

Gantry Robot Kinematic Analysis User Interface Based on Visual Basic and MATLAB

Mahir Abdelwahid Ibrahim Ismail¹, Mohammed Khalafalla Mohammed²

^{1,2}Mechanical Engineering Department, Tianjin University of Technology and Education, Tianjin 300222, China

Abstract: Affordable and competitive industrial automation is of key importance for small and medium enterprises, in World. A key factor is the introduction of new robot automation concepts that ease fast deployment and extend available task repertoire. A range of software tools and methods were found to be useful and necessary for efficient engineering and integration. For experimental evaluation, a full-scale prototype robot was designed and built, robot CAD software was adapted to the configuration needs, and both simulations and physical experiments were carried out. In this paper we created a GUI in Visual Basic.NET which is communicate with Matlab via COM Server. Here Visual Basic is a client and Matlab is a server. GUI is done to simplify Gantry Robot Kinematic Analysis Based On D.H Method to be easy for user to change the design parameter for studying of robot kinematic behavior, Our findings make us believe that enhanced software tools should be integrated on a higher symbolic (or meta-) level to better support transformation of data and code generation, and this bring a new dimension of flexibility into manufacturing.

Keywords: Kinematic Analysis, Matlab, gantry Robot, user interface, visual basic.

1. Introduction

New low-cost and flexible robot concepts are needed to fulfill the needs of small- and medium-sized enterprises (SMEs) in manufacturing; SMEs depend on their ability to cost efficiently produce customized products, and the use of manual labor is common to accomplish the required flexibility. To maintain profitability on a global market, there is a desire to have robots that in an efficient way can assist human workers. This would require robots to be much more flexible to configure and use, and in many cases much more stiff in the sense of motion compliance compared to traditional industrial robot arms ^[1]. We are made of bones, muscles and senses. We control using muscles and measure with senses: touch, vision, etc. Robots are built with links and joints in various configurations. Robot without intelligence can only control and measure the joints directly, such as rotate joint 1 for 300 pulses. We call this joint coordinates (you can also consider angles). To accomplish a task in an application, we need to control the position and orientation in various coordinate systems such as world, work piece to tool. The primitive robot does not know the relationships between joint coordinates and other coordinate systems. It is very difficult to be used in applications ^[2]. That separates a toy robot from an industrial robot. In order for a robot to go to certain place at certain orientation conveniently, it is necessary to know the relationship between the joint coordinate system and some other systems, such as base or tool systems. A configuration of a manipulator is a complete specification of the location of every point on the manipulator. The set of all possible configurations is called the configuration space. In our case, if we know the values for the joint variables (i.e., the joint angle for revolute joints, or the joint offset for prismatic joints), then it is straightforward to infer the position of any point on the manipulator, since the individual links of the manipulator are assumed to be rigid, and the base of the manipulator is assumed to be fixed.

2. Robotic System

A robot manipulator should be viewed as more than just a series of mechanical linkages. The mechanical arm is just one component in an overall Robotic System, illustrated in Figure (1), which consists of the arm, external power source, end-of arm tooling, external and internal sensors, computer interface, and control computer. Even the programmed software should be considered as an integral part of the overall system, since the manner in which the robot is programmed and controlled can have a major impact on its performance and subsequent range of applications.

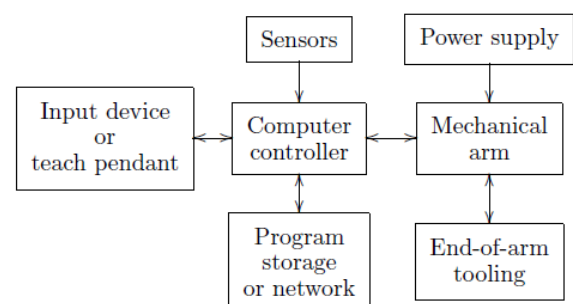


Figure 1: Components of a robotic system

3. Kinematic Analysis

Robot kinematics is mainly of the following two types: forward kinematics and inverse kinematics. As seen earlier, there are two types of coordinates that are useful for describing the configuration of the system. If we focus our attention on the task and the end effector, we would prefer to use Cartesian coordinates or end effector coordinates. The set of all such coordinates is generally referred to as the Cartesian space or end effector space. The other set of coordinates is the so called joint coordinates that is useful for describing the configuration of the mechanical linkage. The set of all such coordinates is generally called the joint space. In robotics, it is often necessary to be able to “map” joint coordinates to end effector coordinates. This map or the procedure used to obtain end effector coordinates from joint coordinates is called direct kinematics. The analysis or

procedure that is used to compute the joint coordinates for a given set of end effector coordinates is called inverse kinematics. Basically, this procedure involves solving a set of equations. However the equations are, in general, nonlinear and complex, and therefore, the inverse kinematics analysis can become quite involved. Also, as mentioned earlier, even if it is possible to solve the nonlinear equations, uniqueness is not guaranteed. There may not (and in general, will not) be a unique set of joint coordinates for the given end effector coordinates.

4. Kinematic Model of 3 DOF Gantry Robot

The kinematic analysis of gantry robot, which has affixed base (Module Y) and 4 links (Module X, Module Z, Arm, and End-effector) With 3DOF in translation movements, connected as an open kinematic chain, as can be seen in fig(3), is referred in the joints space coordinates. The design parameters used in this article are based on gantry robot model in solidworks environment as shown in fig (2).

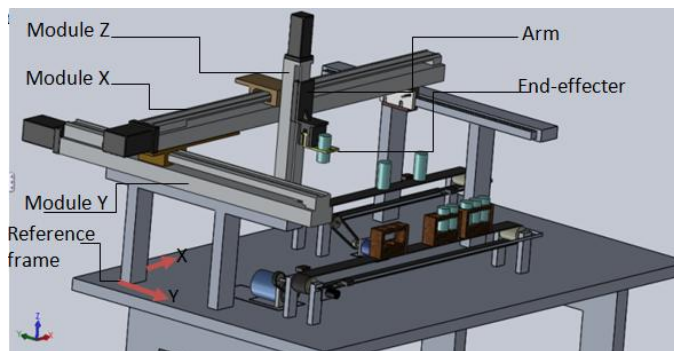


Figure 2: Gantry robot system

Table 1: D-H parameters

Link (i)	a_i (mm)	d_i (mm)	α_i	θ_i
1	a_1 (constant)	d_1 (variable)	$\alpha_1=(\pi/2)$	$\theta_1=0$
2	a_2 (constant)	d_2 (variable)	$\alpha_2=(\pi/2)$	$\theta_2=(\pi/2)$
3	$a_3=0$	d_3 (variable)	$\alpha_3=0$	$\theta_3=0$

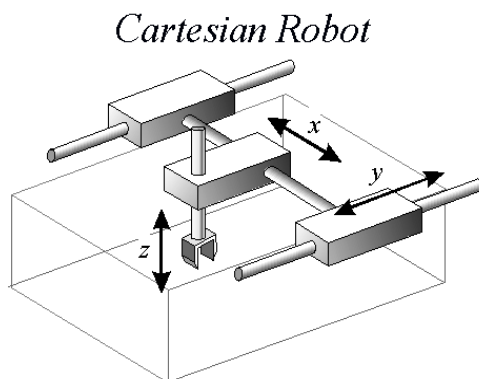


Figure 3: Gantry robot simplified model

5. Position Analysis of Gantry Robot

By using D-H method we can find homogenous transfer matrix as follow

$${}^{i-1}A_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

From Table (1) we can obtain 0A_1 , 1A_2 , AND 2A_3

6. Forward Kinematic Analysis

The position of end-effector (3q) with respect to the base coordinate system (0q) was found and the final equations which describe the kinematic analysis of gantry robot are shown below:

$${}^0q = {}^0A_3 {}^3q \quad (2)$$

$$q_u = q_z - a_3 - a_2 - d \quad (3)$$

$$q_v = q_y - d_2 \quad (4)$$

$$q_w = q_x - a_1 - d_3 \quad (5)$$

q_u , q_v , and q_w are the end effector position with respect to base coordinate system.

7. Using Graphical User Interface (GUI)

It is very important when we do not have enough time to learn using software. So using (GUI) makes software very easy and decrease the time that need to learn it. This fact implies that there will be less time devoted to the software applications being covered. In Matlab we can make GUI non-visually by writing Matlab code which determines the properties of the controls that it seems very difficult and complicated. In Visual Basic we can make GUI visually by choose the control we need from the control menu and double click with left mouse on it then adjust its properties in the properties window that it seems very easy and simplify as shown in figure (4).

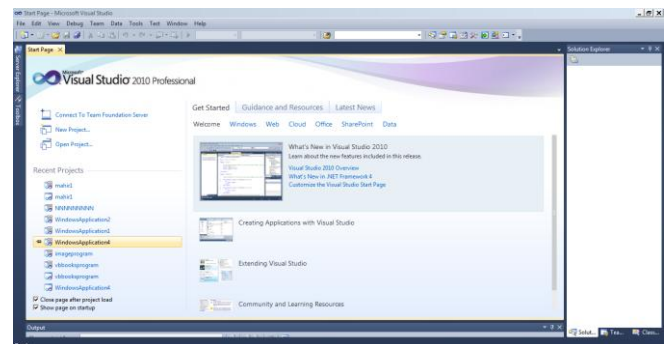


Figure 4: Visual Basic control menu window

Client Specific Information:

In fact how you create Automation server depends on the software you are using. Consult the documentation for your software for this information. All softwares require a programmatic identifier (ProgID) to identify the server. The ProgID registered for MATLAB is matlab.application. Here

our software is a MATLAB application, so we create the Automation server using the MATLAB actxserver function:
`h = actxserver('matlab.application')`
`h = COM.matlab.application`

8. Gantry robot kinematic analysis User Interface Study & Result

Our GUI is constructed from only one form which contained ten label control to define the names of textbox controls, six button control to controlling the execution of program, and nine textbox control which are used by the user to put his own values of variables of the D.H method as shown in figure (5).

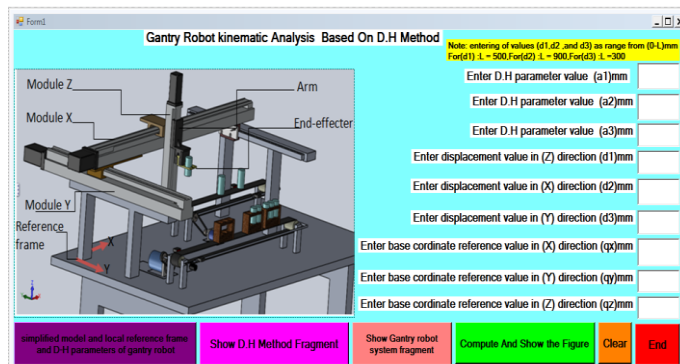


Figure 5: Gantry robot kinematic analysis User Interface

9. Procedures

- Open –start menu-choose Microsoft Visual Studio 2010-then click left mouse on it (do not run Matlab) as shown in figure (4).
- Run the program (by click left mouse on run icon) the program window display as shown in figure (5).
- Enter the nine (9) values of D.H parameters in the coordinating textboxes as shown in figure (6).

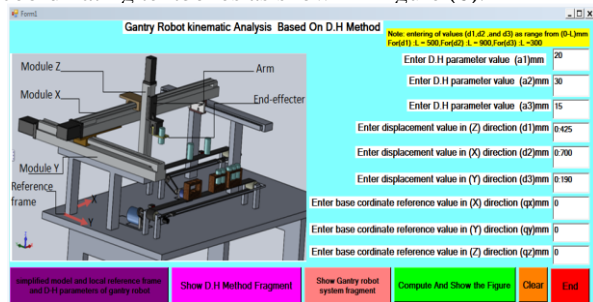


Figure 6: Entering D.H parameters values (run mode)

- Click left mouse on command button named (show D.H Method fragment) to see small fragment about D.H Method as shown in figure (7).

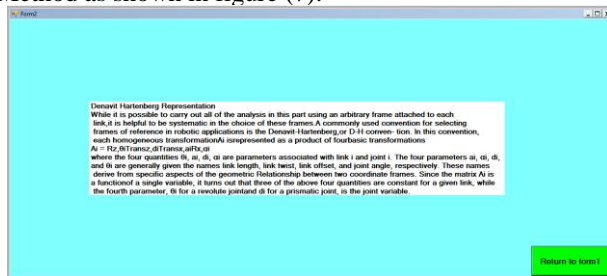


Figure 7: D.H Method fragment

- Click left mouse on command button named (show D.H Method Diagram) to see simplified model and local reference frame and D-H parameters as shown in figure (8).

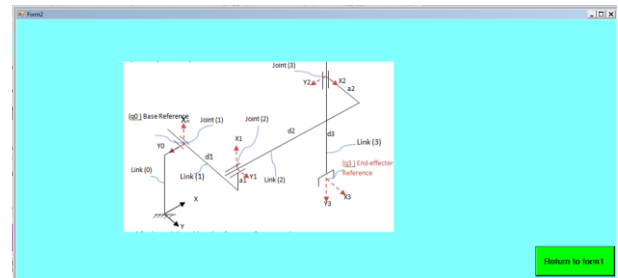


Figure 8: simplified model and local reference frame and D-H parameters

- Click left mouse on command button named (show Gantry robot system fragment) to see small fragment about Gantry robot as shown in figure (9).

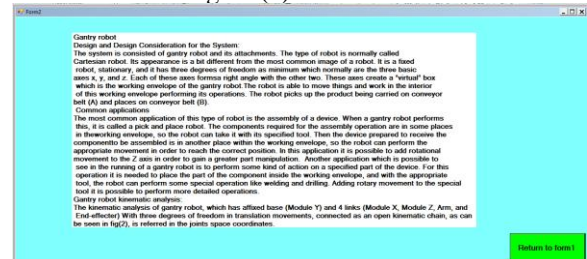


Figure 9: Gantry robot system fragment

- Click left mouse on command button named (compute and show figure) to see the results according to your inputs (the figure is appear immediately) as shown in figure (10).

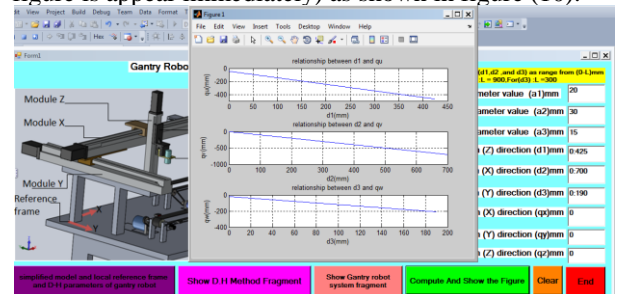


Figure 10: Result after computation process

- Click left mouse on command button named (clear) to clear your old inputs and re-enter new inputs to see another results.

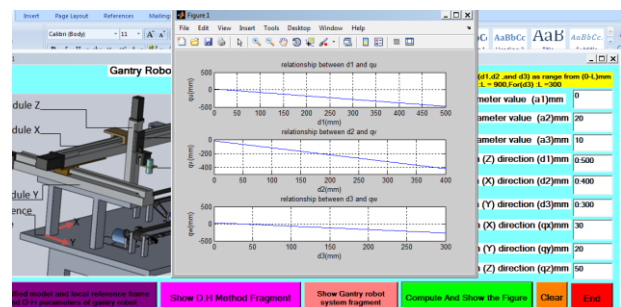


Figure 11: New input and new result

- Click left mouse on command button named (end) to end using of the application.

10. Conclusion

A complete analytical solution to the forward kinematics of gantry Robot is derived in this Paper and the problems was mentioned above was fixed. The forward kinematic analysis of gantry robot is investigated. The mathematical model is prepared and solved for positioning of the end-effectors by preparing a program in MATLAB. Furthermore GUI is done using visual basic to simplify Gantry Robot Kinematic Analysis Based on D.H Method and now it is become easy for researchers and students to study the kinematic of gantry robot and changing the parameters to see the behavior of the system for getting best design and configuration of robot.

11. Acknowledgment

This paper is supported by the National Youth Science Fund Project (601301040); Tianjin Science and Technology Planning Project (13ZCZDGX01500, 14ZCZDSF00022) and the Scientific and Technological Innovation Special Project of Xiqing District, Tianjin (XQCXZX2012-003).

References

- [1] MAHIR A. I. I, Deng Sanpeng -Research on a Gantry Robot Based on PLC-Applied Mechanics and Materials Vols. 716-717 (2015) pp 1654-1657-© (2015) Trans Tech Publications, Switzerland.
- [2] VIVEK DESHPANDE1 & P M GEORGE2, KINEMATIC MODELLING AND ANALYSIS OF 5 DOF ROBOTIC ARM, International Journal of Robotics Research and Development, Apr 2014.
- [3] The Language of Technical Computing- MATLAB External Interfaces, © COPYRIGHT 1984 - 2005 by The MathWorks, Inc.
- [4] Norton, Peter peter Norton's Guide to Visual Basic 4, SAMS Publishing- Indianapolis, IN.
- [5] C. V. Ferreira - V. F. Romano, a Design Methodology for the Compensation of Positioning Deviation in Gantry Manipulators.
- [6] COŞKUN YETİM, KINEMATIC ANALYSIS FOR ROBOT ARM, SENIOR PROJECT, Istanbul, 2009.
- [7] The use of the Mitsubishi PLC systems in student's preparation for realization of industrial tasks J. Świder, K. Foit.
- [8] Hugh Jack , Automating Manufacturing Systems with PLCs.
- [9] Annamareddy Srikanth1 Y.Ravithej2 V.Sivaraviteja3 V.Sreechand4, Kinematic Analysis and Simulation of 6 D.O.F. Of Robot for Industrial Applications, (IJERA), March -April 2013.
- [10] Lewis, F.L.; et. al. "Robotics" Mechanical Engineering Handbook Ed. Frank Kreith Boca Raton: CRC Press LLC, 1999.
- [11] Edited by Thomas R. Kurfess, Robotics and automation handbook.
- [12] R.V. Sakrikar, P.V. Sarngadharan, S. Sharma, V.K. Shrivastava, A.P. Das, V. Dave, N. Singh, P.K. Pal and Manjit Singh, A Material Transfer System using Automated Guided Vehicle.