International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:6, 2007

Software Development for the Kinematic Analysis of a Lynx 6 Robot Arm

Baki Koyuncu, and Mehmet Güzel

***Abstract*—**The kinematics of manipulators is a central problem in the automatic control of robot manipulators. Theoretical background for the analysis of the 5 Dof Lynx-6 educational Robot Arm kinematics is presented in this paper. The kinematics problem is defined as the transformation from the Cartesian space to the joint space and vice versa. The Denavit-Harbenterg (D-H) model of representation is used to model robot links and joints in this study. Both forward and inverse kinematics solutions for this educational manipulator are presented, An effective method is suggested to decrease multiple solutions in inverse kinematics. A visual software package, named MSG, is also developed for testing Motional Characteristics of the Lynx-6 Robot arm. The kinematics solutions of the software package were found to be identical with the robot arm's physical motional behaviors.

***Keywords*—**Lynx 6, robot arm, forward kinematics, inverse kinematics, software, DH parameters, 5 DOF ,SSC-32 , simulator.

I. Introduction

THE transformation between the joint space and the Cartesian space of the robot is very important. Robots are operated with their servo motors in the joint space, whereas tasks are defined and objects are manipulated in the Cartesian space [1].

The kinematics solution of any robot manipulator consists of two sub problems forward and inverse kinematics. Forward kinematics will determine where the robot's manipulator hand will be if all joints are known whereas inverse kinematics will calculate what each joint variable must be if the desired position and orientation of end-effector is determined Hence Forward kinematics is defined as transformation from joint space to cartesian space whereas Inverse kinematics is defined as transformation from cartesian space to joint space.

In this study, the standard Denavit-Harbenterg [2] approach has been used for modeling the Lynx-6 Robot Arm seen in Fig. 1. Many industrial robot arms are built with simple geometries such as intersecting or parallel joint axes to simplify the associated kinematics computations [3]. But their costs are high for students and research workers.

Lynx-6 is a good alternative for such robot manipulators, because it's inexpensive, flexible and similar to industrial robot arms.

There is a large amount of literature which discusses the kinematics analysis of industrial robots [4]. Majority of them shy away from discussing the low cost educational robot arms. So in this paper mathematical model and kinematical analysis of the Lynx-6 educational robot arm is studied. A visual software program was also developed to show the robot arm motion with respect to its mathematical analysis.

Manuscript received August 28, 2007.

Authors are from the Computer Engineering Department of Ankara University, Ankara, Turkey (e-mails: bkoyuncu@ankara.edu.tr, mguzel@eng.ankara.edu.tr).

Fig. 1 Lynx 6 Robotic arm

II. Description

Lynx 6 robot arm has 5 directions of motion (DOF) plus a grip movement (5+1). It is also similar to human arm from the number of joints point of view. These joints provide shoulder rotation, shoulder back and forth motion, elbow motion, wrist up and down motion, wrist rotation and gripper motion. A graphical view of all the joints was displayed in Fig. 2.

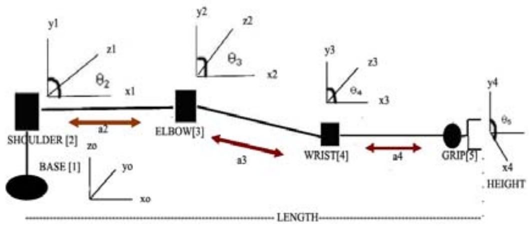

Fig. 2 Modeling of Lynx-6 Robotic Arm

Lynx-6 has five rotational joints and a moving grip. **Joint 1** represents the Base and its axis of motion is z_0. This joint provides a rotational θ_1 angular motion around z_0 axis in x_0y_0 plane. **Joint 2** is identified as the shoulder and its axis is perpendicular to Joint 1 axis. It provides an angular motion θ_2 in x_1y_1 plane. z axes of **Joint 3** (Elbow) and **Joint 4** (Wrist) are parallel to **Joint 1** z axis, They provide θ_3 and θ_4 angular motions in x_2y_2 and x_3y_3 planes respectively. **Joint 5** is identified as the grip. Its z_4 axis is vertical to z_3 axis and it provides θ_5 angular motion in x_4y_4 plane.

Lynx-6 Robot Arm rotational joints and the grip are controlled by dedicated servo motors. These motors are connected to a serial servo controller card (SSC32) to control the Lynx-6 from a computer through the serial port, Fig. 3.

A sequence of 3 consecutive unsigned bytes is sent to serial servo controller from the computer. These are the default sync_byte, the joint servo identifier byte and the

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:6, 2007

desired position byte. Rotational position of a servo motor is determined by a specific angle. This angle value, provided by the computer, was digitized to generate discrete motional steps by the SSC-32 card [5].

SSC-32 servo control card provides the hardware interface between computer and the robot arm. It has a time resolution of 1µs for accurate positioning and a dc motor control to generate extremely smooth moves. The time range is 0.50mS to 2.50mS for an angular range of 0° to 180°. The card generated motion can be a speed controlled, time controlled or a combination (speed and time) motion.

Fig. 3 SSC-32 Servo Controller Card

III. KINEMATICS

Denavit-Hartenberg, [6], representation is used to model the joints of Lynx-6 Robot arm in Fig. 2. All the joints are assigned by using the principles of D-H convention.

The D-H parameters, well explained in [6], are routinely noted d_i (motional distance along z_0 axis), a_i (distance between joints along x axis), θ_i (angle around the z axis on xy plane, α_i (angle between 2 adjacent z axes). These parameters describe the location of a robot link-frame F_i (a joint) from a preceding link-frame F_{i-1} (previous joint) through the sequence of translations and rotations.

D-H parameters for Lynx-6 are defined for the assigned frames in Table I. For example Frame 5 is the grip frame with attached end effectors at joint 5.

TABLE I
THE D-H PARAMETERS FOR LYNX-6

Frame (joint)	θ_i degree	d_i cm	a_i cm	α_i degree
1	θ_1	0	0	90
2	θ_2	0	a_2	0
3	θ_3	0	a_3	0
4	θ_4	0	a_4	-90
5	θ_5	0	0	0
6	0	0	0	0

By substituting these parameters; the transformation matrices A_1 to A_6 can be obtained as shown in Fig. 4. For example A_1 shows the transformation between frames 0 and 1 (designating C_i as $\cos\theta_i$ and S_i as $\sin\theta_i$ etc).

$$A_1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & C_2a_2 \\ S_2 & C_2 & 0 & S_2a_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} C_3 & -S_3 & 0 & C_3a_3 \\ S_3 & C_3 & 0 & S_3a_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_4 = \begin{bmatrix} C_4 & 0 & -S_4 & C_4a_4 \\ S_4 & 0 & C_4 & S_4a_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 4 Transformation Matrices for Lynx-6

A. Forward Kinematics

Calculating the position and orientation of the end effectors with given joint angles is called Forward Kinematics analysis. Forward Kinematics equations are gene- rated from the transformation matrixes shown in Fig. 4 and the forward kinematics solution of the arm is the product of these six matrices identified as ${}^0T_6$ (with respect to base) shown in Fig. 5.

The first three columns in the matrices represent the orientation of the end effectors, whereas the last column represents the position of the end effectors. The orientation and position of the end effectors can be calculated in terms of joint angles shown in (1) to (12).

$${}^0T_6 = A_1A_2A_3A_4A_5A_6 = \begin{bmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 5 Total Transformation Matrix

$$n_x = C_1(C_{234}C_5 - S_{234}S_5) \tag{1}$$
$$n_y = S_1(C_{234}C_5 - S_{234}S_5) \tag{2}$$
$$n_z = S_{234}C_5 + C_{234}C_6 \tag{3}$$

$$o_x = C_1(-C_{234})S_5 - S_1C_5 \tag{4}$$
$$o_y = S_1(-C_{234})S_5 + C_1C_5 \tag{5}$$
$$o_z = -S_{234}S_5 \tag{6}$$

$$a_x = -C_1((C_{23}S_4 + S_{23}C_4) \tag{7}$$
$$a_y = -S_1((C_{23}S_4 + S_{23}C_4) \tag{8}$$
$$a_z = -S_4(S_{23}) + C_4(C_{23}) \tag{9}$$

$$p_x = C_1(C_{234}a_4 + C_{23}a_3 + C_2a_2) \tag{10}$$
$$p_y = S_1(C_{234}a_4 + C_{23}a_3 + C_2a_2) \tag{11}$$
$$p_z = S_{234}a_4 + S_{23}a_3 + S_2a_2 \tag{12}$$

where ,

$C_i = \cos\theta_i$
$S_i = \sin\theta_i$
$S_1C_2 + C_1S_2 = S_{12}$
$C_1C_2 - S_1S_2 = C_{12}$.
$C_{234} = C_2[C_3 * C_4 - S_3 * S_4] - S_2[S_3 * C_4 + C_3 * S_4]$
$S_{234} = S2[C_3 * C_4 - S_3 * S_4] + C_2[S_3 * C_4 + C_3 * S_4]$

B. Inverse Kinematics

Inverse Kinematics analysis determines the joint angles for desired position and orientation in Cartesian space. This is more difficult problem than forward kinematics.

Total Transformation matrix in Fig. 5 will be used to calculate inverse kinematics equations. To determine the joint angles, ${}^0T_6$ matrix equation is multiplied by A_n^{-1} (n=1,2,3,4,5,6) on both sides sequentially and the generated linear equations were solved [7]. See Fig. 6. When a robot arm's degree of freedom (dof) is more than three (dof>3); on the contrary of forward kinematics, there exists multiple solutions in Inverse kinematics. Some constraints can be used to decrease the number of solutions for simplicity.

In this context, θ_{234} (WARTG=Wrist Angle Relative to Ground) is user supplied by entering both xyz position coordinates of the selected target position on the work space and a constant WARTG value [8].

$$A1^{-1} * \begin{bmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} = A1^{-1} * A1 * A2 * A3 * A4 * A5 * A6$$

Fig. 6. ${}^0T_6$ matrix multiplied by A_1^{-1}

The matrix manipulation in Fig. 6 has resulted the following matrix solution in Fig. 7.

$$\begin{bmatrix} n_xC_1+n_yS_1 & o_xC_1+o_yS_1 & a_xC_1+a_yS_1 & p_xC_1+p_yS_1 \\ n_z & o_z & a_z & p_z \\ n_xS_1-n_yC_1 & o_xS_1-o_yC_1 & a_xC_1+a_yS_1 & p_xS_1+p_yC_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} C_{234}C_5 & -C_{234}C_5 & C_{234} & C_{234}a_4+C_{23}a_3+C_2a_2 \\ S_{234}C_5 & -S_{234}C_5 & S_{234} & S_{234}a_4+S_{23}a_3+S_2a_2 \\ -S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 7 Multiplication solution

Both matrix elements in Fig. 7 are equated to each other and the resultant θ values are extracted. For example, equating [3, 4] elements of both matrices gave $pxS1 - pyC1$ = 0 and resulted in

$$\theta_1 = \arctan(p_y/p_x) \text{ or } \theta_1 = \theta_1 + 180° \quad (13)$$

Rearranging and squaring [1,4] and [2,4] elements of both matrices and adding them gives :

$$(p_xC_1 + p_yS_1 - C_{234}a_4)^2 = (C_{23}a_3 + C_2a_2)^2$$
$$(p_z - S_{234}a_4)^2 = (S_{23}a_3 + S_2a_2)^2 \quad (14)$$

and

$$(p_xC_1 + p_yS_1 - C_{234}a_4)^2 + (p_z - S_{234}a_4)^2 = a_2^2 + a_3^2 + 2*a_2*a_3 \quad (15)$$

From reference [7]:

$$C_3 = (p_xC_1 + p_yS_1 - C_{234}a_4)^2 + (p_z - S_{234}a_4)^2 - a_2^2 - a_3^2 / 2a_2a_3 \quad (16)$$

Hence, $\boldsymbol{\theta_3 = \arctan(S_3/C_3)}$ (17)

Referring Equations 14 and 15 and using,

$C_{12} = C_1C_2 - S_1S_2$ and $S_{12} = S_1C_2 - C_1S_2$

$$S_2 = (C_3a_3 + a2)(pz - S_{234}a_4) - S_3a_3(p_xC_1 + p_yS_1 - C_{234}a4) / (C_3a_3 + a_2)^2 + S_3^2a_3^2 \quad (18)$$

$$C_2 = (C_3a_3 + a_2)(p_xC_1 + p_yS_1 - C_{234}a_4) + S_3a_3(p_z - S_{234}a_4) / (C_3a_3 + a_2)^2 + S_3^2a_3^2 \quad (19)$$

θ_2 angle value becomes,

$$\theta_2 = \arctan(S_2/C_2) \quad (20)$$

and θ_4 is calculated as

$$\theta_4 = \theta_{234} - \theta_3 - \theta_4 \quad (21)$$

Similar multiplication procedures for joints 2, 3 and 4 also resulted in [7]:

$$A4^{-1}A3^{-1}A2^{-1}A1^{-1} * \begin{bmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} = A4^{-1}A3^{-1}A2^{-1}A1^{-1} * T_0^6$$

$$= A5 * A6 \quad (22)$$

Equating [2,1] and [2,2] elements of the resultant matrix equation (22) gives [7]:

$C_5 = (C_1n_y \ \ S_1n_x)$
$S_5 = (C_1o_y - S_1o_x)$ (23)

which results,

$$\theta_5 = \arctan(S_5/C_5) \quad (24)$$

The calculated θ_i values were later supplied in digital form to servo controller card from the computer. The card generated the relevant angular motion to every servo motor in the robot arm.

IV. Software

Software package L6KAP (Lynx-6 Kinematics Analysis Package) was developed to compute the forward and inverse kinematics of Lynx-6 Robot arm**.** Visual Studio.Net 2005 Development Platform with C# programming language was used for implementation. Graphical User Interface (GUI) of the software package was shown in Fig. 8.

The user can calculate both forward and inverse kinematics solutions of the robot arm based on the previous analysis. The equations 13, 17, 20, 21, 24 were used in the algorithms to generate the motions of the joints.

A flow chart of the software was given in Fig. 9. An On-line motional simulator of the robot arm was also included on the left side of GUI to show the generated motion based on the theoretical analysis presented here. Simulator simulates the robot arm motion for given angles or position values. When angle values are manually entered into transformation matrices and the Forward Kinematics button was pressed, Final Position Matrix ,${}^{0}T_{6}$, will be generated with new position and orientation angle values.

On the other hand, when the desired position values are entered into Final position matrix and the Inverse Kinematics Button was pressed, the required angle values are generated and shown in the angle text boxes.

V. Discussion

Mathematical modeling and kinematic analysis of a low Cost Robot arm, Lynx-6, was carried out in this study. Robot arm was mathematically modeled with Denavit Hartenberg (D-H) method. Forward and Inverse Kinematics solutions are generated and implemented by the developed software. An analysis technique was introduced to reduce the multiple solutions in inverse kinematics part. The developed software included a simulator part to test the motional kinematics and to show the relevant motion. A typical example, calculated with the generated software, was included here for the user. An initial position matrix is given in equation 25 with zero θ values and d, a and α values as in Table II. This matrix gives the initial position and orientation of the robot arm:

$$\text{Initial Pos:}\begin{bmatrix} 1 & 0 & 0 & 26.3140 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 19.9640 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (25)$$

TABLE II
D-H Parameters of Robot arm

joint	θ degree	d cm	a cm	α degree
1	45	0	0	90
2	30	0	12.065	0
3	30	0	12.065	0
4	-45	0	14.249	-90
5	30	0	0	0
6	0	0	0	0

When θ values are changed from zero to given values in Table II, the transformation matrices are generated as shown in equations (26) to (31):

$$A^{1}=\begin{bmatrix} 0.7071 & 0 & 0.7071 & 0 \\ 0.7071 & 0 & -0.7071 & 0 \\ 0 & 1 & 0 & 7.899 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (26)$$

$$A_2=\begin{bmatrix} -0.5 & -0.86 & 0 & -6.03 \\ -0.86 & -0.5 & 0 & 10.04 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

$$A_3=\begin{bmatrix} 0.5 & 0.86 & 0 & 6.03 \\ -0.86 & 0.5 & 0 & -10.04 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (28)$$

$$A_4=\begin{bmatrix} 0.70 & 0 & 0.70 & 10.07 \\ -0.70 & 0 & 0.70 & -10.07 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (29)$$

$$A_5=\begin{bmatrix} 0.86 & -0.5 & 0 & 0 \\ 0.5 & 0.86 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (30)$$

$$A_6=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (31)$$

The total transformation matrix, ${}^{0}T_{6}$, between the base of the robot arm and the end effectors is shown in equation 32.

$${}^{0}T6=\begin{bmatrix} 0.2380 & -0.9539 & -0.1830 & 9.7322 \\ 0.9451 & 0.2709 & -0.1830 & 9.7322 \\ 0.2241 & -0.1294 & 0.9659 & 32.4841 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (32)$$

${}^{0}T_{6}$ was determined by the developed software and it is the final *forward kinematics* solution of the robot arm. ${}^{0}T_{6}$ matrix values are checked against the physical positions of the robot arm in Table III.

TABLE III
DIFFERENCES BETWEEN CALCULATED AND PHYSICAL VALUES OF LYNX-6 ROBOT ARM

Position values	$^0T_6$ Values (cm)	Measured Values (cm)
Px	9,73	9.3
Py	9.73	9.27
Pz	32,48	31.85

When calculated xyz coordinates of the target position are compared with the measured coordinates as in Table III, it is observed that the values were very close to each other [9].

On the other hand, *inverse kinematics* equations will be used to determine the target position and its orientation for the robot arm.

The developed software will calculate the required angles for target orientation and target positioning.

For example, a WARTG value of 15^0, and the numerical values of $^0T_6$ are used in equation 32 and the angle values are calculated by using equations 13 to 24 as follows:

$\boldsymbol{\theta_1}$ ***= arc tan2 (9.73,9.73) = 45 degree***

$C_3 = ((271.62)^2 + (-19.50)^2) / 291.12 \rightarrow$

$C_3 = 0.86$ and $S_3 = 0.5$;

$\boldsymbol{\theta_3}$ ***= arc tan2 (0.5 / 0.86) = 30.1 degree***

$\boldsymbol{\theta_2}$ ***= arc tan2 ((371 – 99.4) / 371 + 99.4)) = 30 degree***

$\theta_4 = 15 - 30 - 30.1$

$\boldsymbol{\theta_4}$ ***= -44.9 degree***

$\boldsymbol{\theta_5}$ ***= arc tan2(0.49 , 0.86) = 29.99 degree***

VI. CONCLUSION

In this paper, Kinematics analyses of the Lynx-6 robot arm are investigated. A software package, *MSG,* was also developed to test and simulate the motional characteristics of this robot arm. *MSG* can be used as an educational tool in graduate and undergraduate robotic courses to realize the relationships between theoretical and practical aspects of various robot arm motions in real time.

REFERENCES

[1] Shadia Elgazzar, Efficient Kinematic Transformations for the Puma 560 Robot , IEEE Journal of Robotics and automation vol Ra-1, No.3, September 1985.
[2] Denavit J.R.S. Hartenberg, “A kinematic Notation for Lower- Pair Mechanism Based on Matrices,” ASME Journal of Applied Mechanics, June 1955 , 215 – 221.
[3] Rachid Mansuer , A software package for computer-aided robotics education.
[4] Introduction to Robotics Mechanics and Control, John J. Craig pp 109-114, 2005 Prentice Hall.
[5] Heterogeneous Modeling & Design of Robot Arm Control System**,** Antonia Yordan-Nones , University of Puertro Rico , Mayagüez.
[6] De Xu , Carlos A. Acosta Calderon ,Jong Q. Gan , Huosheng Hu , An Analysis of the Inverse Kinematics for a 5- Dof Manipulator, International Journal of Automation and Computing 2(2005) 114 – 124.
[7] Introduction to Robotics, Saeed B. Niku pp 67-76 ,2001 Prentice Hall.
[8] Inverse Kinematics for Lynx Arms by Laurent Gay (www.lynxmotion.com).
[9] Baki Koyuncu , Mehmet Güzel **,** Chessboard Application of 6 Axes Robot Arm by using Inverse Kinematics Equations , Journal of Computer Engineering, Vol. 1, No. 1, pg: 59 – 68, 2007.

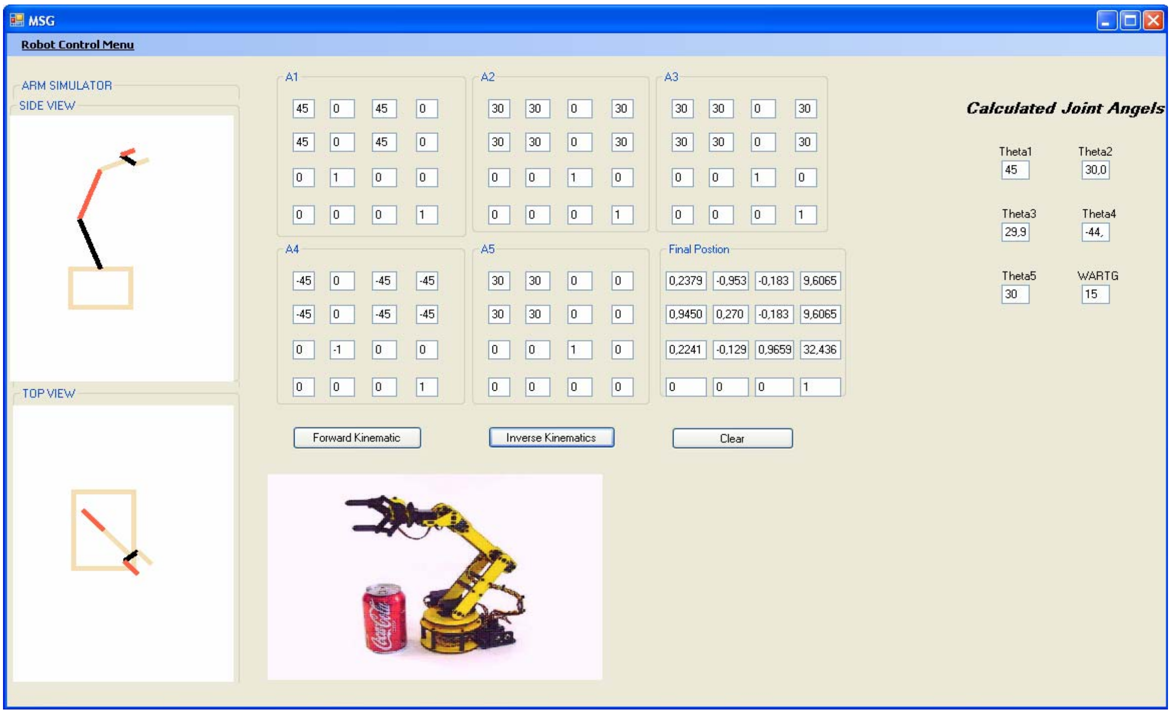

Fig. 8 GUI of L6KAP only base and shoulder angles are entered the others are kept at zero

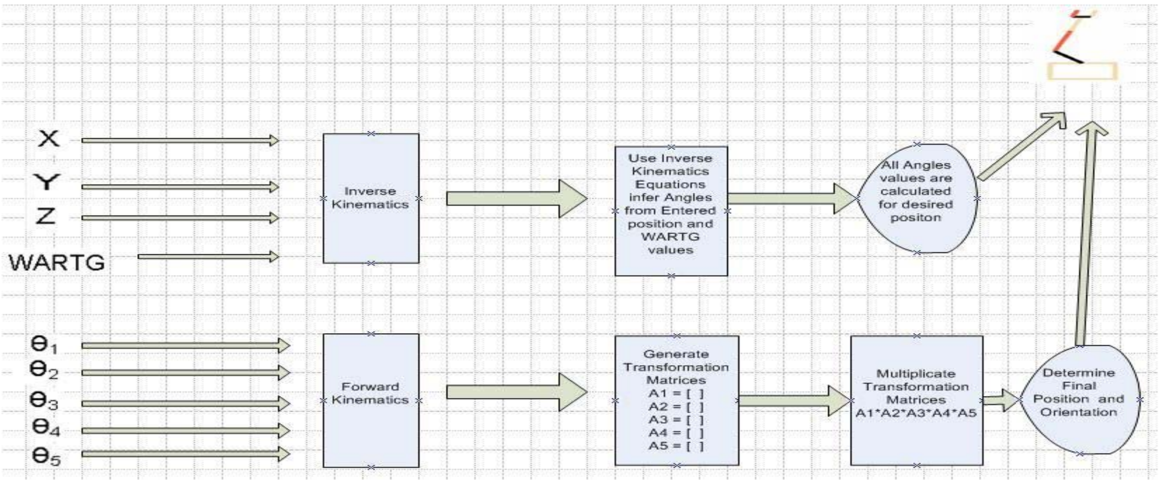

Fig. 9 A flow chart diagram of the software