International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 2, Issue 1 (jan-feb 2014), PP. 53-55

REVIEW OF VIRTUAL ARTICULATED ROBOT

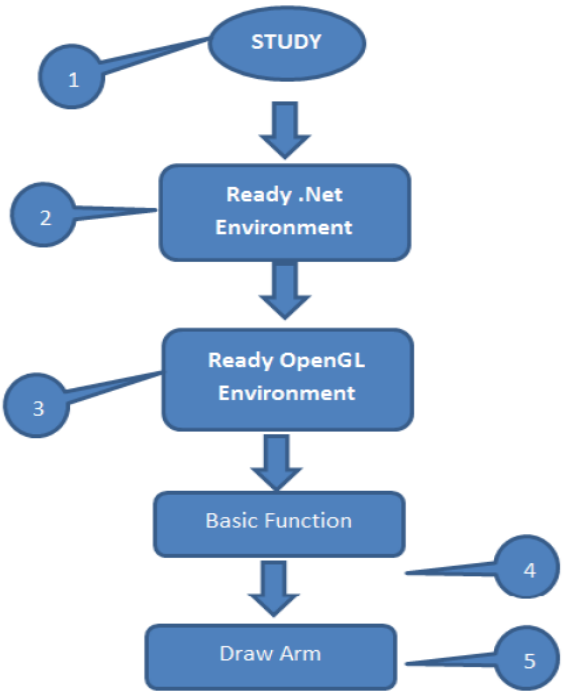
Sudip Chakraborty
Swami Vivekananda University
Madhya Pradesh, India

***Abstract*—In the field of robotic study, practical approach is too much relevant. The successful implementation of different complex algorithm, require practical aspect. At the time of development, major student design a physical robot to clarify the major complex theory. There are some constraints at the initial stage. The major constraints are design, implementation of the coordinate geometry, DH notation, kinematic theory, and so on. All the implementation should be feasible to move the articulated robotic arm. The job is tedious for the beginners. This paper is the reference to build up an articulated robotic arm virtually. Here is a description where an articulated robotic arm can be created virtually, which will act as real robot does. All theory like coordinated geometry, DH notation can be applied.**

***Key words*—5DF Articulated arm, Robot Virtualization, DH-notation, co-ordinate Geometry etc.**

I. INTRODUCTION

Now we are ready to design a 5df articulated robot virtually. Here the following diagram depicts how work flow should be.



II. DESCRIPTION OF BLOCK DIAGRAMS

Study– Many sources are available to gather the Preliminary knowledge to design an articulated robot. I have references few of them. For our recollection, we will review about basic item. At first we have to synchronize our mind that coordination system of computer screen.

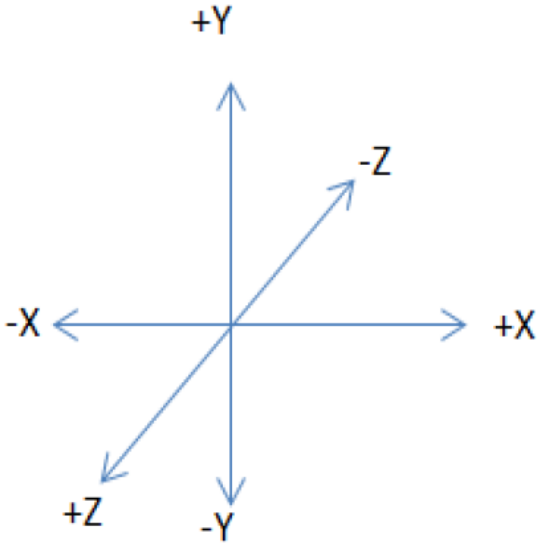


Fig:-1: Computer Screen Coordinate

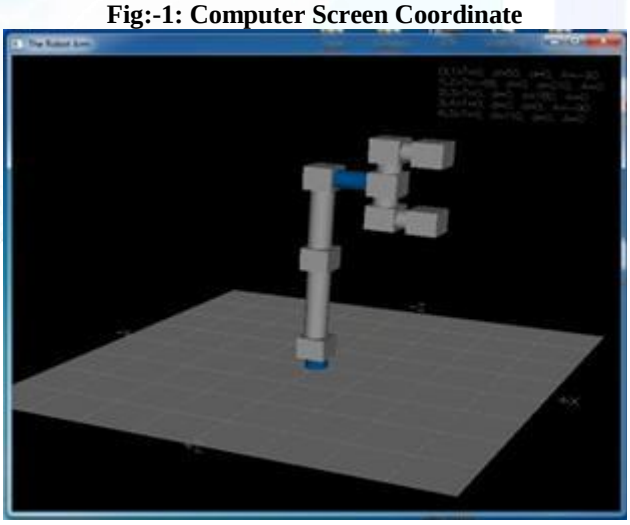


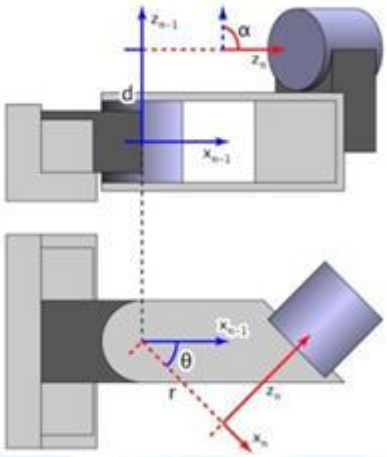
Fig:-2: Screen shot

The DH notation is important. DH notation, Denavit–Hartenberg parameters are

Theta(angle)– angle between Xi-1 and xi axes measured about the Zi-1 axis in the right-hand sense.

d(distance)– Distance along Z axis from the origin of frame

r (radius)– distance or radius along X- axis from the origin of intersection of xi-axis with zi-1 axis to the origin of frame.



Alpha (angle) – Angle between zi-1 and zi axes measured about xi-axis in the right-hand sense.

In the DH notation, Y-axis is eliminated. Only to complete and calculate the 3d geometry we put the value of Y-axis.

We remember that, matrix manipulation is not communicative. So need to maintain the manipulation order. The order is as follows:-

i-1Ti= Rz(theta).Tz(d).Tx(r).Rx(alpha)

i-1Ti= Resultant matrix

Rz(theta) = R-> Rotate 3D matrix ,

z -> along Z-axis

Theta -> amount of rotation angle

Tz(d) = T -> Translation

z-> along Z-axis

D -> amount of translation

Tx(r) = T -> Translation

x -> along x axis

r -> amount of translation is r.

Rx (alpha) = R-> Rotate 3D matrix,

x -> along x axis

Alpha -> amount of rotation angle

The above order is left to right. To test the proceedings, take a 4 by 4 identity matrix.

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

First rotate theta angle along z-axis. on result matrix, translate d amount, along z-axis. Next, on result, translate r amount along X-axis. Last on result, rotate amount of alpha along X-axis. In OpenGL, to implement the above sequence, the commands are as follows –

```
GL.glPushMatrix();
GL.glLoadIdentity();
GL.glRotatef(theta, 0, 1, 0); //theta
GL.glTranslatef(0, d, 0); // d
GL.glTranslatef(r, 0, 0); // r
GL.glRotatef(alph, 1, 0,0);// alpha
GL.glPopMatrix();
```

To view the last result of the matrix, create an array as follows-

```
Static float [] m = new float [16];
```

Add the following code-

```
GL.glGetFloatv(GL.GL_MODELVIEW_MATRIX, m);
```

View the last result. Now change the manipulation order and observe the result is different on different manipulation. So strictly maintain the order otherwise robot's orientation will not move according to our desire position.

III. READY .NET ENVIRONMENT

To create the virtual robot we need visual studio, dot net 2010 and C sharp. Step up as follows-

Step -01) create a folder on desktop (or your choice) rename "test".

Step-02) download a zip file from the following link:- https://sites.google.com/site/leniel/blog/RobotArmOpenGLCSharp.zip?attredirects=0

step-03) unzip the file. Copy the following file and paste to "test" folder.

freeglut.dll, glut32.dll, Tao.FreeGlut.dll, GLU.cs, OpenGL.cs

IV. READY OPENGL ENVIRONMENT

step-01) open VS2010> File> New> project> Windows Forms Application> Type a name> ok.

Step-2) Delete Form1.cs from solution explorer

step-3) go to the solution explorer window -> right clock on project name ("test") -> add -> existing item ->select GLU.cs, OpenGL.cs -> press "Add" button.

Step-4) right click on project name -> Add Reference -> browse -> select Tao.FreeGlut.dll -> Ok

V. BASIC FUNCTION

Some basic function that forms a robot:-

```
Static void Draw_Link(float th,float _d, float _a, float alph)
```

The above function is the main function to create different link. Four parameters are passed to it. In spite of them only theta are dynamic for R-joint or Revolute joints. And for prismatic joint or P-joints, d is dynamic.

```
Static void DrawUnitCylinder(int numSegs)
```

The above function create joint with shaded number of shade by "numSegs" parameter. It creates a 3D effect on material of robot.

```
Static void Create_Work_Space(void);
```

This function can be used to create the workspace.

The above and other required function should be in main loop, so that every time OpenGL can create all node and link. Display buffer imposed on previous drawing. The above

functions are for concept only. Many more function may come to create a realistic robot.

VI. DRAW ARM

Degree of freedom means number of movable or rotatable joints. When draw a link, manipulate matrix on previous. So that robot orientation can build up as a tree like structure. It starts with Base0 that is immobile. Like root of the tree. Now create a new matrix on base matrix. Create a new link means transforming the previous matrix, yield new matrix that is an arm, and adds one degree. Generally Except theta, all parameters are static (for R-joint), that is initially defined. Entire project these parameters (d, r, alpha) are not required to change.

VII. CONCLUSION

Now a day's computer is available to everyone. Creation anything virtually is cheap and less time consume. It is also more flexible to move anywhere. Even in train we can work with the robot, where practical robot is less flexible to move anywhere, anytime. The main purpose to write this paper is, before to create a practical robot, test the robot virtually. The concept will be more cleared to design a practical robot.

REFERENCES

[1] Mittal, R.K & Nagrath, I.J "Robotics and Control", Tata McGraw-Hill Education Private Limited, ISBN-10: 0-07-048293-4. 2012, Page no- 71-111

[2] Saha, S.K "Introduction to Robotics", Tata McGraw-Hill Education Private Limited, ISBN-10: 0-07-066900-0. 2011, Page no- 113-121

[3] Niku, Saeed B. "Introduction to Robotics, Analysis, Systems, Applications", PHI Learning Private Limited, ISBN-978-81-203-2379-7, 2009, Page no- 113-121

[4] Klafter , Richard D. , Chmielewski, Thomas A., Negin, Michael "Robotic Engineering: An Integrated Approach", PHI Learning Private Limited, ISBN-978-81-203-0842-8,2009, Page no- 610-619

[5] Blankenship, J and Mishal, S "Robot Programmer's Bonanza",BPB Publications, ISBN-10:81-8333-290-0, 2008, Page no-65-72

[6] Appin Knowledge Solutions "Robotics",BPB Publications, ISBN-10:81-8333-212-9, 2007, Page no-189-215

[7] http://colinfahey.com/csharp_wrapper_for_opengl/csharp_wrapper_for_opengl.html

[8] https://sites.google.com/site/leniel/blog/RobotArmOpenGLCSharp.zip?attredirects=0

[9] http://www.glprogramming.com/blue/ch03.html

[10] http://www.glprogramming.com/blue/ch03.html#id46859

[11] http://www.movesinstitute.org/~mcdowell/mv4202/notes/lect9.pdf

[12] https://www.cs.duke.edu/courses/cps124/fall09/notes/05_pipeline/cox_04transformations.pdf

[13] http://en.wikipedia.org/wiki/Denavit–Hartenberg_parameters.