Design and Construction of a Computer-Controlled Robotic Arm

Eduardo M. Gallofin, Jr.
Ateneo de Manila University
Egallofin@yahoo.com

Michael Loe G. Fernandez and Carlos M. Oppus
Ateneo de Manila University
Milo@atenista.net, coppus@ateneo.edu

ABSTRACT

The robot arm that was developed was modeled after a commercially available 5-axis robotic arm. The movements of the robot arm were controlled via the parallel port of a computer using various keys on the keyboard. Several DC motors made possible the movements of the parts/joints of the robot arm. Various circuit controllers other than the DC motor controller circuit were implemented so as to protect the computer from damage resulting from certain failures of the hardware and to automate the robot for repetitive tasks. A program was also developed using Visual C++ version 6 under Windows 98 platform to act as an intermediary between the user and the robot arm. Moreover, speed control for the parts/joints was included through a software-implemented pulse width modulation technique.

Key Words – Robotics, Computer Interfacing, Control System.

I. INTRODUCTION

The word robot is derived from the Czechoslovakian term robota which is generally translated as 'forced labor.' This means that the original conception of a robot, as far the etymology of the word is concerned, was to be a capable servant. It was first used in the play by the Czechoslovakian author Karel Capek entitled R.U.R. (Rossum's Universal Robots). In the play, robots were portrayed as small, artificial and anthropomorphic creatures strictly obeying their master's orders. From this humble conception, many authors began getting inspirations from the concept of a robot. The most famous of all the authors that wrote about robots is Isaac Asimov. He was the one who formulated the four laws of robots: (0) a robot may not injure humanity, or through inaction, allow humanity to come to harm, (1) a robot may not injure or harm a human being, or through inaction, allow a human being to come to harm, (2) a robot must obey orders given to it by human beings, except where such orders would conflict the 0th or 1st law, and (3) a robot must protect its own existence as long as such protection does not conflict with the previous laws. As time passed, people began formulating an encompassing definition of a robot.

As currently defined, robots exhibit three key elements: (1) programmability, implying computational or symbolic manipulative capabilities that a designer can combine as desired (a robot is a computer), (2) mechanical capability, enabling it to act on its environment rather than merely function as a data processing or a computational device (a robot is a machine), and (3) flexibility, in that it can operate using a range of programs and manipulate and transport materials in a variety of ways.

This kind of description does not sway too far from what really most robots in the world are doing. Most robots used nowadays are designed for heavy, repetitive manufacturing work. They are specifically designed to handle certain tasks that are difficult, dangerous, or to boring to human beings. Robots can do more work more efficiently than humans can since robots are precise. They always do the same task with such precision over and over no matter how long they have worked. Robots nowadays are becoming more and more important in most industries of the world.

The most common of all these manufacturing robots is the robot arm. A typical robot arm is made up of seven segments joined by six joints. Usually a stepper motor is used in order to track the movement of the robot arm. The reason for this is quite obvious since stepper motors are designed to move in exact increments unlike DC motors. With such configurations, a computer may be able to control or maneuver the robot very precisely, repeating exactly the same environment over and over again.

Some advanced robot arms make use of sensors like motion and pressure sensors in order for it to detect foreign obstacles and avoid breaking or dropping what it is carrying. Robot arm also vary with the type of end effector that they are using. The kind of end effector that a particular robot arm is using is very much dependent on the kind of task the robot is designed for: blowtorches for auto assembly lines robots, drills for metal application robots, and spray paints for decoration oriented robots.

II. OBJECTIVES

The main objectives of the thesis are (1) to be able to design and construct a robot arm, and (2) to be able to control the robot arm using a computer through a keyboard. The first object is very straightforward and does not merit much discussion. The second objective requires a working knowledge of PC to hardware communication. Additional hardware components aside from the robot arm like opto-isolator circuits and limit switches will be implemented in order to facilitate the safe control of the arm.

An additional objective will be to program the robot arm to do a simple task. This option, if to be implemented with accuracy and precision, requires a more challenging task of familiarizing the science of kinematics both forward and reverse kinematics. However, the implementation used for the automation of this robot arm is time-based. This means that when automating the robot arm, a program records the length of time of a certain joint from moving from one position to another. This kind of automation, however, is not very accurate or precise since it doesn't take into consideration the actual load that the arm is carrying.

III. SCOPE AND LIMITATIONS

The scope of the thesis only covers the basics of design and computer control of a robot arm. As was mentioned before, the main goal of the thesis is to create an 'educational' robot arm that may provide a basis or template for a creating complex robot arm. It does not take into consideration other complex yet pertinent factors that are usually being considered by expert designers and programmers like gripper pressure sensors, torque computations, exact position of the parts/joints, and mathematical formulations for movement control. Although the arm may be programmed to do a rather simplistic task of getting and putting of some things, complex task such as chess playing or solving a problem like the towers of Hanoi will not be part of the program.

In terms of the physical aesthetics, the arm doesn't really resemble the usual industrial robot arms currently being used. Moreover, when it comes to the movement of the joints, the arm will not be very precise in moving from one point to another due to the fact that the motors that drive the joints are all DC motors and precision is not very common in most DC motors.

IV. METHODOLOGY

The main goal of the thesis is to design and construct a robot arm that may be controlled through a computer. In order to go about the thesis, the whole thesis was divided into two major parts each having modules of its own.

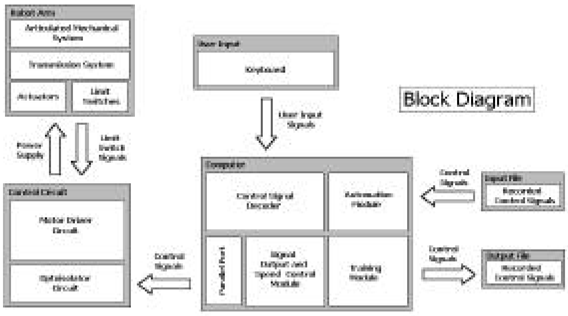

Figure 1. Block Diagram of the Thesis

4.1 Robot Arm Design

The design of the robot arm was largely based on the Lynxmotion 5-axis robot arm. The materials used for the construction were chosen in connection to the type of dc motor that were used to move the parts/joints.

4.2 Circuit Controllers

There are six (4) main controller circuits of the thesis: the opto-isolator circuit, the limit switch circuit, the dc motor controller circuit, and the initialization circuit.

4.3 Opto-isolator

An opto-isolator is a semiconductor device that allows signals to be transferred between circuits or systems, while keeping those circuits or systems electrically isolated from each other. In this regard, an opto-isolator circuit was used in the thesis so as to protect the computer from any disruptions coming from any of the main hardware components.

4.4 Limit Switches

Limit switches were placed strategically on the each of the moving parts of the robot arm except for the gripper. The purpose of these limit switches is to stop a certain part/joint from moving out of its range of flexibility or reach. This is a safety precaution so that even if there a signal coming from the computer to move the joint in a certain direction, then the joint will not move in the said direction if the limit switch of the joint associated with the said direction is asserted.

4.5 DC Motor Controller

In order to move the different parts/joints of the robot arm, 5V DC motors were attached to the different parts/joints of the arm that would drive their movement. To be able to control the movement of the motors, a circuit that will be interfaced with the computer was designed. The circuit would accept (indirectly, since the signal has to pass through the opto-isolator circuit and the limit switch circuit) an input from the computer, depending on whether the user wants the motor to move in a certain direction or not. There is a previous thesis that used this type of circuit. Last year, the hospital bed automation with voice control thesis used a 12V DC motor for the control of the joints of the bed. The group opted to use the same circuit but with some minor modifications.

4.6 Software

A program that would control the motion of the robot arm was developed. The program would act as the intermediary between the user and the robot arm. The controls for the movement of the parts/joints will be through a keyboard. Specific buttons will be specified by the software in order to move certain parts/joints of the robot arm. Moreover, the pulse width modulation for speed control of the robot arm was implemented through software.

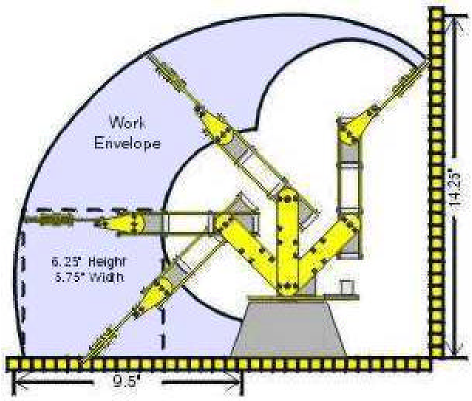

Figure 2. Lynxmotion 5-axis Robotic Arm

V. HARDWARE DESIGN

5.1 Robot Arm: Robotic Articulations

Robot arms are basically composed of three major mechanical components. First is the Articulated Mechanical System, or the articulations which is composed of the robots limbs and end effectors (i.e., gripper), the Actuators (i.e. motors) and the Transmission system (i.e. gears). The design of the Robot Arm was originally based on the Lynxmotion Robot Arm kit. However, several major changes were made in the design to satisfy the specifications set for this thesis.

The gripper design is simple which includes two sliding clamps which are driven to close or open by a motor. Included with the design is the gripper handle that supports and provides motion to the gripper relative to the upper arm.

The upper arm supports to the gripper and the gripper handle and provides rotation relative to the lower arm. The lower arm, on the other hand, provides support and rotation to the gripper and the upper arm relative to the rotating base.

The rotating base supports the gripper, upper arm and the lower arm and provides them with rotation with axis perpendicular to the ground. The rotating base rotates relative to the main base, which in turn provides support and stability to the whole arm.

Plexiglass is lightweight, considerably durable and easy to cut. Thus, it was the material chosen for the construction of the robot's limbs. The plexiglass components were attached to each other using commercial cyanoacrylate adhesives and reinforced with screws and nuts.

5.2 Robotic Actuators

The robot's actuators were five old DC motors recycled from a non-functional robot arm thesis. The DC motors were Futaba S148 DC servo motors. These servo motors were originally designed by the vendor to include a built-in Pulse Width Modulation circuit that should allow changing of the motor's speed. Also, the servo motor includes a small potentiometer that rotates with the main shaft. This is supposed to give feedback on the position of the motor's shaft. The potentiometer limits the rotation of the shaft to about 90 degrees only.

However, the servomotors were found to have been modified in the previous thesis to allow free running, 360 degree rotation of the shaft. The potentiometers were disconnected from their respective main shaft and the PWM inputs were also disconnected. Without the PWM feature and the positional feedback, the servo motors were reduced to function as ordinary DC motors.

But these servo motors were still used because of the significantly higher torque they offer compared to ordinary DC motors.

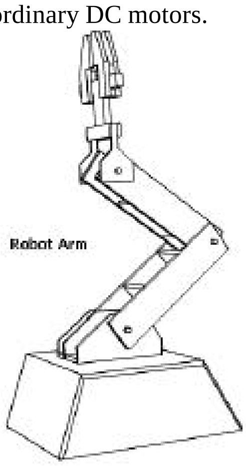

Figure 3. Robot Arm Design

5.3 Robot Transmission System

Ordinary plastic spur gears recycled from the old robot arm thesis were the components of the robot arm's transmission system. Gear trains consisting of two, three and four gears were used to increase the torque delivered by the motor, just enough to lift the articulations plus a considerable amount of load. Also, the gear trains provided a high gear ratio enough to hold the limbs in place even when no power is supplied to the motor.

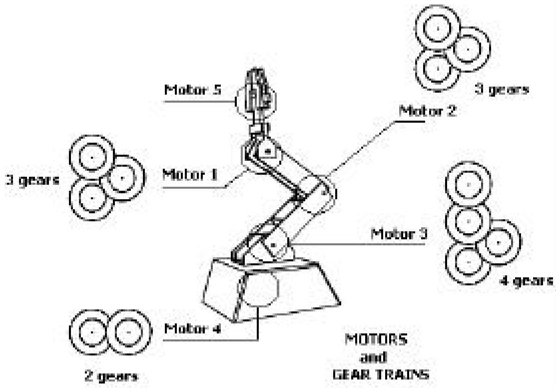

Figure 4. Motor and Gear Trains.

Illustrated in Figure 4 are the locations of the five motors in the robot arm and their respective gear trains. The ideal amount of torque delivered by the gear trains is also computed.

5.4 Limit Switches

To ensure that the user or any control program will not drive the motor's limbs beyond its mechanical limits and damage it, limit switches were used in the robot. The limit switches are activated when the limbs reach its limits. The positions of the limit switches are specified below. The connection of the limit switches to the control circuit will be discussed later.

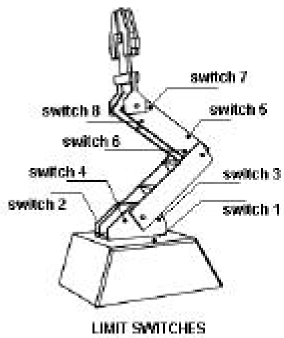

Figure 5. Location of Limit Switches.

5.5 Pin Assignments

A flat cable with a 25-pin female connector attached to the robot arm's base provides the connections to the inputs for the power to the motor and the outputs of the limit switches. The pin assignments are as follows:

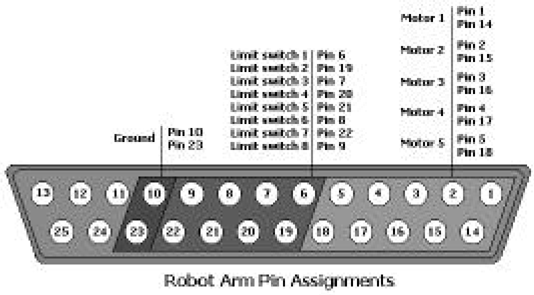

Figure 6. Robot Arm Pin Assignments.

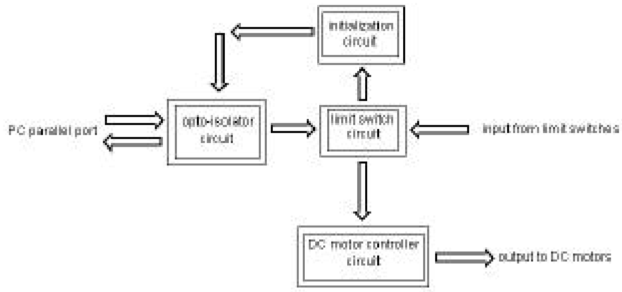

Figure 7. Block Diagram of the Controller Circuit Connections.

A. Controller Circuits

The opto-isolator circuit, limit switch circuit, DC motor controller circuit, and the initialization circuit provide the necessary elements for interfacing the robot arm with the computer. The opto-isolator circuit handles all the input and output signals of the computer. The outgoing signals from the computer are then connected to the limit switch circuit where the circuit will check if the limit switch associated with the signal coming from the computer is asserted. If the limit switch associated with the signal is asserted to low, then the limit switch circuit will output a logic low to the DC motor controller circuit and thus will disable the movement of the part/joint that is associated to the output signal of the computer. Otherwise, the limit switch circuit will output a logic high to the DC motor controller circuit and thus would allow the part/joint associated to the output signal to move.

On the other side of the coin, the input signals to the computer will start mainly from the limit switches. The limit switch circuit aside from disabling part/joint movement also functions as a control signal for the computer. The program for the robot arm was designed to have an initialization state wherein all parts/joints will be moved to one of its maximum reach. Whenever the initialization state is invoked (usually when the program first starts), the program enters into the initialization state wherein it sends a logic high to all of the part/joints making them move to one of their maximum reach and after which the initialization circuit sends a high signal to the computer. When that signal comes, the program exits the initialization state and the user may then be able to control the movements of the robot arm.

VI. SOFTWARE DESIGN

The control program for the robot arm is programmed using Visual C++ 6.0 and is composed of 4 major components, namely, the Control Signal Decoder, the Signal Output and Speed Control Module, the Training Module and the Automation module.

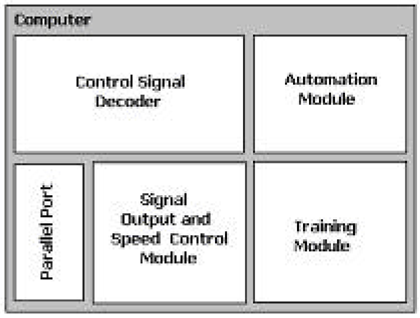

Figure 8. Block Diagram of the Control Program.

The control signal decoder receives input from the keyboard and decodes the key inputs to their respective output values to the parallel port. The key assignments for the control of the robot arm's movements are shown in the figure on the next page.

The signal output and speed control module is a thread executed entirely dedicated to output the value of the global variable generated by the signal decoder. The program also includes the pulse width modulation module that enables the user to specify the speed for the movement of the robot's limbs.

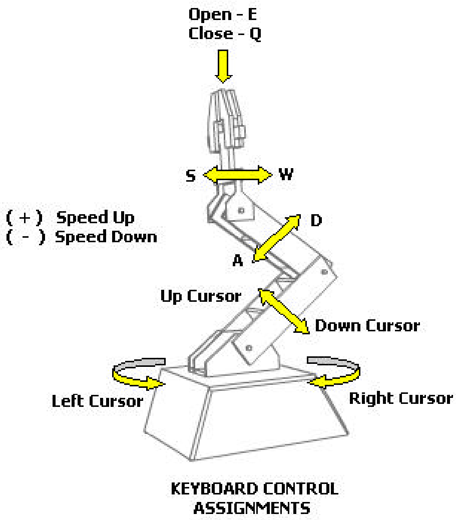

Figure 9. Key Assignments for the Movement of the Parts/Joints of the Robot Arm

The training module enables the user to record the instructions he/she inputs to the robot arm. These instructions which are the output values to the two parallel port addresses are written to a text file.

Training the robot arm involves the actual movement of the robot's limbs and the recording of the output values to the parallel port. Thus, the user must control the robot arm to his/her desired positions while the program records the parallel port output values.

The automation module executes the instructions written in the input text file. The program parses the file by line, outputs the values in the first and second lines to the addresses 0x378 and 0x379 respectively. The duration of the output is indicated in the third lines of the file. The module also allows the user to input the number of repetitions it the program should execute the instructions in the file.

Provided below is the program interface where the user inputs the instructions to the robot arm.

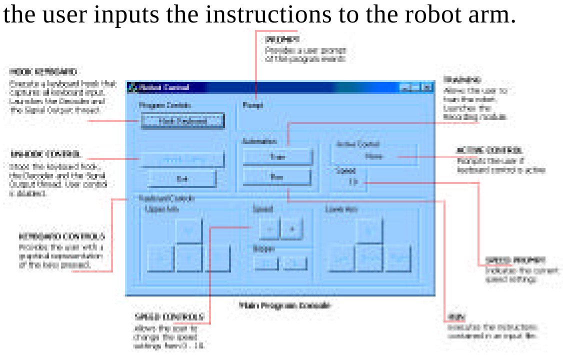

Figure 10. Program Interface.

VII. RESULTS AND OBSERVATIONS

The results and observations of this thesis would come mainly from the ways in which the robot arm behaves when it is controlled manually or automatically. In the manual control for the robot arm there seem to be no problems that are visible when it comes to the movement of the part/joints but when it comes to automation of the robot arm, certain problems would occur when it concerns the precision and accuracy of doing a repetitive task. The robot arm is capable of doing repetitive task but after doing the task a certain number of times there would be a visible error when it comes to the position of the parts/joints. The cause of this positioning error would come mainly from the fact that the motors used for the movement of the part/joints were not really designed for precision or accuracy. In order to minimize this type of positioning error, as much as possible the robot arm would have to undergo the initialization stage before repeating the task assigned to it. However, there is also a positioning error caused by the load that the robot arm would be carrying since the automation is solely time-based. There was no mechanism implemented in order to minimize this type of positioning error.

The maximum weight capacity for the robot arm was not tested fully since there wasn't a clear data on how much stress the material that the robot arm is made from can handle. The maximum load capacity that was tested was only up to 230 grams with only the lower arm motor and the robot arm fully stretched. The reason why it was only the lower arm motor that was tested is that it is this motor that usually carries the most load. Making the arm fully stretched while doing the maximum load test assumed that the heaviest position that the robot arm can handle. However, the lifting capability of the robot arm was tested under certain small objects and the results were as expected.

As for the speed of the robot arm, the PWM algorithm implemented was able to control the speed of the robot arm to a certain degree. The only limitation to the maximum and minimum speed of the robot arm in moving its parts/joints would come mainly from the program. The program specified a certain range of values in which the speed of the robot arm may be varied. The user cannot force any speed outside of this specified range of values.

VIII. RECOMMENDATIONS

The main component that needs improvement is the automation of the robot arm. It needs to have more precision and accuracy in moving its parts/joints. Forward and reverse kinematics will have to be included in the program so as to make the movements of the robot arm more independent from the user. Moreover, feedback from the motor will be needed in order for the program to know exactly where the robot arm is in space. All the user needs to do is to specify to a location within the immediate space and the robot arm will move to that point in space. The robot will need not go through the initialization when starting an automation program. Furthermore, using feedback would minimize the positioning error when the robot arm is carrying a load. This feature may be implemented using optical encoders attached to the robot arm. The signals from the encoders may be input to the parallel port and should provide information on the robot's position.

Another improvement that may be included is to create a set of commands that will instruct the robot to do certain tasks. Furthermore, these set of commands may be put together like a program that would enable the robot arm to do complex tasks.

REFERENCES

[1] Andeen, Gerry B., ed. " *Robot Design Handbook*". McGraw-Hill Book Company: USA, 1988.

[2] Fu, K.S., R.C. Gonzales, and C.S.G. Lee. *"Robotics: Control, Sensing, Vision, and Intelligence*". McGraw-Hill Book Company: Singapore, 1987.

[3] Clarke, Roger. "*Asimov's Laws of Robotics: Implications for Information Technology*". September 1994.

[4] Amil, Nashieba, Alexis Artes, Frederico Condes, and Willy Custodio. " *Hospital Bed Automation with Voice Control*." Thesis paper, Ateneo de Manila University, 2002.