AUTONOMOUS ROBOTIC ARM WITH ENHANCED INTELLIGENCE

[1]SANJU S, [2]NIPPUN KUMAAR A A, [3]SUDARSHAN TSB

[1,2,3]Amrita Vishwa Vidyapeetham-University; Department of Computer Science Engineering, Amrita School of Engineering, Bangalore Campus, India

Abstract– In robotics for accomplishing simple and complex tasks, manipulation is a very important aspect. For maximizing the robustness and minimizing the cost there is no ideal solution for manipulation. Service robot systems have long been a goal of robotics research. This proposed work deals with the design and development of an autonomous robotic arm with enhanced intelligence. Amrita Articulated Robotic Manipulator, AARM-II is designed and fabricated, and is used for the realization of the proposed algorithm. Servomotors are used as actuators. A computer through a serial port interface using Flash Magic terminal controls the arm. The arm will be capable of picking and placing the objects in the sequence given by the user through PC terminal; if the arm is ideal it enters into random pick and place mode. Object's source and destination order is entered in the user interface application. The trajectory planning is done with the help of Inverse Kinematics (IK). Geometric approach is followed to obtain inverse kinematic solutions. The work also aims at addressing the user interrupts instantly while the robot is in random pick and place mode and arrange the object in the sequence provided by the user in an efficient manner. The Microcontroller used is ARM7 based Blueboard LPC2148.

Keywords: User Interface, Pulse Width Modulation, Servomotor, Servo Gripper.

I. INTRODUCTION

Robots are proving to be powerful and efficient working force in modern days industrial environment to do mundane work. The motions of a robot are controlled through a controller. Robot manipulators were developed to enhance precision and repeatability of the work of a robot. These are also finding useful applications in education sector.

At present, the focus in robotics is towards service robots. The humanoid robots play a very important role in service robotics. The robotic arm is capable of handling more complicated task-level operations. Robot manipulator is developed to enhance human's work such as in the manufacturing or manipulation of heavy materials, working in hazardous and unpredictable environments.

The aim of this research work is adding intelligent to the robotic arm "Amrita Articulated Robotic Manipulator Version - II" (AARM-II). It is designed for 4 degree of freedom with end effector and has a hemispherical 3-D workspace. The intelligence for the robot includes addressing user interrupts instantly while the robot is in random pick and place mode and arranging the objects in a specified sequence by the user in more efficient manner. This article is organized as follows. Section II gives brief account of literature survey and related work on robot construction and operation. The hardware design and dimensions of AARM-II and its workspace is described in Section III. The kinematic analysis of AARM-II using geometric approach is explained in Section IV. The proposed algorithm is explained in Section V. The experiment and results are shown in Section VI.

II. LITERATURE SURVEY

A robotic arm reaching a particular position defined by a set of coordinates, pick up the object and place it at target location according to another set of coordinates, in a well defined hemispherical 3D space is described in the paper [1]. This robot arm in [1] has 5 degree of freedom and micro servomotors are used as principle actuators. The arms control is done by an interface application developed in Rapid Application Development (RAD) software, using micro-C programming language. The user in the interface application enters the object position and destination co-ordinates and the object is grabbed by using gripper. The onboard microcontroller used is PIC 16F88 (8 bit microcontroller), which generates a PWM signal and establish a communication between computer and hardware. Visual Basic 6.0 serves as GUI software and is used to communicate with the microcontroller from the computer.

In paper [2], YH050 a 7-axis robot manipulator developed by Hyundai Heavy Industries Co. is used. Three methods: Numerical approach, geometrical approach and combined approach are evaluated for finding optimum method, to solve Inverse Kinematics problem. Then motion planning is applied to avoid singularities, obstacles and to occupy minimal workspace for suitable application in an industrial environment. The results are obtained carrying out simulation using MATLAB and HR Space, a 3D simulator. From the simulation results, it is evident that numerical approach takes more time to solve inverse kinematics problem. In geometrical approach, the inverse kinematic solution has to be derived every time when the shape of the manipulator is changed. The combined approach produces a better performance for solving the inverse kinematic problem.

Motion planning for an anthropomorphic arm is discussed in paper [3]. A three-level motion planning framework joint space, movement primitive space, and task space is established by introducing movement primitives as the bridge connecting the task space and joint space. The proposed method can control the motion process of an anthropomorphic arm. It can also simplify the motion planning of complicated operation tasks. A specific human arm triangle model is proposed to describe the motion state of the anthropomorphic arm. The forward and inverse kinematics among joint space, human arm triangle space, and task space are derived by coordinate transformation and geometric analysis through introducing the concept of working plane. After this, the joint trajectories of two fundamental movement primitives based on the human arm triangle, including the motion of moving on the working plane and self-motion of switching working plane, are obtained by solving differential equations. Comparing simulation result with a traditional method and showing two real experiments, verify the validation and feasibility of the proposed method.

Kinematic model of a 6-axis robotic arm and its workspace analysis is presented in paper [5]. The robotic platform used is a six-axis robotic arm manipulator ED7220C that was developed by ED Corporation, Korea. The forward kinematic model is implemented using Denavit-Hartenberg (DH) method. DH method is used to develop the kinematic model of the robot in this work. With a specified desired position and orientation of the robot end-effector a realization of the inverse kinematics model will provide the required corresponding joint angles. The validation has been carried out using robotics toolbox of MATLAB.

The literature survey shows that giving intelligence to a robot required to do a task is important. Geometric method is simpler to implement to realize inverse kinematic model.

III. HARDWARE FABRICATION

AARM-II is fabricated using acrylic sheets with sizes 4mm, 6mm, 8mm and 10 mm. Acrylic sheet is chosen as it is lighter and gives good strength. The robotic arm has 4-axis excluding gripper assembly.

The principle actuators for this robotic arm are servomotors of various torques. There are four links and an end effector that is made of acrylic sheets.

For the movement of the robot arm, servomotors were used in this work. Even at low voltage ratings it offers high-speed operation and delivers excellent holding torque. Servomotors are DC motors that are having servomechanism and are used for accurate control of angular position. They have a small DC motor, a gear reduction drive for torque increase, an electronic shaft position sensing and a control circuit. In order to manipulate a servomotor to a desired position, pulse width modulated signals are used. A pulse of width varying from 1 to 2 milliseconds is sent to the servo at a frequency of 50 times per second. The angular position of the servomotor is determined by the width of the pulse.

Specifications for the servomotors and the number of motors used are given below:

VIGOR- VS-10A:
4 motors used - One at the base, one at the wrist, two at shoulder
Stall Torque: ≥ 7.5 kgf.cm ≥ 9.0kgf.cm
NEX - NRS 995:
1 motor used - One at fore arm
Stall Torque: 4kg.cm at 6V
NEX - NRS 585:
1 motor used - At end effector
Dual bearing with metal gear
Stall Torque: 15.5kg.cm at 4.8V, 17kg.cm at 6V

To make the whole design optimized to work with a high degree of accuracy, care is taken to minimize size and weight of the hardware. Servomotor brackets are used to hold these servomotors as shown in Fig.3.1.

Fig. 3.1 Complete servo motor bracket assembly with servomotor

Gripper used is a parallel gripper assembly manufactured by NEX ROBOTICS and is implemented with aluminum alloy. A small gear arrangement is present in the gripping action. The gripper assembly part is shown in the Fig. 3.2.

The complete arm design is divided into 4 parts i.e. the base, the lower arm, the upper arm and the gripper. The servo gripper is attached at the end of the upper arm.

All the servomotor brackets are joined together along with the base and gripper assembly, as shown in Fig.3.3, to form the whole assembly of the robotic arm.

In addition, a Motor control PCB is designed (in PCB board) to interconnect ARM7 based Blueboard LPC 2148 with the servomotors and the DC power supply. The Motor control PCB is as shown in Fig. 3.4.

Fig. 3.2. Gripper assembly

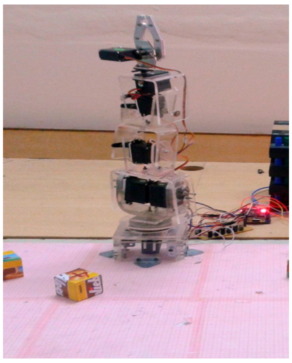

Fig. 3.3 Complete robot arm structure of AARM-II

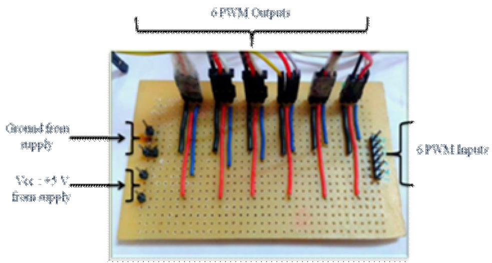

Fig. 3.4. Motor control PCB

Structural design of manipulator depicting the servomotor arrangement is as shown in Fig. 3.5.
Servo 1 is at base
Servo 2 and 3 is at link 1 (L1)
Servo 4 is at link 2 (L2)
Servo 5 is at link 3 (L3)
Servo 6 is at motor is at the end effector
Where L0 = 15.5 cm, L1 = 12.5 cm, L2 = 14.1 cm, L3 = 16 cm.

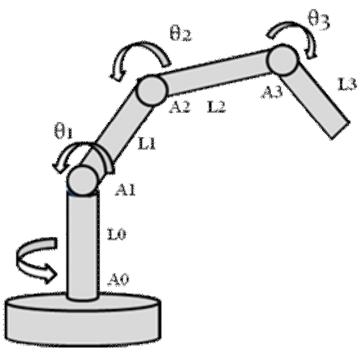

Fig. 3.5. Structural figure of Manipulator

Dimensions:

Total Length	:	58.1cm
Width Max	:	8.5cm
Breadth	:	4cm
Base	:	10cm x 10cm x 8cm
End effector length	:	16cm

The workspace used is hemispherical as shown in Fig. 3.6.

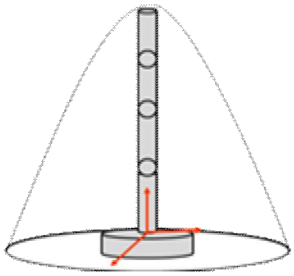

Fig 3.6. Workspace

VI. KINEMATIC ANALYSIS

The study of kinematic behaviour is involved in modelling a robot. A kinematic model is concerned with the robot's motion. It does not consider the forces producing the motions. The kinematics of a robotic arm deals with the study of geometric and time based properties of the motion and in particular the motion of various links of a robot with respect to each other and with respect to the time. It provides a relationship between the position and orientation of robot end effector and its joint variables. For deriving inverse kinematics solution there are two methods; geometric method and an algebraic method. The problems in Inverse Kinematics are: there may be multiple solutions for some situations; no solutions and redundancy problem. Deriving equations for large number of unknowns is difficult and time consuming. So the geometric approach is used in this work for implementing the proposed algorithm for robot arm with enhanced intelligence.

Geometric Approach:
Consider a robotic arm with 4-axis as shown in Fig. 4.1. By using IK-Cartesian method, the user specifies the desired target position of the gripper in Cartesian space as (x, y, z). Here z is the height.

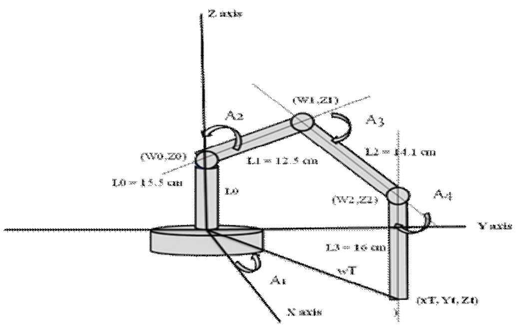

Fig. 4.1. The robot arm structure for Geometric approach calculation

The robot arm is having 4-axis: Base Rotation, Shoulder, Elbow and Wrist. All degrees of freedom of this arm act in the same vertical plane. The vertical plane can rotate around the z-axis and all the links will be in this plane. The direction and rotational angle of the base depends on the plane. Simple trigonometric relations can be used to obtain it.

The base rotation angle (A0) is a simple arcTan function of the desired actuator target xT and yT coordinates.

$$A0 = \arctan\left(\frac{yT}{xT}\right) \qquad (1)$$

Here there are no degrees of freedom that go out of the vertical plane. Once A0 is calculated, the value of wT can be obtained from the Fig. 4.1 using simple Pythagoras Theorem as shown below.

$$wT = xT^2 + yT^2 \qquad (2)$$

wT will enable to obtain required angles for the two dimension plane wz. In most of the scenarios, gripper is considered to act on a certain fixed angle. Knowing the target gripper coordinates, wT and zT, and the desired angle of the actuator link (AG=Gripper Angle), coordinates w2 and z2 are calculated.

$$w2 = wT - L3 * \cos(AG) \qquad (3)$$

$$z2 = zT - L3 * \sin(AG) \qquad (4)$$

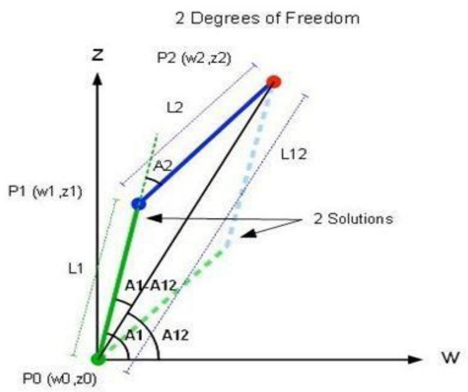

Fig. 4.2 Angle calculations using cosine

Knowing the coordinates of Link 1 & 2, the angles A1, A2 and A3 are calculated.

$$L12 = w2^2 + z2^2 \qquad (5)$$

$$A12 = \arctan\left(\frac{z2}{w2}\right) \qquad (6)$$

If L12 > L1 + L2, then there is no solution as arm cannot reach the target.

From the Law of Cosines

$$L22 = L12^2 + L1^2 - 2 * L1 * L12 * \cos(A1) - A12 \qquad (7)$$

$$A1 = \arccos(L12) + L12^2 - L22^2 * L1 * L2 + A12 \qquad (8)$$

Using A1,

$$w1 = L1 * \cos(A1) \qquad (9)$$

$$z1 = L1 * \sin(A1) \qquad (10)$$

$$A2 = \arctan\left(\frac{z2 - z1}{w2 - w1}\right) - A1 \qquad (11)$$

Having A1 and A2, calculate the angle of Link 3,

$$A3 = AG - A1 - A2 \qquad (12)$$

Thus the angles for all the links are obtained for the specified coordinates.

V. PROPOSED ALGORITHM

Addressing user interrupts instantly while the robot is in random mode:

The controller gets input from the terminal through the UART port. i.e., receiving input from the computer terminal. The algorithm is based on addressing user interrupts instantly while the robot is in random pick and place mode. The proposed algorithm comprises of three modes: Pick and Place mode, Random mode and Interrupt mode.

The flowchart of the proposed algorithm is as shown in Fig. 5.1. In pick and place mode the objects are picked from predefined source and according to user input they are placed in the destination positions. For every object inverse kinematics are used to find the joint angle between the source and destination and corresponding PWM signals are generated to actuate the servomotors. After completing the pick and place mode, the program enters a random mode. AARM-II continues working in random mode until an interrupt signal is asserted. Random array is generated in the program and the user input during an ongoing operation acts as an interrupt. When an interrupt is asserted; the Interrupt mode gets activated. According to the user input, the Update table having the source and destination location is checked and accordingly the object is placed. At each stage of pick and place operation the Update table is refreshed. After Interrupt mode, the random mode will be activated again.

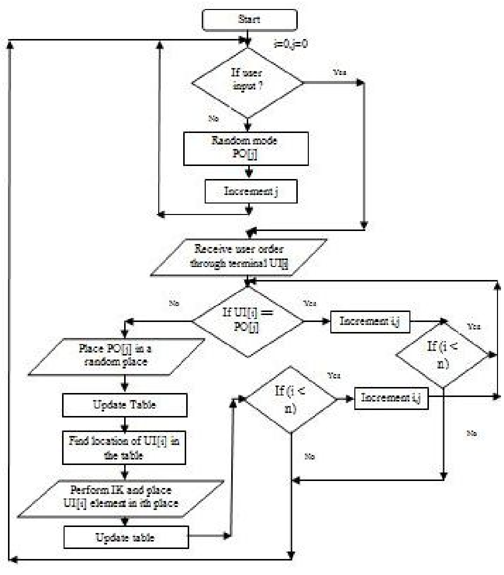

Fig. 5.1 Flowchart for Proposed Algorithm

Also when user input is given after interrupt occurs; the program validates the destination. The destination and source order are verified and if found to be identical then no operations are carried out. If both orders are different then pick and place is carried out. Thus it reduces the time taken to execute.

VI. EXPERIMENT AND RESULTS

ARM7 based Blueboard LPC 2148 is used for the implementation. The compiler used in this work is Keil using Embedded C program. The embedded C program is burned through serial port using Flash Magic and user input is passed through Flash Magic Terminal serial port.

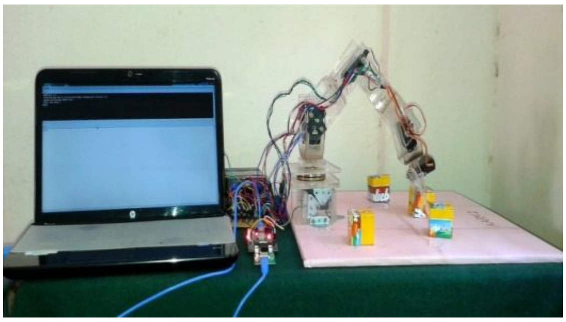

Fig. 6.1 Implementation setup

A) Implementation Setup:
The implementation setup is as shown in the Fig.6.1.

B) Implementation Result:
Orders generated at each iteration, till 2nd random order generation, is implemented on AARM-II shown in TABLE I:

TABLE I
Orders generated at each iteration

	Pick and place mode		Random mode	Interrupt mode	Random mode
	Default order	User input order	1st Random order	User Input in interrupt mode	2nd Random order
	1	5	1	5	4
	2	2	5	4	5
	3	3	4	2	3
	4	1	3	3	2
	5	4	2	1	1
Sequence:	I	II	III	IV	V

Sequence I and II: Pick and place mode:
The implementation results in pick and place mode as shown in Fig. 6.2. Initially the objects 1, 2, 3, 4, 5 are kept at locations (22, 0), (14, 20), (20, -20), (0, -20) and (5, 20) respectively; The objects are picked and are placed at locations (10, -20), (22, -10), (20, 20), (22, 0) and (0,-21) as per the specified input. If the object is picked from a location, then that location will be updated with value 0.

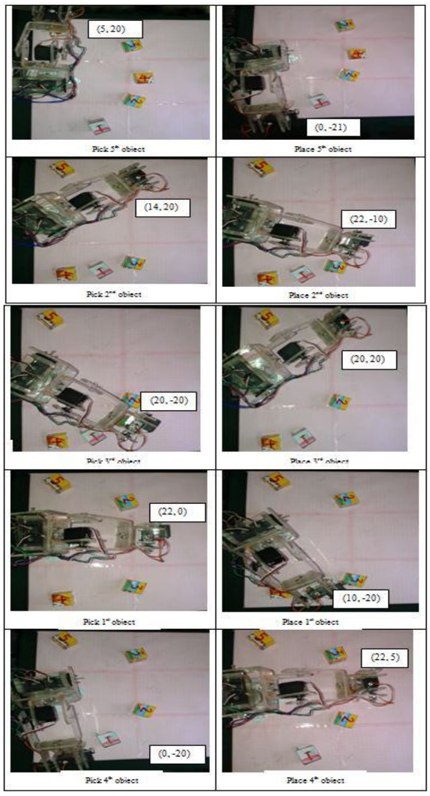

Fig. 6.2. Pick and Place mode

Update Table:

Source	Destination
1	5
2	2
3	3
4	1
5	4

Sequence III: Random mode:
In this mode manipulator generates a random number between 1 and 5 and picks the source and places it at the destination.
Update Table:

Source	Destination
5	1
2	5
3	4
1	3
4	2

Assume that after random order is generated and 1, 5, 4 is placed at destination, an interrupt is generated. Then Source and Destination will be updated as shown below:
Update Table:

Source	Destination
0	1
2	5
3	4
0	0
0	0

Sequence IV: Interrupt mode:
After the interrupt activated, the proposed algorithm will check the source and destination locations. The table is being refreshed and will place the object as per the user command.

Update Table:

Source Destination

Previous Update Table

0	1	5
2	5	2
3	4	1
0	0	3
0	0	4

For example, when the AARM-II picks an object 1 from destination of Previous Update Table and tries to place it at the destination of refreshed Update table; the following operations are carried out in sequence:

i. Before picking 1 check whether any objects are kept at destination 3.
ii. Here object 3 in the source position of Previous Update Table is present at 1's destination, pick object 3 and place it at location of Destination where there is no object placed.
iii. Then pick 1 and place at its original destination.
iv. Repeat the above steps.

Sequence V: Random mode:
After interrupt mode, Random mode will continue. The next random sequence 4, 5, 3, 2, 1 is generated and refreshed Update Table as shown below. The random mode continues until next interrupt is generated.

Update Table:

Source	Destination
5	4
2	5
1	3
3	2
4	1

CONCLUSION

Based on the literature survey carried out, parameters have been chosen. They are; use of geometric analysis for obtaining angle equation instead of D-H method because of its simple approach. Servomotor is used for movement of AARM-II and Flash Magic Terminal is used as an interface for the user to provide input. Algorithm for continued random mode operation of the arm is implemented on the robotic arm (AARM-II). Adding user interface and intelligence to the robot for picking and placing of objects is done by addressing user interrupts instantly while operating in random pick and place mode. If the arm is ideal it should enter into random pick and place mode. Each experiment was implemented and the results are verified.

As future work, another 4-axis articulated robotic arm has to be fabricated. The second arm has to be coupled with AARM-II. The efficient algorithm for actuating two manipulators to perform a single task has to be developed.

REFERENCES

[1] Himanshu K. Patel, Prabuddh Verma, Shreya Ranka. "Design and Development of Co-ordinate Based Autonomous Robotic Arm", International Conference on Current Trends in Technology, NUiCONE – 2011, Institute of Technology, Nirma University, Ahmedabad. Page(s): 1 – 6, 8-10, Dec. 2011.

[2] Jayoung Kim, Minyoung Sin and Jihong Lee. "Kinematics Analysis and Motion Planning For a 7-DOF Redundant Industrial Robot Manipulator", 2011 11th International Conference on Control, Automation and Systems (ICCAS), in KINTEX, Gyeonggi-do, Korea 26-29, Page(s): 522 – 527, Oct. 2011.

[3] Xilun Ding and Cheng Fang "A Novel Method of Motion Planning for an Anthropomorphic Arm Based on Movement Primitives", IEEE/ASM transactions on mechatronics, Vol.18, No.2, May 2012.

[4] Servobotics RA-02 Robotic Arm & Controller, Manual & User's Guide. Images SI Inc. http://www.imagesco.com

[5] Jamshed Iqbal, Raza ul Islam, and Hamza Khan. "Modelling and Analysis of a 6 DOF Robotic Arm Manipulator", Canadian Journal on Electrical and Electronics Engineering Vol. 3, No. 6, Page(s): 300-306, July 2012.

[6] Mohammed Abu Qassem, Iyad Abuhadrous, Hatem Elaydi. "Modeling and Simulation of DOF Educational Robot Arm" Advanced Computer Control (ICACC), 2nd International Conference, vol.5, page(s):569-574, 2010.

[7] LPC214x User Manual, UM10139 Volume 1:, Rev.01—15, August 2005.

★★★