Design and Development of an Educational 5-DoF Robotic Arm

A.O. Oluwajobi* and A.A. Oridate

Department of Mechanical Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria

Abstract: The effective training of engineering students require hands-on experience in what they would later work with in the industry. The trend in manufacturing nowadays is towards robotics and automation. Relevant to this technology is the robotic hand. This paper outlines the design and the development of a 5-DoF (Degree of Freedom) robotic arm. The robotic arm is intended for educational purposes. The design proposed the use of servos to power the joints and to implement the inverse kinematics of the robotic arm. A simulation of the robotic arm was achieved by using the Matlab Robotics Toolbox, to visualise the joint movements. A suitable servo controller was selected for the implementation and a control software for the robotic arm was developed using Microsoft's C# programming language. The software allows the robotic arm gripper to be postioned in space, by specifying the coordinates of its centre position. Polymethyl methacrylate was selected to fabricate the components of the robotic arm. The robotic arm was tested by specifying various coordinates for the gripper with reference to the robot's base and measuring the corresponding coordinates of the centre position of the gripper, which gave satisfactory results.

Keywords: Forward and Inverse Kinematics, Denavit-Hartenberg Convention, 5 DoF, Educational Robotic Arm.

1. INTRODUCTION

The use of robots has led to increased productivity and improved product quality in manufacturing. It has helped to keep manufacturing industries viable in high-labour cost countries. Many products in the market place today, have been assembled or handled by a robot [1]. Tracing the history of robots; in 1969, Victor Scheinman at Stanford University invented the Stanford arm, an all- electric, 6-axis articulated robotic arm. The robot could accurately follow arbitrary paths in space and could be employed in assembly and arc welding [2]. In 1973, the wide range of technological experience gained by KUKA, combined with demands from the automotive industry for powerful and reliable robots culminated in the development of KUKA's own industrial robot. Known as FAMULUS, it was the first robotic arm to have six electric motor-driven axes [3]. The PUMA on the other hand was a very versatile pick-and-place industrial robotic arm that could perform a variety of tasks in assembly, material handling and transfer, joining, inspection, and palletizing [4]. Chen *et al.* (2006) [5] of the Department of Electrical Engineering, National Chung-Hsing University designed and implemented an Omnidirectional Mobile Home Care Robot for the Nios II Embedded Processor Design Contest. The project involved the design of a smart robot to expand the applications for which robotic arms were traditionally used. The robotic arm was installed on an omnidirectional mobile vehicle and a network camera was installed on the front of the arm. Jenne (2009) [6] of the Swiss Federal Institute of Technology, Zurich designed and implemented an adaptive motion control for a complex robot system, consisting of a mobile platform, realized with a differential drive and a 6-DoF robotic arm. Firstly, the kinematic models of the system were analyzed separately. With the theory of the homogeneous matrix exponential method, a combination of the kinematic models was established. Then, the motion of the mobile platform was realized with a kinematic position controller.

Blackmore *et al*. (2009) [7] from Portland State University worked on a mobile robot capable of human-like behaviours. The robot had a mobile base; it had a head and neck capable of producing several human-like gestures coordinated with speech; and it also had an arm and hand assembly. Adhikari *et al.* (2010) [8] of the Worcester Polytechnic Institute, USA, designed, simulated and implemented a mobile robotic base with an attached robotic arm. The aim of the project was to create a robot capable of assisting elderly people with tasks in their everyday lives. Akinwale (2011) [9] of Obafemi Awolowo University developed a robust remote laboratory platform for conducting robotic arm experiments. The online laboratory platform was implemented using the Massachusetts Institute of Technology (MIT) interactive iLab architecture. Error-checking algorithms were implemented to ensure laboratory safety without human supervision.

Adebola (2012) [10] of Obafemi Awolowo University designed an improvised robotic arm system for educational instructional purposes. The system was implemented using improvised materials, while the software was developed using the C# programming language. Herman *et al*. (2012) [11], of the University

Address correspondence to these authors at the Department of Mechanical Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria;
E-mail: j.oluwajobi@gmail.com, aogunjob@oauife.edu.ng

E-ISSN: 2409-9694/19

of Zaragoza (Spain), designed and built a low-cost mobile platform equipped with a robotic arm. The robotic arm was built with off-the-shelf parts and controlled with an Arduino board. Ishola (2012) [12] of Obafemi Awolowo University developed a remote online Strength of Materials laboratory, a platform for performing static bending experiments over the internet. A robotic arm with four degrees of freedom (4-DoF) was designed and implemented. The joints of the arm were powered by AX-12™ dynamixel™ servos manufactured by Robotis®. A continuous rotation servo was used to effect a 360° rotation at the base joint. The end-effector was a gripper used for picking the beams. The control of the servos from a PC was achieved using the USB2Dynamixel™ - a TTL (Transistor–transistor logic) level converter. The USB2Dynamixel is basically a "Virtual Serial Port" that attaches to the USB port on a PC. It was used as the connector between the dynamixel servos and the computer, thus allowing the servos to receive instructions from the PC. The software interface for communicating with the robotic arm was developed using Microsoft's C# programming language. The current 21st century engineering students require hands on experience on working robotic systems, to adequately prepare for current and future industrial challenges; hence this study.

2. ROBOTIC DESCRIPTION AND MECHANICAL DESIGN

The robotic arm has five degrees of freedom (5-DOF) with the inclusion of a gripper. A CAD model of the robotic arm structure was designed and rendered using Autodesk's AutoCAD® software. The entire implementation plan of the robotic arm is represented by a block flow diagram shown in Figure **1**, while Figure **2** shows a schematic diagram of the robotic arm setup.

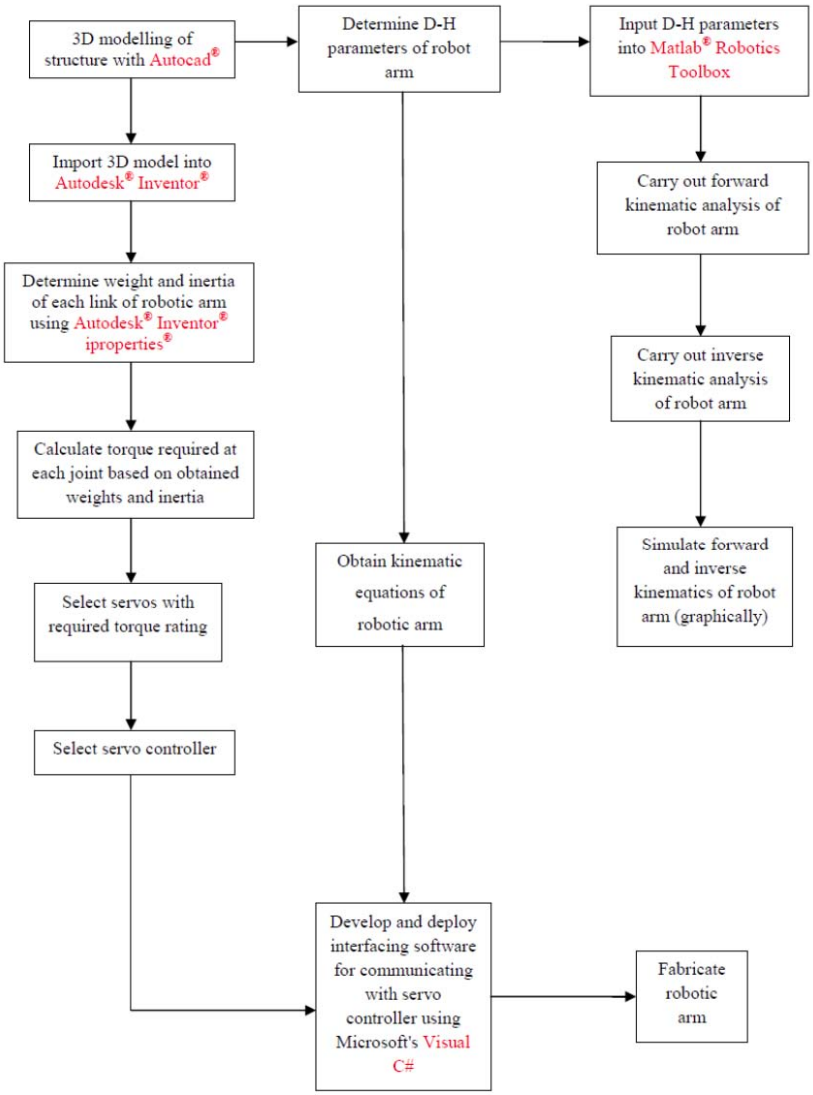

Figure 1: The Implementation Plan of the Robotic Arm.

In order to compute the forward kinematics equation for an open-chain manipulator (robotic arm), a systematic, general method is needed to define the relative position and orientation of two consecutive links. However, it is convenient to set some rules for the definition of the link frames [13]. The D-H convention is a representation technique used to achieve this [14]. The modified D-H representation was used for various analyses in the design process of the robotic arm [1].

2.1. Kinematic Analysis using Denavit-Hartenberg Convention

The Denavit-Hartenberg (D-H) convention was used to compute the forward kinematics of the robotic arm. Based on the procedure for determining the Denavit-Hartenberg parameters of a serial link, the D-H parameters of the robot were determined. A schematic diagram of the robotic arm showing the chosen joint axes is shown in Figure **3**. The D-H parameters of the robotic arm (shown in Table **1**) were obtained from the schematic diagram.

Figure 2: The CAD model of the Proposed Robotic Arm.

The location of the central point of the end effector (gripper) of the robotic arm is given as (x, y, z), where:

$$x = \cos\theta_1 [L_4 \cos(\theta_2 + \theta_3) - L_3 \sin(\theta_2 + \theta_3) - L_2 \sin\theta_2] \quad (1)$$

$$y = \sin\theta_1 [L_4 \cos(\theta_2 + \theta_3) - L_3 \sin(\theta_2 + \theta_3) - L_2 \sin\theta_2] \quad (2)$$

$$z = L_4 \sin(\theta_2 + \theta_3) + L_3 \cos(\theta_2 + \theta_3) + L_2 \cos\theta_2 \quad (3)$$

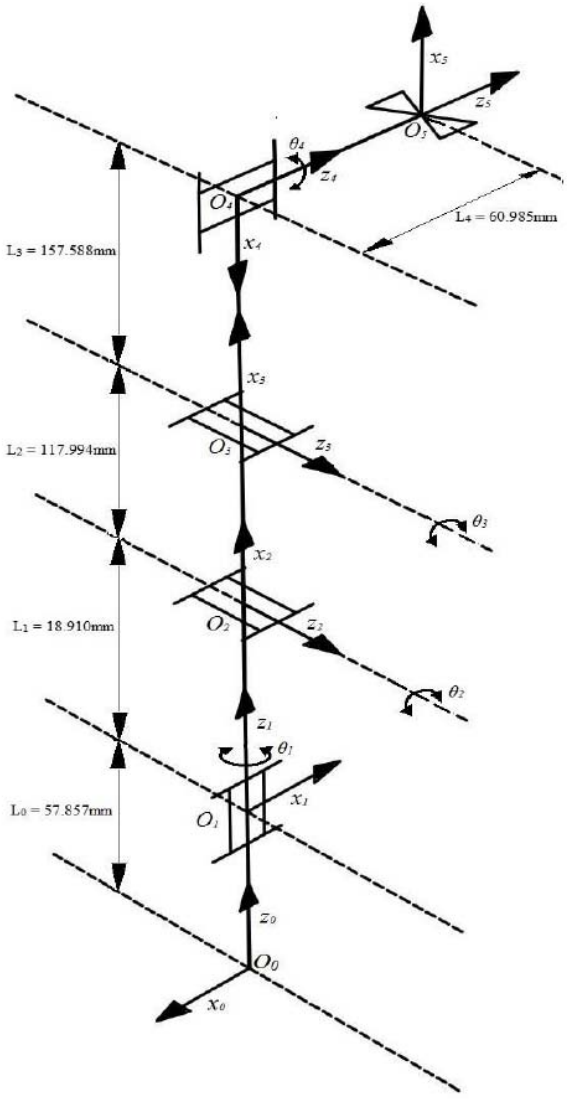

Figure 3: Schematic of robotic arm showing chosen joint axes.

Note: O_i is the origin of the *i*th frame.

Table 1: D-H Parameters for the Robotic Arm

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0°	0	0	θ_1
2	90°	0	0	90°- θ_2
3	0°	L_2	0	θ_3
4	90°	L_3	0	180° - θ_4
5	0°	0	0	180°

The L_i are the links and the θ_i are the joint angles.

The above equations represent the forward kinematics of the robotic arm - given the values of the joint angles; they can be used to obtain the location of

the end effector. The inverse kinematics problem would be to obtain the joint angles, given the position and the orientation of the end-effector. A feedback algorithm was implemented to check if the solution would successfully place the robotic arm at the desired location after finding the inverse kinematics of the robotic arm. The feedback algorithm runs the obtained inverse kinematics through the forward kinematics to obtain a location in space after which the location is compared against the target location specified. The first solution to yield the correct location in space is chosen.

2.2. Singularities and the Workspace of the Robotic Arm

Singularities occur in robotic arms, when two or more links are lined up or when two or more joints have reached their geometric limits. At these points the robotic arm would not be able to move the end effector in some directions, no matter how it moves its joints. In practice, at the vicinity of singularity configurations, the robotic arm would have unpredictable actions and velocities. The singularity points can be calculated by equating the determinant of the Jacobian matrix to zero or by searching for the row rank deficiency conditions of the Jacobian matrix. In practice, the singularity points should be avoided. Closely related to the singularities is the robotic arm workspace. It is all the reachable points of the robotic arm end effector [15, 16].

2.3. Kinematic Analysis and Simulation using MATLAB®

MATLAB® was used to graphically illustrate the kinematic analysis of the robotic arm. The toolbox is based on a very general method of representing the kinematics and dynamics of serial-link manipulators, known as the Denavit-Hartenberg (D-H) notation [17].

The D-H parameters (given in Table **1**) were used to create a vector of 'Link' objects as shown below:

```
>> %        theta      d        a          alpha      joint type     D-H type
L(1) = Link ( [0        0        0          0          0          ],    'modified');
L(2) = Link ( [(pi/2)   0        0          pi/2       0          ],    'modified');
L(3) = Link ( [0        0        0.117994   0          0          ],    'modified');
L(4) = Link ( [pi       0        0.157588   pi/2       0          ],    'modified');
L(5) = Link ( [pi       0        0          0          0          ],    'modified');
```

The joint angles *alpha* are variables since all the joints are revolute. The zeros only act as placeholders for the joint angle variables. This provides a concise description of the robot. The robot has 5 revolute joints denoted by the structure string 'RRRRR'. It is defined in terms of modified Denavit-Hartenberg parameters and gravity acts in the default z-direction. By invoking the plot method, the robot could be visualized graphically as stick figures. The graphical representation included the robot's name, the joints and their axes, and a shadow on the ground plane. MATLAB returned a graphical representation of the robot pose corresponding to each set of joint angles. Also**,** the effect of changing the set of joint angles on the pose of the robotic arm was simulated using the "teach" method in the Robotics Toolbox. The teach method is a useful tool in the MATLAB® Robotics Toolbox for simulating robots. It is used to drive (the joints of) a graphical robot by means of a graphical slider panel. The graphical slider panel provides a means of changing the set of joint angles of the robotic arm [1]. Figure **4** shows a screenshot of the teach method in use. Each pose of the robotic arm, positions the frame of the end effector at a point in space that can be represented by a transformation matrix. The goal of the forward kinematic analysis of the arm using the Robotics toolbox is to obtain this transformation matrix for any set of joint values. The transformation matrix is a representation of the position of the end effector of the arm in space relative to a reference frame. The "fkine" method was used to illustrate this.

2.4. Determination of Torque Requirement of Joints

To effect motion of the links of a robotic arm, an actuator is fixed at each joint of the arm. The actuator applies torque at the joint to overcome the resistance of the link to motion. Resistance of any link to motion is due to gravity and inertia effects. The effect of gravity on any link of the robot is to pull and accelerate it towards the centre of the earth due to its own weight, thus exerting a resistive torque on it. Therefore, a portion of the torque generated by the actuator is needed to overcome this resistive torque (due to gravity). It was necessary to calculate the value of the gravity-induced resistive torque acting on each link of the arm, so that an actuator with sufficient torque rating could be selected for each joint.

The resistive torque exerted on a joint due to gravity acting on the robot depends very strongly on the robot's pose. Intuitively the torque on the shoulder joint is much greater when the arm is stretched out horizontally. So, in order to calculate the torque required at each joint, the worst case scenario (arm completely stretched out) was chosen. The arm is

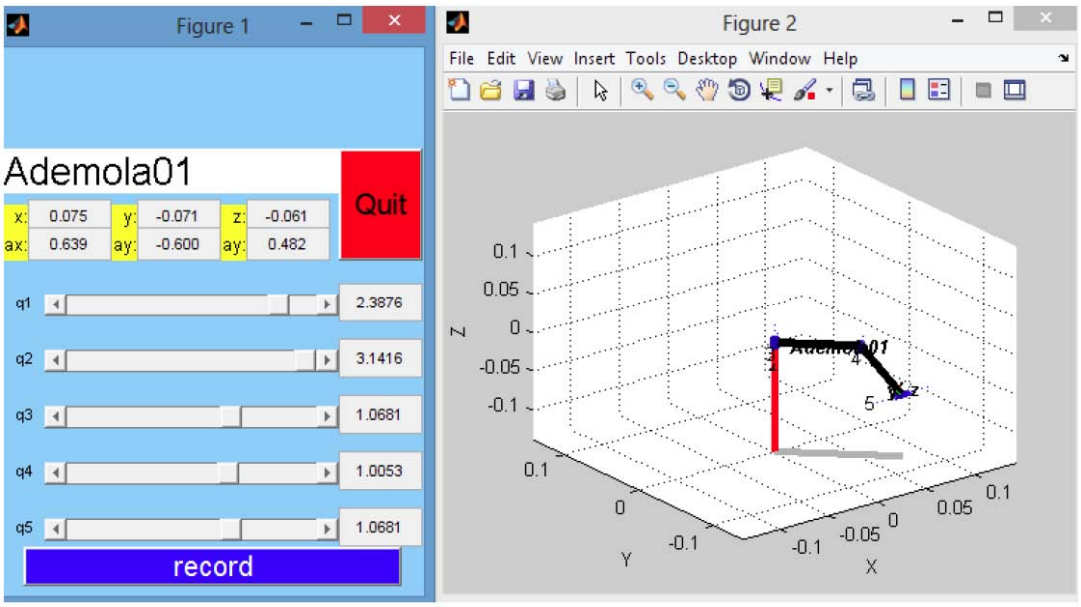

Figure 4: Illustration of the use of teach method for simulating a robotic arm.

subjected to the highest torque when the arm is stretched horizontally [1]. Figure **5** shows a labelling of the lengths of the robot arm links, while a free-body-diagram (FBD) of the stretched-out arm is shown in Figure **6**.

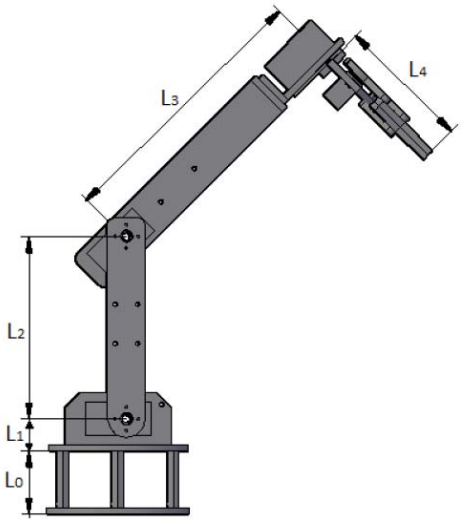

Figure 5: Labelling of lengths of the robotic arm links.

The following expressions hold, namely;

T_{1g} = 0 N.m - the waist rotation does not cause motion of any link in the vertical

plane (i.e. against gravity). (4)

$$T_{2g} = W_2\left(\frac{L_2}{2}\right) + W_{j3}L_2 + W_3\left(L_2 + \frac{L_3}{2}\right) + (W_{j4} + W_4 + W_{j5} + W_{gripper} + W_{payload})(L_2 + L_3) \quad (5)$$

$$T_{3g} = W_3 + \left(\frac{L_3}{2}\right)(W_{j4} + W_4 + W_{j5} + W_{gripper} + W_{payload})(L_3) \quad (6)$$

T_{4g} = 0 N-m - wrist rotation does not result in motion against gravity (7)

T_{5g} = 0 N-m - opening and closing of gripper jaws does not result in motion against gravity (8)

Where:

W_1 - Weight of link L_1

W_2 - Weight of link L_2

W_3 - Weight of link L_3

W_4 - Weight of link L_4

$W_{gripper}$ - Weight of gripper

$W_{payload}$ - Weight of payload

W_{j2} - Weight of joint 2

W_{j3} - Weight of joint 3

W_{j4} - Weight of joint 4

W_{j5} - Weight of joint 5

T_{1g} - Resistive torque at joint 1 due to gravity

T_{2g} - Resistive torque at joint 2 due to gravity

T_{3g} - Resistive torque at joint 3 due to gravity

T_{4g} - Resistive torque at joint 4 due to gravity

T_{5g} - Resistive torque at joint 5 (gripper joint) due to gravity

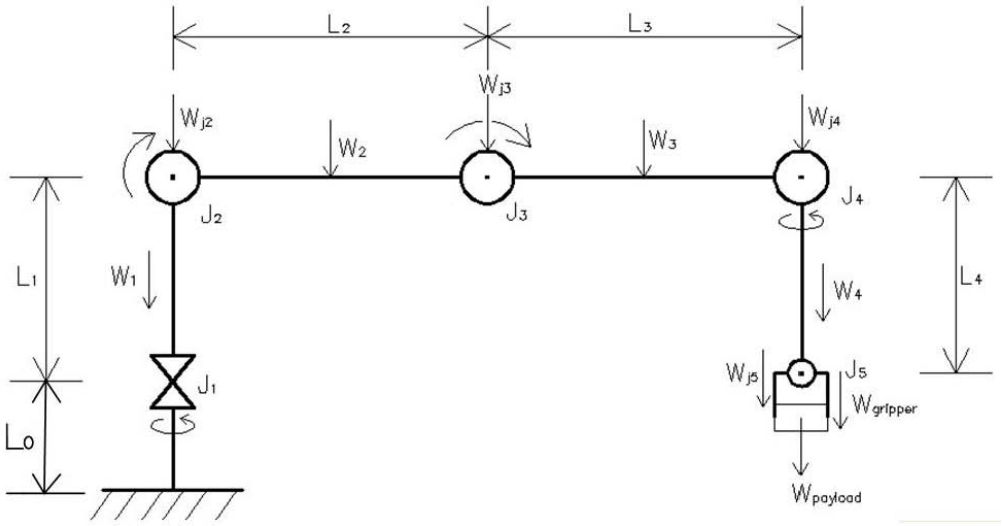

Figure 6: Free-body-diagram of robot arm in stretched out pose.

The resistance that is shown by a body to change its rotation is called moment of inertia (I). Thus, the difficulty faced to change the angular motion of a body about an axis is measured by calculating its moment of inertia about that particular axis. The moment of inertia plays the same role for rotation as the mass does for a translational motion. It describes the angular acceleration produced by an applied torque [18]. It follows that, for a given angular acceleration of any link of the robot, the inertia of the link results in a resistive torque applied at the joint where the lower end of the link is connected. The value of this torque (T_i) is given by the product of the moment of inertia of the link and it's angular acceleration, α [18].

$$T_i = I\alpha \quad (9)$$

According to Benenson *et al*. (2006) [19], the moment of inertia of a body depends on its shape and mass distribution as well as the orientation of its rotational axis, so it was necessary to use a different formula to calculate the moment of inertia for each link of the robotic arm. The moments of inertia was determined by using a Modelling software. The 3-D model of the robotic arm (created with AutoCAD®) was exported into Autodesk® Inventor® Professional. The moments of inertia of the various parts of the robot were determined by using the "iproperties" feature of Autodesk® Inventor® Professional, viz;

I_{XX} = 0.164 lb_m -in^2 = 0.164 x 2926.4g-cm^2 = 479.929g.cm^2

I_{YY} = 0.226 lb_m -in^2 = 0.226 x 2926.4g-cm^2 = 661.366g.cm^2

I_{ZZ} = 0.072 lb_m -in^2 = 0.072 x 2926.4g-cm^2 = 210.701g.cm^2

The robot was designed such that each motor rotates by one degree step every 20 milliseconds, which is equivalent to an angular velocity of 50 degrees/second or 0.873rad/s. Although, the motors are capable of rotating at a higher speed, the moderate speed selected will reduce the rate of wear and tear on the robotic arm joints. Also, each link moves from the rest position (0 rad/s) to the designed angular velocity of 0.873rad/s in 0.5 seconds, which translates into an angular acceleration of 1.746 rad/s^2. For the arm to move from a rest position, this acceleration is required. In the case of a robotic arm, the moment of inertia must take into consideration that the part is being rotated about a pivot point (the joint in this case) located a distance away from the centre of mass. To evaluate the moments of inertia about the joint axes, the parallel axis theorem was applied. The parallel axis theorem states that if the body is made to rotate instead about a new axis *z'* which is parallel to the first axis and displaced from it by a distance *h*, then the moment of inertia *I* with respect to the new axis is related to I_{cm} by;

$$I = I_{cm} + mh^2 \quad (10)$$

where *h* is the perpendicular distance between the axes *z* and *z'* [18].

Thus, for each joint, the moment of inertia of each link pivoted about it was calculated by adding the products of each individual mass (*m*) by the square of the respective perpendicular distance (*h*) between the body's centre of mass and the joint.

2.5. Hardware and Software Selection

The following hardware and software choices were made viz;

2.5.1. Motor Selection

Servomotors (commonly called servos) were employed to provide the required torque at each joint location. The servos can control the angular position, velocity and acceleration of the output shaft in a precise manner. Inside the servo is a combination of gears and a potentiometer with circuitry that make it possible to set its position fairly quickly and precisely and within a 180-degree range of travel [20]. The resistive torque expected to act at each joint was evaluated and a motor (servo) with an appropriate torque rating was selected for each joint.

2.5.2. Gripper Joint

A standard mini servo gripper template from Lynxmotion® Robotics was used to implement the gripper design. It is based around two counter-rotating gears which use levers to make the grippers open and close laterally [21]. The design uses a Hitec® HS-225MG servo. Thus, a Hitec® HS-225MG servo was selected to provide the needed torque for the opening and closing of the gripper jaws. Note: The 'MG' added to the servo name is an abbreviation for 'Metal Gear'. It indicates that the drive gears of the servo are made of metals as opposed to plastics used for most standard servos.

2.5.3. Wrist Joint

Total resistive torque at wrist joint was calculated to be 6.012 x 10^{-5}N-m or 6.012 x10^{-4} kg-cm. Hence, a Hitec® HS-225MG servo with a torque rating of 3.9kg.cm was selected to provide sufficient torque at the joint.

2.5.4. Elbow Joint

The resistive torque at elbow joint was calculated to be 0.2756N.m or 2.756kg.cm. Hence, a Hitec® HS-422 standard servo with a torque rating of 3.30kg.cm at 4.8volts was selected to provide the torque needed at the joint location. The extra torque of 0.544kg.cm caters for the extra resistive torque due to inertia of the payload, servo bearing friction, irregularities in the robot frame design and future modifications of the arm design.

2.5.5. Shoulder Joint

The resistive torque at shoulder joint was calculated to be 0.6055N.m or 6.055kg.cm. Hence, a Hitec® HS-645MG Ultra Torque servo with a torque rating of 7.70kg.cm at 4.8volts was selected to provide the torque needed at the joint location. The extra torque of 1.645kg.cm caters for the extra resistive torque due to inertia of the payload, servo bearing friction and irregularities in the robot frame design.

2.5.6. Base Joint

Total resistive torque at base joint was calculated to be 0.0304N.m or 0.3040kg.cm. Thus a Hitec® HS-422 servo with a torque rating of 3.30kg.cm at 4.8volts was selected to provide the torque needed at the joint location. The properties of the selected servos are presented in Table **2**.

2.5.7. Servo Controller Selection

A servo motor is controlled by sending a series of pulses to it. This is called Pulse Width Modulation (PWM) signal. A pulse is sent to the servo every 20 milliseconds. Depending on the pulse width (duration of the pulse), the motor spline or shaft moves to a position within a 180-degree range. For a hitec® servo, a 0.9 millisecond (900µs) pulse moves the motor to 0 degrees, a 2.1 millisecond (2100 µs) pulse moves it to 180 degrees. Any pulse between 900µs and 2100µs moves the motor to a position proportionally between 0 and 180 degrees [20].

Three servo controllers were considered for the implementation of this project - the Pololu® Micro Maestro 6-Channel USB (Universal Serial Bus) servo

Table 2: List of Selected Servos and their Specifications

Joint	Servo	Torque at 4.8v (kg.cm)	Mass (g)	Dimensions (L x W x H) (mm)
Gripper	HS-225MG	2.60	27.94	32.26 x 16.76 x 31.00
Wrist	HS-225MG	2.60	27.94	32.26 x 16.76 x 31.00
Elbow	HS-422	3.30	45.36	40.39 x 19.56 x 36.58
Shoulder	HS-645MG	7.70	55.00	40.39 x 19.56 x 37.59
Base	HS-422	3.30	45.36	40.39 x 19.56 x 36.58

Source: hitecrcd.com.

controller, Images Company's® SMC-05A servomotor controller and Lynxmotion's® SSC-32 servo controller.

The Pololu® Micro Maestro is a highly versatile servo controller. It supports three control methods: USB for direct connection to a computer; TTL (transistor-transistor logic) serial for use with embedded systems; and internal scripting for self-contained, host controller-free applications [22]. Its extremely precise, high-resolution servo pulses has a jitter of less than 200 ns, making it well suited for high-performance robotics applications, and its in-built speed and acceleration control for each channel make it easy to achieve smooth, seamless movements without requiring the control source to constantly compute and stream intermediate position updates to it [22].

Images Company's® SMC-05A servomotor controller on the other hand was ascertained from the manufacturer's website to be compatible with Hitec® servos. It allows manual and PC (Personal Computer) control of five servomotors (Hitec® and Futaba® brands). Servomotors may be controlled manually, via on-board switches, or by a Windows program through a serial communication port (RS232) on a Windows PC. Universal three position headers make it easy to connect servomotors by just plugging them into the board [23]. The SSC-32 (serial servo controller) is a small preassembled servo controller with 32 channels of 1µS resolution servo control, thus enabling accurate positioning of the servo output shaft. The range of its generated Pulse Width Modulation (PWM) signal is 0.50ms to 2.50ms for a range of about 180° and it features an Atmel ATMEGA168-20PU microcontroller. It also supports Futaba® and Hitec® servos [24]. The Pololu® Mini Maestro was preferred to the SSC-32 because it has a USB Software Development Kit that allows a less tedious programming with Microsoft's C# programming language. Also, it can be directly connected to the host PC (using the USB connection feature) without the need for a USB-to-serial adapter.

2.5.8. Interfacing Software

To enable the control of a robot from a PC, an interfacing software, which enables the operator to send instructions to the servo controller is needed. The Pololu® Mini Maestro was used for this function.

2.6. Final Arm Model

The components of the arm were cut from acrylic plastic sheet with the use of a fret saw and a hand-drilling machine. The acrylic plastic used has a thickness of 3mm and to further reduce the load imposed on the shoulder and elbow servos, a load-balancing spring was added to the elbow joint of the robotic arm. The fabricated robotic arm components were assembled using the 3D model as a guide. The servos were inserted at the various joints in such a way that centring all the servo horns would make the arm to assume an upright pose as shown in Figure **7**. The centring of the servo horns was achieved by connecting the servos to the Pololu® Micro Maestro™ servo controller and sending PWM (Pulse Width Modulation) signals of 1500ms to them from the Pololu® Maestro™ Control Centre. The servo controller was connected to the PC running the Pololu® Maestro™ Control Centre via a USB cable which also supplied enough current to power the servo controller board. The servos on the other hand were powered using a 3-Ampere regulated DC power pack.

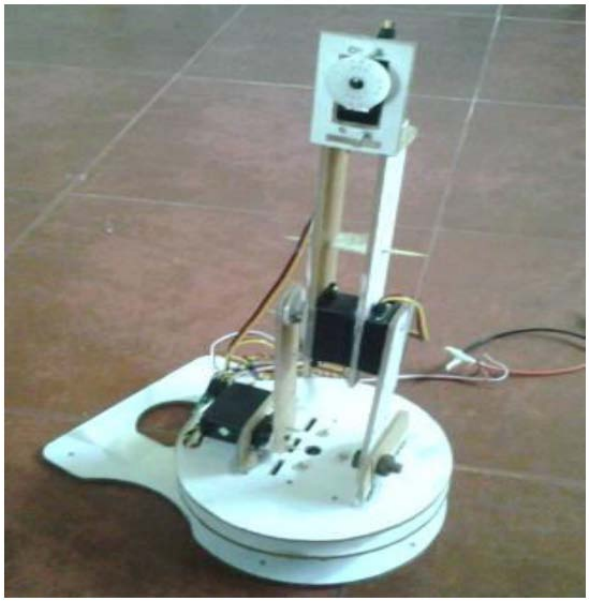

Figure 7: Robotic arm in upright pose.

The final robotic arm model with the selected servos is shown in Figure **8**, on a pick and place task.

3. RESULTS AND DISCUSSION

The interfacing software (shown in Figure **9**) requires the user to enter the desired coordinates of the centre of the gripper after which he clicks a button labelled "click to compute inverse kinematics" to compute the inverse kinematics of the robotic arm. On clicking the button, the angle through which each joint will rotate in order to position the gripper at the specified coordinate is displayed, if the specified

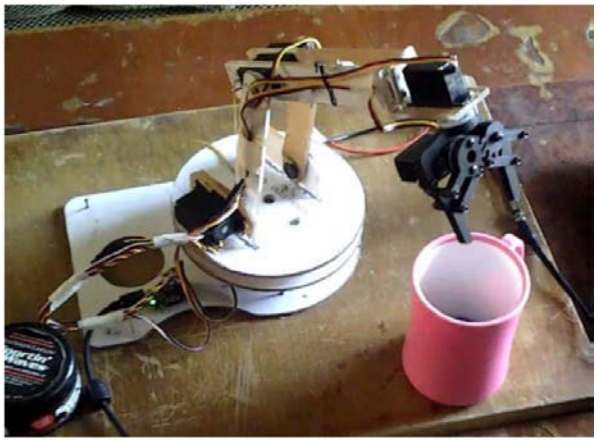

Figure 8: Robotic arm performing a pick-and-place task.

coordinates are within the robotic arm workspace. If the specified coordinates are not within the workspace of the robotic arm, the user is notified on clicking the button. The user then clicks the button labelled, "click to position gripper" to move the gripper to the specified coordinates.

The robotic arm set-up was tested by inputting series of coordinates in the control software. The resulting position of the gripper centre for each set of coordinates was then measured. To ensure ease of measurement, a calibrated cardboard was placed behind the robotic arm to measure the y and z coordinates while another one was placed below the robotic arm to measure the x coordinates. A picture of the set-up is shown in Figure **10**, while the test results are presented in Table **3**.

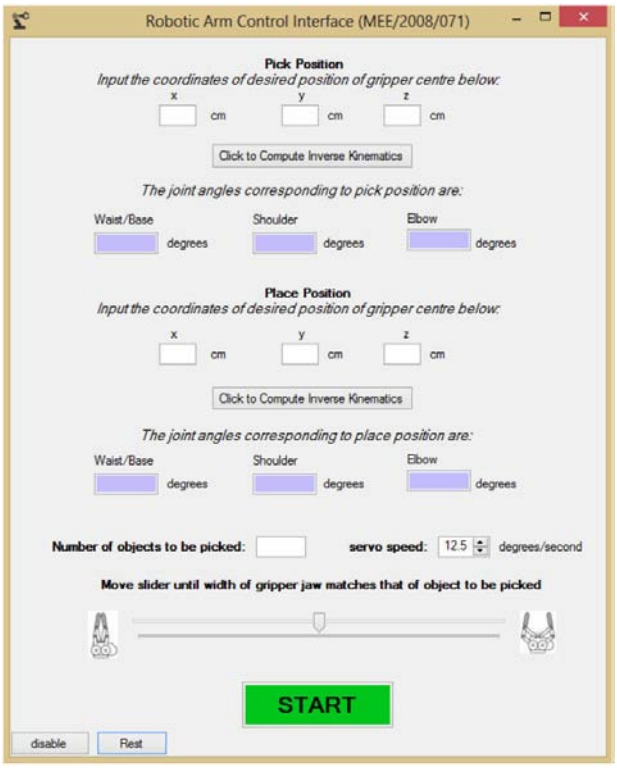

Figure 9: GUI of interfacing software for controlling robotic arm.

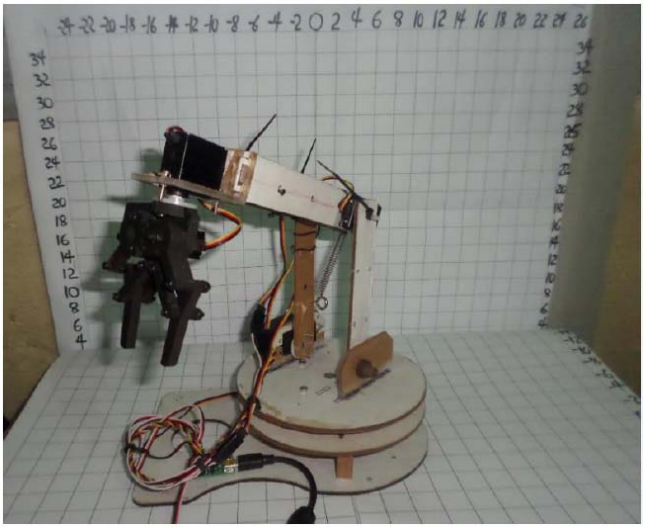

Figure 10: Setup for measuring coordinates of gripper position.

It could be observed in Table **3**, that the robotic arm performed reasonably well, with minimal deviations. This provides a platform for students to experiment and learn the basics of the robotic arm.

4. CONCLUSION AND RECOMMENDATIONS

A 5 DoF robotic arm was designed and fabricated, which can be used for demonstrative and educational purposes. The robotic arm can be reprogrammed in order to adapt it to a variety of tasks. This could prove also to be useful in an industrial environment, especially in manufacturing, packaging and mining industries. It is believed that the use of the arm for educational and instructional purposes will stimulate the interest of students in the field of robotics, especially in a developing country like Nigeria. There is however room for improvement in the design and implementation of the robotic arm system as outlined below:

- The number of degrees of freedom of the robotic arm can be increased in order to expand its workspace, thereby making it more versatile.
- Motors with higher torque ratings can be used to power the joints so as to ensure that the robotic arm remains in position even when electric current is not supplied to the motors.

Table 3: Coordinate Input Test Result for Robotic Arm

Input Coordinates (cm)			Output Coordinates (cm)			Deviation (cm)		
x	***y***	***z***	***x***	***y***	***z***	**$\|\Delta x\|$**	**$\|\Delta y\|$**	**$\|\Delta z\|$**
10	4	19	11	5	19	1	1	0
5	4	23	5	3	25	0	1	2
8	-7	19	10	-7	19	2	0	0
8.5	0	33.5	9	0	34	0.5	0	0.5
14.9	0	10.1	17	0	11	2.1	0	0.9
8.5	-23	9	12	-21	7	3.5	2	2
10	12	18	12	13	17	2	1	1
9	-7	27	10	-7	25	1	0	2
8	-23	10	9	-23	9	1	0	1
10	7	28	11	7	27	1	0	1
15	-7	15	15	-6	15	0	1	0
15	15	15	14	16	14	1	1	1
10	-20	8	11	-21	7	1	1	1
12	-10	13	13	-11	12	1	1	1
11	12	14	13	11	14	2	1	0
12	16	10	14	15	9	2	1	1

- Wireless control of the robotic arm using Bluetooth™ or Wi-Fi™ (IEEE 802.11x) technology can be implemented.
- Object detection and collision avoidance can be implemented by adding proximity sensors to the robotic arm.

REFERENCES

[1] Corke P. Robotics, Vision and Control Fundamental Algorithms in MATLAB. (Siciliano B, Khatib O, Groen F, Eds.) Berlin: Springer-Verlag 2011.

[2] Ryan D. History MV for Industrial Robots. In D. Ryan, E - Learning Modules: Dlr Associates Series. Bloomington: AuthorHouse 2012; pp. 28-29.

[3] KUKA. History 2013. KUKA: http://www.kuka-robotics.com/usa/en/company/ group/milestones/1973.htm. Accessed September 5, 2013.

[4] Angelo JA. Robotics: A Reference Guide to the New Technology. Westport: Greenwood Press 2007.

[5] Chen C, Huang H, Wan T. 'Omnidirectional Mobile Home Care Robot'. National Chung-Hsing University, Department of Electrical Engineering, Taichung 2006.

[6] Jenne F. 'Combined control of a mobile robot and a robot arm'. Swiss Federal Institute of Technology , Autonomous Systems Laboratory, Zurich 2009.

[7] Blackmore M, Furniss J, Ochsner S. '5 DOF Robotic Arm'. Poland State University, Embedded Robotics, Poland 2009.

[8] Adhikari U, Islam MM, Noser J, Puliafico K. 'Elderly Assist Robot'. Worcester Polytechnic Institute, Worcester 2010.

[9] Akinwale OB. 'Development of a robust ILab platform for conducting robotic arm experiments'. M.Sc. thesis. MSc. Thesis, Obafemi Awolowo University, Electronic and Electrical Engineering, Ile-Ife 2011.

[10] Adebola SO. 'Design of an improvised robot arm system'. BSc. thesis, Obafemi Awolowo University, Ile-Ife 2012.

[11] Herman P, Mosteoyz AR, Muri AC. 'A low-cost custom-built robotic arm on the TurtleBot platform'. University of Zaragoza, Dpt. de Informática e Ingeniería de Sistemas, Zaragoza 2012.

[12] Ishola BI. Final Report on The Development of Strength of Materials Remote Laboratory: A Non-EE/CS Ilab For Static Bending Test. BSc. Thesis, Obafemi Awolowo University, Department of Electronic and Electrical Engineering, Ile-Ife 2012.

[13] Siciliano B, Sciavicco L, Villani L, Oriolo G. Robotics: Modelling, Planning and Control. London: Springer-Verlag London Ltd. 2009.

[14] Denavit J, Hartenberg RS. Kinematic notation for lower-pair mechanisms based on matrices. Journal of Applied Mechanics 1955; 23: 215-221.

[15] Abdel-Malek K, Yeh H. Analytical Boundary of the Workspace for General 3-DOF Mechanisms. International Journal of Robotic Research 1997; 16(2): 198-213.

[16] Sanchez CAM. Dynamics and motion of a Six Degree of a Six Degree of Freedom Robotic Manipulator. MSc. Thesis, University of Saskatchewan 2012.

[17] petercorke.com. Robotics Toolbox 2013. PeterCorke.com: http://petercorke.com/Robotics_Toolbox.html. Accessed August 24, 2013.

[18] Jha DK. Dynamics of Rigid Bodies. In D. K. Jha, Text Book Of Rotational Mechanics. Delhi: Discovery Publishing Pvt. Ltd. 2006; pp. 12-17.

[19] Benenson W, Harris JW, Stöcker H, Lutz H. Handbook of Physics. New York: Springer Publishing 2006.

[20] Sullivan DO, Igoe T. Physical Computing: Sensing and Controlling the Physical World with Computers. Boston: Thomson Course Technology PTR 2004.

[21] Lynxmotion. AL5B Robotic Arm Combo Kit (with RIOS Software. Lynxmotin: Imagine it. Build it. Control it. 2013. http://www.lynxmotion.com/p-673-al5b- roboticarm-combo-kit-with-rios-software.aspx. AccessedOctober19, 2013.

[22] Pololu.com. Micro Maestro 6-Channel USB Servo Controller (Assembled). Pololu Robotics and Electronics 2013. http://www.pololu.com/catalog/product/1350 Accessed August 24, 2013.

[23] Imagesco.com. SMC-05 Servo Motor Motion Controller. Images Scientific Instruments 2013. http://www.imagesco.com/servo/smc05.html. Accessed August 24, 2013.

[24] Lynxmotion.com. SSC-32 Servo Controller. Lynxmotion-Imagine it, build it, control it 2013. http://www.lynxmotion.com/p-395-ssc-32-servo-controller.aspx. Accessed August 24, 2013.

Received on 03-12-2019 Accepted on 23-12-2019 Published on 29-12-2019

DOI: https://doi.org/10.31875/2409-9694.2019.06.7