, debugging, and execution. You can view all the VBA projects loaded in the current AutoCAD session by

using the VBA Manager. It is an AutoCAD tool that allows you to load, unload, save, create, embed, and extract VBA projects.
To open the VBA Manager from the Tools menu choose Macro VBA Manager and In AutoCAD invoke the VBAMAN command.

1.2 Kinematic Modeling of Serial Manipulator

A serial consists of several links connected in series by various types of joints, typically revolute and prismatic joints. One end of the manipulator is attached to the ground and the other end is free to move in space. The fixed link is called the base, and the free end where a gripper of a mechanical hand is attached, is called end effector. In designing a robot manipulator, kinematics and dynamics play a vital role. The kinematic model gives relations between the position and orientation of the end effector and spatial positions of joint links.
The kinematic modeling problem is split into two problems as:
1. Given the set of joint link parameters, the problem of finding the position and orientation of the end effector with respect to a known reference frame for an n-DOF manipulator is the first problem. This is referred to as forward kinematics.
2. For a given position and orientation of the end effector (of the n-DOF manipulator), with respect to an immobile or inertial reference frame, it is required to find a set of joint variables that would bring the end effector in the specified position and orientation. This is the second problem and is referred to as the inverse kinematic model or inverse kinematics. Figure 1 shows direct and inverse kinematic models

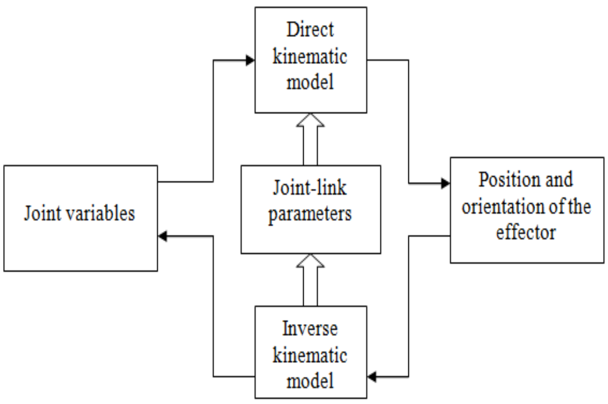


Figure 1 Direct and Inverse kinematic models

The matrix representation of a rigid mechanical link to describe the spatial geometry of a robot arm was first used by Denavit-Hartenberg. The definition of the manipulator with four joint link parameters (a_i, α_i, d_i, θ_i) for each link and a systematic procedure for assigning right handed orthogonal coordinate frames, one to each link in an open kinematic chain, was proposed by Denavit and Hartenberg

a_i = Link length α_i = Link twist d_i = Joint distance θ_i = Joint angle

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

${}^{i-1}A_i$ is the Denavit-Hartenberg homogeneous transformation matrix. For a revolute joint a_i, α_i and d_i are constant, and θ_i is a variable that measures the orientation of the link ι with respect to link ι-1. For a prismatic joint a_i, α_i and θ_i are constant and d_i is a variable that measures the relative location of the link ι with respect to link ι-1. The location of the end effector can be specified by the following 4×4 homogeneous transformation matrix:

$$ {}^{0}A_n = \begin{bmatrix} u & v & w & q \\ 0 & 0 & 0 & 0 \end{bmatrix} $$

Where the upper right 3×1 submatrix describes the position of a reference point Q and the upper left 3×3 submatrix describes the orientation of the end effector. From the geometry of the links, the transformation matrix ${}^{0}A_n$ above can be thought of as the resultant of a series of coordinate transformations beginning form the base coordinate system to the end effector coordinate system. That is,

$$ {}^{0}A_1\, {}^{1}A_2\, {}^{2}A_3 \ldots\ldots {}^{n-1}A_n = {}^{0}A_n = T $$

T is also called arm matrix. T will be in the form

$$ T = \begin{bmatrix} u_x & v_x & w_x & p_x \\ u_y & v_y & w_y & p_y \\ u_z & v_z & w_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

This is the loop closure equation of a serial manipulator, which consists of 16 scalar equations, four of which are trivial. Equating the upper right 3×1 sub matrix results in three independent equations, representing the position of the end effector. Equating the elements of the upper left 3×3 submatrix results in nine equations, representing the orientation of the end effector. The loop closure can be used to solve both direct and inverse kinematic problem.

The procedure for obtaining direct kinematic equations is

1. Write joint link parameters table for a given manipulator.
2. Using the joint link parameters table, obtain the transformation matrices.
3. Combine these transformation matrices to obtain forward kinematic model using loop closure equation.

1.3 Solvability of Inverse Kinematic Model

Inverse kinematics is complex because the solution is to be found for non-linear simultaneous equations, involving transcendental (harmonic sine and cosine) functions. The number of simultaneous equations is also generally more than the number of unknowns, making some of the equations mutually dependent. These conditions lead to the possibility of multiple solutions or nonexistence of any solution for the given end effector position and orientation. The existence of multiple solutions is a common situation encountered in solving inverse kinematics problem. Multiple solutions pose further problem because the robot system has to have a capability to choose one probably the best one.

The procedure for obtaining inverse kinematics is

1. The location of the end effector is given (i.e., ${}^{0}A_{n}$, the right hand part in the loop closure equation).
2. Left hand part of the loop closure equation is obtained by multiplying the corresponding transformation matrixes.
3. Left hand equation consists of n unknowns ($\theta_1, \theta_2 \ldots \theta_n$). Both right and left hand parts represent same transformations.
4. Pre or post multiply loop closure equations by the inverse of the matrix ${}^{i-1}A_{i}$ to obtain alternative loop closure equations.
5. By rearranging the loop closure equations the unknown joint angles are calculated and the problem of inverse kinematics is solved.

II. INTRODUCTION OF FOUR SERIAL MANIPULATORS AND THEIR KINEMATIC ANALYSIS

This paper is concentrated on the mathematical formulation of mainly four serial manipulators

1) STANFORD,
2) SCORBOT,
3) SCARA and
4) ELBOW.

But only for Stanford manipulator direct and inverse kinematic equations are presented .For the remaining manipulators the procedure is same.

2.1 Stanford Manipulator

The Stanford manipulator as shown in Figure 2 is a six-dof industrial manipulator and is characterized by a three degree of freedom arm and three degree of freedom wrist. The first three joints, two revolute and one prismatic, constitute the arm of the spherical (RRP) configuration. The last three revolute joints constitute a wrist of RRR configuration. The first three links are larger in size and are used to position the wrist and the last three links for wrist are small in size and are used to orient the end effector.

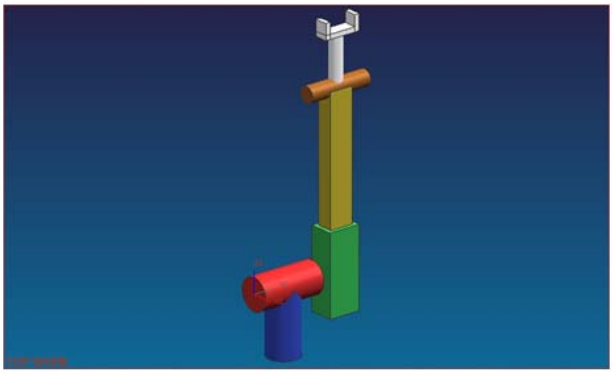


Figure 2 Stanford Manipulator

Table 1 Joint link parameters table of Stanford Manipulator

Joint	θ	α	a	D
1	θ_1	-90	0	0
2	θ_2	90	0	d_2
3	0	0	0	d_3
4	θ_4	-90	0	0
5	θ_5	90	0	0
6	θ_6	0	0	0

$c_1 = \cos\theta_1$, $c_{ijk} = \cos(\theta_i + \theta_j + \theta_k)$

Direct kinematic equations of Stanford manipulator

$$u_x = c_1[c_2(c_6c_4c_5 - s_4s_6) - s_5s_2c_6] - s_1[c_6c_5s_4 + c_4s_6]$$
$$u_y = s_1[c_2(c_6c_4c_5 - s_4s_6) - s_5s_2c_6] + c_1[c_6c_5s_4 + c_4s_6]$$
$$u_z = -s_2c_6c_4c_5 - s_5c_2c_6 + s_6s_2s_4$$
$$v_x = c_1[-c_2(s_6c_4c_5 + s_4c_6) + s_5s_2s_6] - s_1[-s_6c_5s_4 + c_4c_6]$$
$$v_y = s_1[-c_2(s_6c_4c_5 + s_4c_6) + s_5s_2s_6] + c_1[-s_6c_5s_4 + c_4c_6]$$
$$v_z = s_2s_6c_4c_5 + s_5c_2s_6 + c_6s_2s_4$$
$$w_x = c_1[c_2c_4s_5 + c_5s_2] - s_1s_4s_5$$
$$w_y = s_1[c_2c_4s_5 + c_5s_2] + c_1s_4s_5$$
$$w_z = -s_2c_4s_5 + c_5c_2$$
$$p_x = c_1s_2d_3 - s_1d_2$$
$$p_y = s_1s_2d_3 + c_1d_2$$
$$p_z = d_3c_2$$

Inverse kinematic equations of Stanford manipulator

$$\theta_1 = \mathrm{Tan}^{-1}\left(\frac{p_y}{p_x}\right) - \mathrm{Tan}^{-1}\left(\frac{d_2}{+/-\sqrt{r^2-d_2^{\ 2}}}\right) \qquad r = \sqrt{p_x^{\ 2}+p_y^{\ 2}}$$

$$\theta_2 = \mathrm{Tan}^{-1}\left(\frac{c_1 p_x + s_1 p_y}{p_z}\right)$$

$$d_3 = (p_x c_1 + s_1 p_y)\, s_2 + p_z c_2$$

$$\theta_4 = \mathrm{Tan}^{-1}\left[\frac{-s_1 w_x + c_1 w_y}{c_2(c_1 w_x + s_1 w_y) - s_2 w_z}\right]$$

$$\theta_5 = \mathrm{Tan}^{-1}\left(\frac{c_4[c_2(c_1 w_x + s_1 w_y) - s_2 w_z] + s_4(c_1 w_y - s_1 w_x)}{s_2(c_1 w_x + s_1 w_y) + c_2 w_z}\right)$$

$$\theta_6 = \mathrm{Tan}^{-1}\left[\frac{-c_5[c_4(c_2 I - s_2 v_z) + s_4 n] + s_5(s_2 I + c_2 v_z)}{-s_4(c_2 I - s_2 v_z) + c_4 n}\right]$$

$$I = c_1 v_x + s_1 v_y,\ n = -s_1 v_x + c_1 v_y$$

2.2 Scorbot Manipulator

It is a five dof robot. The second, third, fourth joint axes are parallel to one another, the first joint axis points up vertically, and the fifth joint axis intersects fourth perpendicularly. This manipulator is useful for spray painting, spot welding. The Scorbot manipulator is shown below in Figure 3

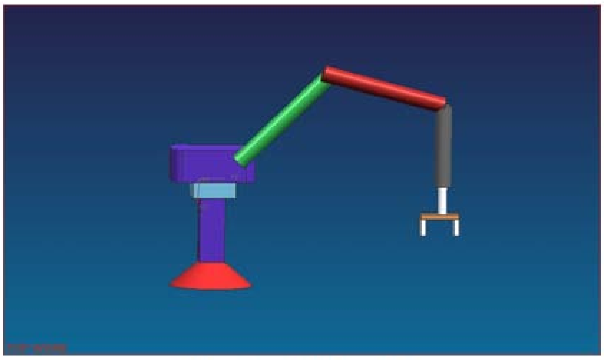


Figure 3 Scorbot Manipulator

2.3 Scara Manipulator

The SCARA arm is an important type of four-dof manipulator. A SCARA arm is constructed with four joint axes parallel to each other. The first two and the fourth are revolute joints and the third is a prismatic joint. The SCARA manipulator is shown below in Figure 4

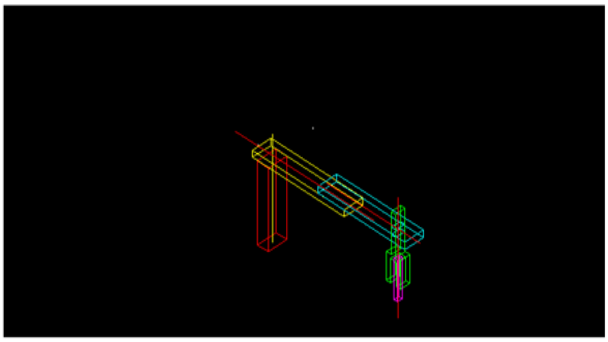


Figure 4 Scara Manipulator

2.4. Elbow Manipulator

The elbow configuration is the most prevalent among existing robot installations. Elbow manipulators are anthropomorphic. The first three axes form the positioning arm and the last three axes form the orienting wrist. This arrangement maximizes the reachable workspace of the robot for a given total link length. It is six degrees of freedom robot and all joints are revolute. The figure 5 shows Elbow manipulator.

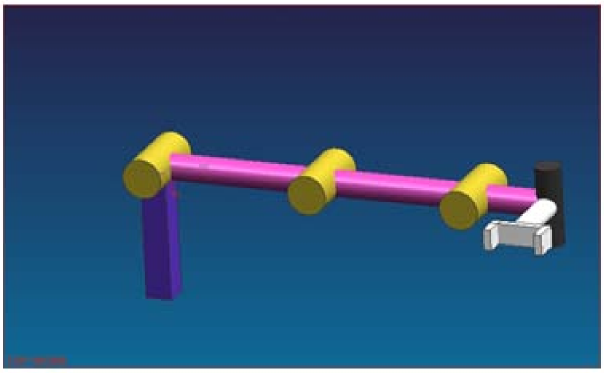


Figure 5 Elbow Manipulator

III. SYSTEM DESIGN

The calculations based on the theoretical procedure are laborious, less accurate and time taking. Hence a program has been developed using visual basic 6 for the mathematical formulation part to calculate the direct and inverse kinematic analysis problem of the four serial manipulators STANFORD, SCORBOT, SCARA, and ELBOW.

3.1Results when calculated manually

Input data for Stanford forward kinematics

$$\theta_1 = 30^0, \theta_2 = 60^0, \theta_4 = 45^0, \theta_5 = 75^0, \theta_6 = 90^0,$$

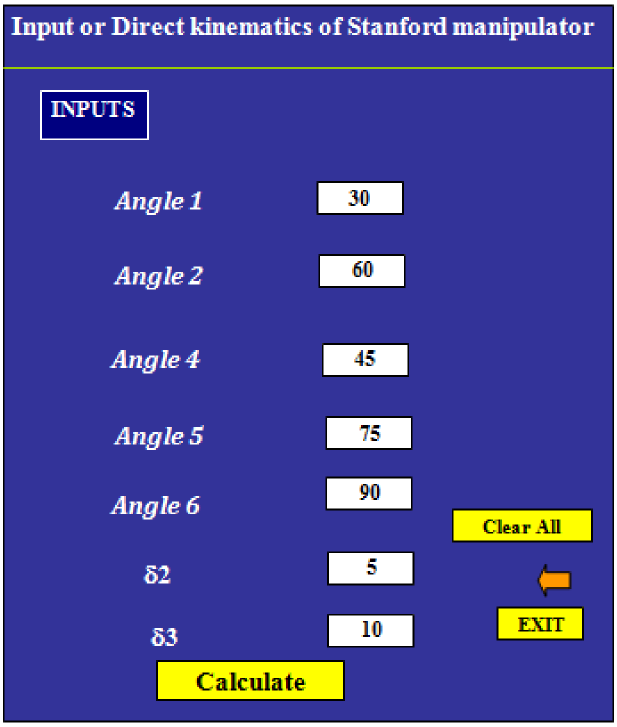
$$d_2 = 5, d_3 = 10$$

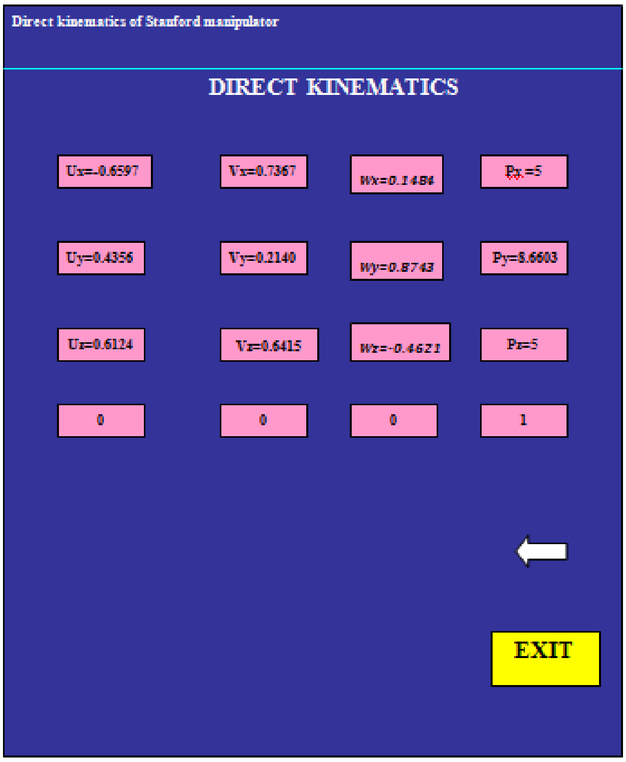
Output data for Stanford forward kinematics

$$\begin{matrix} u_x = -0.6597, & v_x = 0.7367, & w_x = 0.1484, & p_x = 5, \\ u_y = 0.4356, & v_y = 0.2140, & w_y = 0.8743, & p_y = 8.6603, \\ u_z = 0.6124, & v_z = 0.6415, & w_z = -0.4621, & p_z = 5, \\ 0 & 0 & 0 & 1 \end{matrix}$$

3.2 Results obtained using VB

After developing a program it is necessary to check the correctness of the program. Hence the system testing is done. The below Figure shows the input form of the Stanford manipulator and Out of forward and inverse kinematic simulation, only forward kinematic simulation is shown here





Both the results obtained are same

IV. PROCEDURE OF SIMULATION

Step 1

Stanford manipulator diagram is modeled using AUTOCAD.

AUTOCAD provides design-centric collaboration tools, standards, and deployment management features so you can seamlessly and efficiently share information across your design teams. AUTOCAD provides facilities that allow the users to customize autocad to make it more efficient and therefore increase their productivity. Autocad preserves your existing CAD investments by maintaining DWG. Autocad is useful for drawing two and three-dimensional figures.

The following commands are used to model Stanford manipulator.

Circle, Extrude, Line, Move, Polyline, Cylinder, Box

Step 2

The visual basic programming is done in the environment of visual basics with applications. The program consists of one form and the code is written in which the drawing of AutoCAD is called for animation. The description of the form in the program is given below.

Form1

Purpose: To show the forward kinematics simulation of Stanford manipulator.

Contents:

This form consists of twelve labels, six text boxes, and two command buttons

Input: Set of six angles with their range.

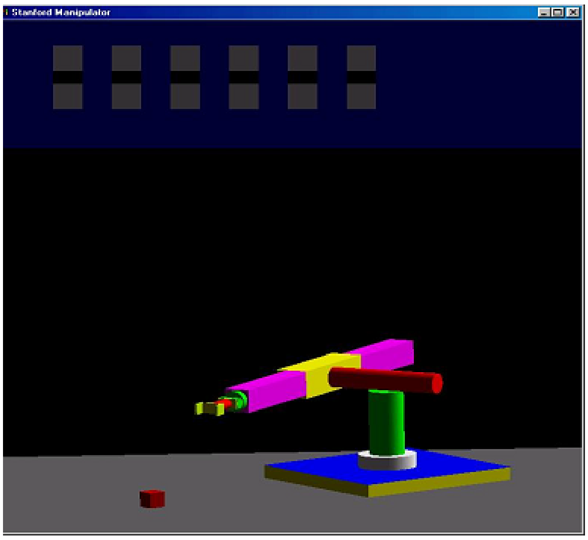
 ISSN (ONLINE): 2250-0758, ISSN (PRINT): 2394-6962

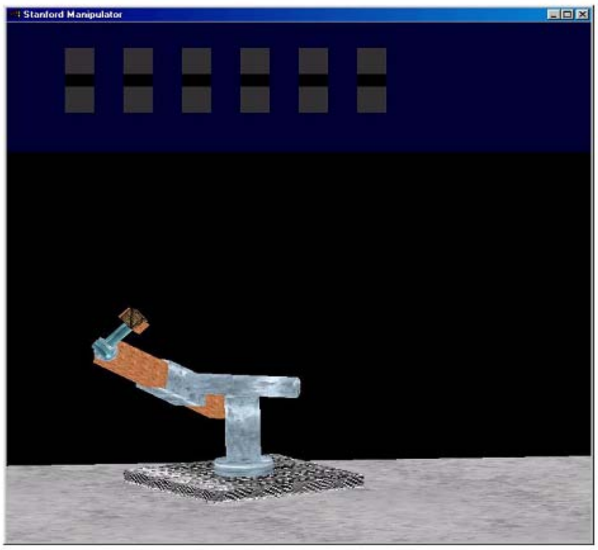
Output: Forward kinematics simulation of Stanford manipulator.

Flow of data:

Click event on the OK button gives a message box that tells us to enter the angles.

Click event on the APPLY button shows the simulation of Stanford manipulator.





V. CONCLUSION

Direct and inverse kinematics of four serial manipulators were calculated using visual basic 6 and modeling and simulation of Stanford manipulator was done using AutoCAD, AutoCAD ActiveX Automation and VBA.

VI. REFERENCES

[1] ROBOT ANALYSIS -The mechanics of serial and parallel manipulators by LUNG-WEN TSAI

[2] A.Srikanth, M. Sravanth, V.Sreechand ,K.Kishore Kumar, Vol4,Issue 4, April 2013 IJETT 'Kinematic anslysis of 3dof of serial robot for industrial applications' .

[3] A study on simulation model and kinematic model of welding robot J.W.Jeong, I.S.Kim, R.R.chand J.H.Lee, Vol55, Issue1, Nov 2012, JAMME.

[4] Kinematic and dynamic analysis of a robot arm used for all Terrain robot Amey V.sutar, Dr S.S.Ohol, Vol 3, Issue 5, May 2014, IJERT

[5] M.A Gonzlez-Palacios The unified orthogonal architecture of industrial serial manipulators robotics and computer integrated manufacturing, 29(1) 257-271,2013.

