DESIGN AND DEVELOPMENT OF THE CUSTOM CONTROLLER FOR RHINO XR4 ROBOTIC ARM

NAME	SEAT NUMBER
MUHAMMAD ARSHAD HASAN	BM-11029
MUHAMMAD DANYAL SIKANDER	BM-11019
MUHAMMAD HASSAN KHAN	BM-11024
HINA MUNIR	BM-11031

PROJECT SUPERVISOR: PROF. DR. ALI RAZA JAFRI

EXTERNAL SUPERVISOR: MR. SAAD ABDULLAH

DESIGN AND DEVELOPMENT OF THE CUSTOM CONTROLLER FOR RHINO XR4 ROBOTIC ARM

BATCH 2011-2012

BY

NAME	SEAT NUMBER
Muhammad Arshad Hasan.	BM-11029
arsh.hasan@hotmail.com	
Muhammad Danyal Sikander	BM-11019
danial.sikander@yahoo.com	
Muhammad Hassan Khan.	BM-11024
hassan_muhammadi@yahoo.com	
Hina Munir	BM-11031
munirhina3@gmail.com	

DEPARTMENT OF BIOMEDICAL ENGINEERING

NED UNIVERSITY OF ENGINEERING AND TECHNOLOGY

CERTIFICATE

This is to certify that the following students of batch 2011-2012 have successfully completed the final year project in partial fulfillment of requirements for a Bachelor's Degree in Biomedical Engineering from NED University of Engineering and Technology, Karachi, Pakistan.

MUHAMMAD ARSHAD HASAN	BM-11029
MUHAMMAD DANYAL SIKANDER	BM-11019
MUHAMMAD HASSAN KHAN	BM-11024
HINA MUNIR	BM-11031

PROJECT SUPERVISORS

----------------------------------	-------------------------------
(Internal)	(External)
(Prof. Dr. Ali Raza Jafri)	(Mr. Saad Abdullah)
Department of Biomedical Engineering	

NED University of Engineering and Technology

ACKNOWLEDGEMENTS

In the name of Allah most gracious and most merciful, Glory be to HIM who enabled us to learn, explore and provided us with the opportunity of working on this project and successfully completing it.

It is not possible to return what we owe to our parents, they helped us to their best and without their support, prayers, care and encouragement we could have never been able to accomplish this goal.

We are obliged to Prof. Dr. Ali Raza Jafri for his supervision and motivation. His valuable suggestions have helped us a lot with our work. We are also gratified to Mr. Saad Abdullah who considered us eligible for doing this project. His guidance lead us to the right way of fulfilling this task.

We are indebted to Mr. Osama Mazhar for his constant support through every phase of the project. He helped us in all the possible ways he could do. We are also grateful to Miss Uzma Nadeem for her help regarding the compilation of this report.

A very special thanks to the LabVIEW Makers hub members specially Mr. Nathan and Mr. Sam for their guidance and valuable suggestions.

We really want to thanks our teachers who taught us and delivered their best to us. In the last we really want to thanks our class mates for their support and appreciation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Robotics is one of the most studied and researched field these days. The ultimate goal of Robotics is to make such robots that are equally capable as humans with regard to their cognitive skills. Robots find applications everywhere, so in Education. One such robot is RHINO XR4, it is an Educational Robot for teaching students the basics of Robotics. This project aims to design and develop a custom controller for RHINO XR4 robotic arm. The developed controller will replace MARK IV controller (Original controller of RHINO XR4). Making of this custom controller involves the development of its various modules like power, controller and motor driver modules along with the development of Graphical User Interface. This development also incorporates the Forward Kinematics of RHINO XR4. The custom controller has been developed with an ambition of replacing MARK IV with certain enhancements and improvements. Custom controller has Power module, Controller module and Motor Driver module in it. Controller in the custom controller is Arduino Mega 2560. PID algorithm has also been implemented for accurate control. The Motor Driver module of the custom controller is made up of three motor driver shields Pololu MC33926. Graphical User Interface for the custom controller has been developed using LabVIEW. The custom controller has been designed in such a way that it can be controlled serially with a help of an operating system, having LabVIEW installed in it. The GUI along with its serial control through the operating system, make the robot control easy and interactive. The developed controller has less weight, size and cost than the previous controller MARK IV.

CHAPTER NO.01

INTRODUCTION

Now a days Robotics is grabbing the most attention of researchers and engineers. It has revolutionized our lives. Robots are found everywhere and their applications are numerous. They are mainly used in industries, educational institutions and research laboratories. They are the backbone of industries these days. Being accurate, repeatable, and precise and requiring no safety measures as human do, they have increased industrial productivity. Robots with appropriate mechanical structure, robust controlling and efficient programming give us plentiful possibilities.

Depending upon the functions and purposes for what they are built, they have structures and components accordingly. Robots, besides their mechanical structures, incorporates many other components but they mainly have all the listed parts below;

1. Manipulators
2. End effector
3. Actuators
4. Sensors
5. Controller
6. Processor
7. Software

Robotic arm is a very common configuration in which robots are usually made as they are good manipulators depending upon their Degrees Of Freedom. The term DOF represents the number of independent variables defining the motion of the robot whether translation or rotation or a combination of both. One such robotic arm is RHINO XR4 servo robot made by RHINO Robotics Ltd.[1] Here XR stands for E**X**PERIMENTAL **R**OBOTICS and 4 is its series number that means three robots from XR series have been made already and it is the fourth. It is a five axis robot having six motors serving as actuators for the robot. Here axis refers to the degrees of the freedom of the robot. The sixth motor is used for the opening and closing of the gripper. Figure1.1 shows RHINO XR4 servo robot.

Figure.1.1 RHINO XR4 Servo Robot:

Table 1.1 Specifications of RHINO XR4. [2]

Model	XR-4 Robot Arm
Applications	Education, training, research
Configuration	5 Axes plus gripper All axes completely independent All axes can be controlled simultaneously
Drives	Six PMDC servo motors with integral gearboxes and incremental optical encoders for real time closed loop operation
Controller	Mark VI controller from Rhino.
Payload	4.4 pounds (2 kilograms)
Speed (Gripper)	1 second to open; 2 seconds to close
Speed (Axes)	Programmable, 40 degrees per second
Repeatability	0.157 inches at full extension
Weight	25 pounds (11.4 kg) without base; base is 10 pounds (4.5 kg)
Reach	24 inches from center of waist to finger tips
Work Envelope	Motor "F" Body Rotation - 350 degrees Motor "E" Shoulder Rotation - 150 degrees Motor "D" Elbow Rotation - 180 degrees

	Motor "C" Wrist Rotation - 250 degrees Motor "B" Gripper Rotation - +/- 7 revolutions
Standard Gripper	1-1/4 inch (31.7 mm) opening for position 1 2-1/2 inch (63.5 mm) opening for position 2
End Effector Options	Long finger attachment Triple finger attachment Narrow finger attachment Clam shell finger attachment Shovel attachment Moto Dremel hand Rhino writer hand Vacuum finger attachment End effector package (includes above)
Accessories	Rotary carousel Tilting rotary carousel X-Y table Belt conveyor Slide base Experimental motor kits Support hardware kit
Software for PC	RoboTalk robot control language Kernel control language

1.2 MARK IV:

RHINO XR4 comes with its controller named MARK VI. Actually the MARK IV controller is capable of controlling either the XR-3, XR-4 or SCARA robot arm. It can be configured as a general purpose motor controller. Figure.1.2 shows MARK IV controller and it’s teach pendant .Some of the specifications of MARK IV are listed below in the table [3]

Fig.1.2 MARK IV: [4]

Table 1.2: Specifications of MARK IV[3] :

Model	Mark IV controller and teach pendant
Applications	Education, industrial training, research
Configuration	8 Port for optically encoded motors to operate the XR series and SCARA robots and accessories. Each port with full PID control. Two ports for auxiliary motors with full PWM capability.
Input/output	8 Input line pairs 8 Input switches 8 Output line pairs Full teach pendant support 48 Line pairs I/O module available (Optional)
Microprocessor	16 Bit main processor 8 Bit motor controller 8 Bit teach pendant controller
Communications	RS-232C, full handshake on both the host interface and the pendant interface
Compatibility	Can be run from any computer with an RS-232C interface.

Commands	Over 100 kernel commands form a comprehensive control language
Software	RoboTalk robot control language package available for the DOS/Windows family of computers.
Voltage	Available in either 120 Volts or 240 Volts, 50 or 60 Hertz, single phase
Dimensions	15" wide x 18" deep x 6" high
Weight	35 lbs.

<u>TEACH PENDANT</u>:

The teach pendant has its specific microprocessor and it communicates with the host computer through RS232C. The teach pendant is customary apparatus with the MARK IV controller. Under the control of RoboTalk for Windows, the teach pendant takes on sensational and stretched proficiencies, because you can interactively teach the robot a series of move. Figure.1.3. shows the teach pendant.

Fig.1.3 Teach pendant: [5]

MODES OF OPERATION[6]:

The MARK IV controller can operate either from a host computer or as a standalone unit with its teach pendant. These are referred as host mode and teach pendant mode respectively. While under the host mode the full power and flexibility of the controller is made available. In addition the teach pendant can be used as an input device providing a simple means to move the robot or motors without requiring any knowledge of the position of the robot or motors. Teach pendant mode provides a mean for stand-alone operation. No host computer is required. A program can be taught, executed and stored for later retrieval even after the system is turned off.

ASSIGNED TASK

Controller for RHINO XR4 servo robot MARK IV is not functional in the Department of Biomedical Engineering, NEDUET. It had been assigned to develop a custom controller for RHINO XR4 that could replace MARK IV functionally and would be equally capable with regard to operations and controls. Studying the assigned task thoroughly the challenge of developing a custom controller with only utilizing the mechanical structure of the robot was undertaken. Without the detailed help from the manufacturer of MARK IV, modular structure for the controller of RHINO XR4 was planned along with the development of user friendly and interactive Graphical User interface. The controller to be developed was planned to be more advanced and to have much more flexibility of operations then the previous controller, however it was planned to be a dedicated controller for RHINO XR4 and RHINO XR3, with the only limitation of auxiliary ports and number of I/O. The detailed information regarding the planned layout of carrying the task has been described in chapter 02.

PREVIOUS WORKS:

Literature review proved to be very helpful in the development of the custom controller since there are previous works in this regard. The work that we thoroughly reviewed was “An Educational Robotic Workstation based on the Rhino XR4 robot” [7]. This work described the development of a controller that was built to execute the tests on robotic manipulations. This workstation was built for RHNIO XR4. This workstation was developed for educational aspect of robotics hence it used the MATLAB environment for the interfacing and controlling the robot, since MATLAB is popular among the engineering students. Other reviewed literatures include [8] it defined a project developed at the University of Lleida, Spain. The aim of this project was to develop a computer vision system that must sense a tiny object kept arbitrarily on a target surface and control an educational robotic arm to grab it up and place it to a coded location. And [9] The chief job of this project was to enable the XR-4 work correctly from its earlier non-working state. There were also extra features added on the system. The added features were the sensing capabilities, where the objects that came across the conveyor can be captured in terms of its length and the process of rejection could be done. This project is very useful in mass production units. Also [10] This paper discourses the glitches with the RHINO educational robot original software interface that has made RHINO incapable of performing certain tasks. A novel software interface, built on a prevalent high-level structured language, had been introduced that made it an excellent teaching tool. This could be helpful for teaching purpose. Specifically the flexibility for inverse kinematics solution for RHINO XR4 was an additional task, original RHINO interface did not include that. Some more added features of the newly developed interface were clock and software library units, opened new ways for RHINO XR4 flexibility in control. Also [11] it described the flexible pick and place task performed using robotic arm with the aid of visual insight. Object manipulation using visual feedback require the knowledge of the orientation of the object with respect to the manipulator. For visual feedback a camera and a distance sensor was used. This programming system was modular and it was made up of various dynamic libraries that are independent of hardware and it offered a responsive graphic interface where the user could describe pick and place object localities on the image space. Also [12]A robotic workstation was built around RHINO XR4 upon which Students can perform experiments on kinematics, trajectory generation and motion control. An innovative controller was further designed to introduce motion control by visual feedback. These novel laboratory experiments enabled execution of various vision-based control arrangements and [13] were reviewed.

CHAPTER.02

ANTICIPATED/PLANNED SOLUTION

As chapter one described briefly the nature of the assigned task, it also described in detail the assigned task. Having no idea of how a robot can be controlled with a Graphical User Interface, the development of a custom controller was challenge in fact. Every innovation is just an idea in the beginning, so same was this case. This chapter describes in detail how a layout was set for the development of the custom controller, where was the work started from? And how did the goals were achieved? What problems were faced? And how they were resolved?

2.1 UNDERSTANDING THE NATURE OF THE ASSIGNED TASK

It was asked that whether a custom controller for RHINO XR4 could be developed, the group replied in assertion, as all of the group members were like minded and had the inclination towards instrumentation, controls and programming. A Line Following Robot had been developed by the group in the 6th semester and that experience with the Line Following Robot enhanced the group's desires to get some more hands on robotics and controls.

Work was started with the detailed examination of RHINO XR4 and its MARK IV controller. Firstly, RHINO XR4 was examined. Its mechanical structure was examined first and also its specification were searched on its official website. As the specifications were studied, but due to the lack of robotics background knowledge, its specification were not very well understood. But the examination was continued and along with it mechanical structure the actuators incorporated in it were also studied. Actuators of RHINO XR4 are PITTMAN Permanent Magnet Direct Current servo motors[14].

- ❖ Base, Shoulder, Arm and Wrist actuators are PITTMAN GM9413K046-R1 PMDC motors. These motors are named F through C respectively.

- Tool Roll and Tool Pitch motors are PITTMAN GM8712465 PMDC motors. These motors are named B and A respectively.

Table 2.1.1 Actuators Specification.

S.no	**Motor model**	**Voltages requirements**	**Gear ratio**
1.	PITTMAN GM9413K046-R1	12 Volts	65.5:1
2.	PITTMAN GM8712465	24 Volts	96:1

These PMDC servo motors are with integrated gearbox and optical encoders. Gear ratios have been mentioned in the table 2.1.1 above. The optical encoders are Quadrature encoders as the generate four logic levels and they have two output waveforms that are 90 degrees out of phase and the wave output of the optical encoders are shown below in the figure 2.1.1.

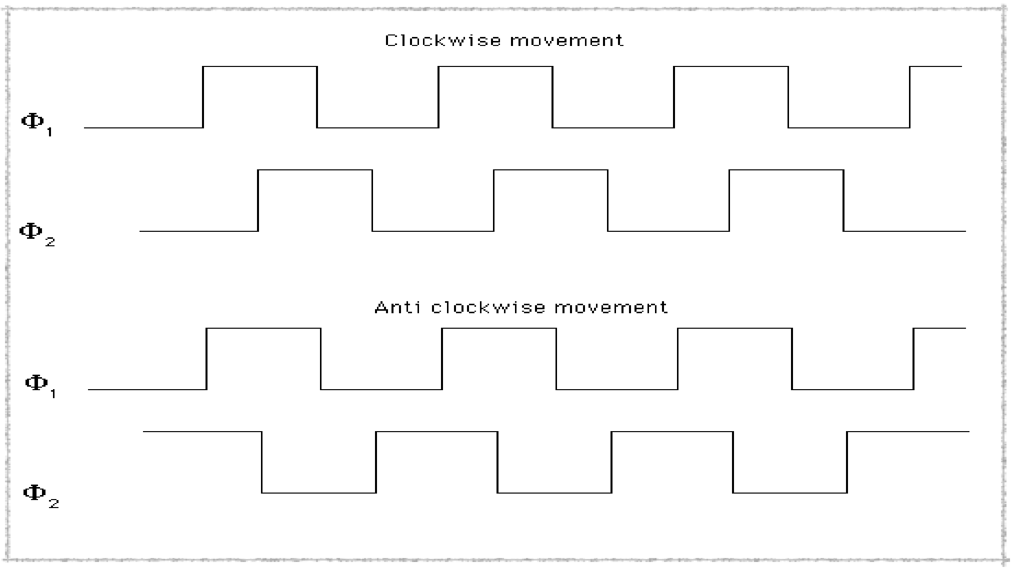


Fig.2.1.1 Output of the optical encoder

As can be seen in the fig. 2.1.1, the output waveforms are 90 degrees out of phase, this helped in to control of position and the direction of the motors as the encoders are incorporated on the shaft of the motors. There is a pair of Infra-Red transmitter receiver on the encoder circuit. The encoder disks are striped. Motor C through F have an encoder disk that has six black and six white stripes on it while encoder disks of motors A and B have an encoder disks that have three black and three white stripes on them.

There is an IDC connector connected to every motor on the circuitry integrated on the motor shaft. These IDC headers have ten pins in them and their pin description is given as follows;

1. TTL logic ground.
2. Encoder Channel A.
3. TTL +5Volts source.
4. Encoder Channel B.
5. Chassis ground.
6. Limit Switch (Active low).
7. Motor Power, Positive Terminal
8. Motor Power, Positive Terminal
9. Motor Power, Negative Terminal
10. Motor Power, Negative Terminal

These IDC headers are shown in the Fig.2.1.2 (a) and the pin numbering of IDC headers are shown in Fig.2.1.2 (b). A blur pictures in minds was forming that how the control of a PMDC servo motor will be done. The Limit switches of all the actuators were then observed, that where they are located and how do they work. It was founded that all the limit switches are active low and with these limit switches the HOME position of RHINO XR4 was determined.

After carefully examining the mechanical structure of RHINO various documentations related to it on the internet and provided in the form of its operating manuals, were thoroughly studied. The relevant information (motor specifications, RHINO XR4 specification.) regarding RHINO XR4, was collected that could be of help in developing its controller.

The MARK IV controller was examined. The hidden mechanism of MARK IV was explored. A huge population of Integrated Circuits dwelling inside MARK IV were observed. Transistors, opto-couplers, logic ICs, transformers, capacitors and resistors were founded inside the controller box. Knowing the functions of IC's in the controller box it was guessed that how MARK IV could have controlled RHINO XR4. The modules in MARK IV were identified and the function of each module and the function of every component in the module through the datasheets were determined. The specifications of MARK VI that have also been listed in chapter 01 were also thoroughly studied.

After going through all the relevant documentations the literature review was done that is presented in detail in chapter 01. Literature review helped in refining the sketches of what we really wanted to do. Specially[7] , this work enlighten the vision and encouraged us to do our level best. It was reviewed thoroughly and understood the way they had worked.

After observing the Mechanical structure of RHINO XR4, collecting its relevant specifications, studying MARK VI, reading operating manuals and going through the literature review it was decided what to do and how to do.

2.2 PLANNED PROPOSAL OF THE ASSIGNED TASK

We were assigned a task of developing a custom controller for RHINO XR4. Everything was planned and the weight every aspect of the development was done so that the work could meet the wanted or desired requirements. Work was broken down and divided into various parts and every possibility of doing a particular part of our work was determined. It was divided into the following parts;

1. Learning Robotics.
2. Selection of hardware.
3. Learning the development of Graphical User Interface.
4. Learning code writing.
5. Learning hardware software interface.

6. Estimating cost.
7. Designing the final casing of the.

2.2.1 Learning Robotics

The responsibility of Putting RHINO XR4 back to work was assigned to us. As RHINO XR4 is a multiple degree of freedom robot so the first and the foremost thing that had to done was to understand the basics of robotics. Learning robotics on one's own was a difficult task itself. Saeed B. Niku, it's basically the name of the writer whose book is prescribed usually as a textbook, proved to be very helpful. It helped us a lot as we were beginners in robotics. Tutorials of Mr. Osama Khatib were also watched. Mr. Osama Mazhar also helped us in the process of learning robotics. The basics of robotics was learnt and implemented it on RHINO XR4. Robotics terminology became familiar. The forward kinematics of RHINO XR4 was also understood and developed. Detailed explanation of the forward kinematics of RHINO XR4 will be presented in the later chapter, but here it is a brief list of the steps of the development of forward kinematics of RHINO XR4;

- ❖ Finding the D-H parameters of RHINO XR4.
- ❖ Forming Homogenous Transformation Matrices for each frame.
- ❖ Finding the Forward Kinematic equations of RHINO XR4.

RoboAnalyzer is Software for developing the 3D models of robots by using its D-H parameters. It is a very useful software for learning the basics of Robotics. It was used for learning Robotics. Further details of RoboAnalyzer and its utilization in the task will be described in the latter chapter. After learning robotics a more clear idea was there that what to do. The forward kinematics equations have been developed, now it was the turn that these equations are utilized and verified. The need of developing some Graphical User interface that could hold the forward kinematics of RHINO XR4 was identified and we could provide only a joint variable 'θ'.

2.2.2 Selection of Hardware

For turning the ideas into reality it was needed to get some hardware, assembled it and tried to control the robot. To design a custom controller, one must had an idea of developing standalone controller that could control all the actuators of RHINO XR4 precisely one at a time or more than one. The hardware requirements were listed and they were;

- Controller module.
- Motor Driver module.
- PCB (Printed Circuit Board) to in-corporate Controller and the motor driver module.
- Power Supply module.
- System for incorporating GUI and the interfacing between system and the controller.

- *Controller module:* It was founded that what had to control and how it would be controlled, this made the selection of the controller easier. What were those selecting parameters will be identified later in the chapter named Hardware Integrated. Arduino Mega was selected as it was according to the needs, cost effective, easily available and with enormous support on the internet.
- *Motor driver module:* MPMDC servo motors can be controlled easily with an ordinary motor driver like L298. But for efficient motor control one must had some robust motor drivers. There were two options Four Channels Motor control unit and Polulu MC 33926 dual motor driver shields. One could have more choices but one must have to consider the availability and the cost as well. As mentioned above that what were the selecting measures will be discussed later. Four Channels Motor control unit was tested first and then finally Polulu MC 33926 dual motor driver shields were utilized.
- *PCB (Printed Circuit Board):* A PCB had to be developed for all the modules of our controller. Proteus is very popular among engineering students, it is a software

for designing and simulating circuits. PCB was design and developed and finally integrated controller module and motor driver modules on the PCB.

- *Power Supply module:* During the examination of MARK IV and RHINO XR4 it was founded that what are the power requirements of the custom controller. Two different power supplies for the controller as it has been mentioned in table 2.1.1 that there are two types of motors and they have different power requirements as well.
- *System for incorporating GUI and the interfacing between system and the controller:* A system was needed that could be interfaced with the controller. Laptop is used for this purpose. The system had GUI in it and it was communicating serially with the micro controller.

2.2.3 Learning the Development of Graphical User Interface

With the purpose of making the custom controller more user friendly and interactive it was decided to develop a Graphical User Interface. Developing the GUI was also tried with MATLAB, Visual Studios and LabVIEW. But finally it was done using LabVIEW. It was decided to develop the GUI that would have following controls and indicators in it;

1. Individual actuator position control.
2. Individual actuator speed control.
3. Multiple actuators speed and position control simultaneously.
4. A 3D simulator displaying the movements of RHINO XR4 as the joint variable is given as input.
5. Homogenous transformation matrix displaying the final position and orientation of the End Effector.
6. Home Positon control.
7. Start and Stop controls.

The GUI would have forward kinematics Equations in it. One only have to input the joint variable in Degrees. The given input will then be processed by the system and converted into the equivalent pulses and sent to the controller through the USB cable.

Those pulses in turn would be sent to the motor driver shield, finally the motor driver will provide PWM to the actuators and eventually the actuators will move according to the given input. There would also be a closed loop control that will be described in the later chapter. The Graphical programming of the LabVIEW for the development of GUI had to be learnt and for the serial communication between the controller and the system having GUI.

2.2.4 Learning Code Writing

Controlling a robot with six joints, means controlling the six actuators simultaneously is really a big deal. There would be hardcore coding for the control of these actuators. It was aimed to achieve the home position first, for this one had to incorporate the limit switches and each encoder’s two output signal in the form of pulse train in the code. One had to control the PWM and direction as well. Moreover the code for Proportional-Integral-Derivative (PID), for the smoother and accurate operation had to be written. It was not done yet, one had to write code for the serial communication between Arduino Mega and the system incorporating the GUI, this was supposed to be very difficult and it proved the same!

2.2.5 Learning Software-Hardware Interfacing

The interfacing of hardware and the software was also an important task. One had to consider the power requirements and the hazards relating to the power, secondly one had to learn the architecture that helps in interfacing between the controller and the GUI. The details of interfacing between hardware and software will be discussed in detail in a dedicated chapter on this topic.

2.2.6 Checking Repeatability, Precision, Verification Of Forward Kinematics

Repeatability and precision are most demanding features of any robot. One had to test them. One had to achieve the desired accuracy. For this one had to verify the forward kinematics of RHINO XR4 using the forward kinematics equations and the simulation software RoboAnalyzer.

CHAPTER NO. 03

HARDWARE INTEGRATED.

3.1 INTRODUCTION

Chapter one described the specifications of RHINO XR4 and the MARK IV that helped to select the hardware required for the development of the custom controller. Also the hardware requirements of the controller to be developed and the robot were studied through the Mechanical structure of the robot that incorporates its actuators. Mainly the power requirements of the actuators is the basis for the selection of the components of the power module and the motor driver modules. This chapter describes the selection criteria for the hardware integrated in the development of the custom controller it also discusses the advantages of using them and their limitations.

3.2 POWER MODULE

Table 2.1 describes the power requirements of the two different actuators of the RHINO XR4. From that table only the power ratings i.e. operating voltages can be determine but the details are acquired from the data sheets of these actuators[14]. One had to select the power module considering the worst case scenario. One must specially consider the peak stall currents of both the actuators and searched for the most appropriate power module that could fulfil the power requirements of RHINO XR4 and the controller to be developed. Two different power supplies with the ratings as follows were incorporated;

- Power module for powering the motors C through F rated as 12 Volts and 15 Amperes
- Power module for powering the motors A and B rated as 24 Volts and 10 Amperes

These power supplies forming the power module of the controller are compact and they dissipate less heat as they do not get heated for even longer operations. But considering the worst case scenario we have installed a cooling fan as well in the controller casing.

3.3 MOTOR DRIVER MODULE

Actuators are just like our muscles, as our muscles need proper nutrition for their smooth functioning and they also require someone to drive them likewise the actuators of the robot need appropriate power input and proper controlling. One of their requirements was fulfilled but the other remained to be done. Considering the power requirements and the incremental encoders attached to their shaft that generates the pulse train, and the limit switches for every joint and the gripper, one had to select a motor driver that could be in line with our requirements. Two different motor drivers were tested one was Dagu 4 Channel 5-12V, 2A Brushed DC Motor Controller it was capable of driving four 12 volts motors simultaneously while providing 4A stall for each channel. This board integrates Low resistance FET H-bridges. Anyone who has studied Basic Electronics course must be familiar with the name of H-bridge. FET H-bridges are used for driving the actuators of RHINO XR4 because they are capable of controlling inductive load up to high current and they can Pulse Width modulate the loads up to very high frequencies, precisely up to 20KHz. Usually for Permanent Magnet Direct Current motors high power MOSFETS are used. Dagu 4 channel motor drivers didn’t live up to the requirements and it became useless. Then the next option was tried for the motor driver module which was Pololu Dual MC33926 Motor Driver Shield Figure 3.1[15] . It proved to be the most appropriate match for driving the actuators of RHINO XR4. It is basically a dual motor driver shield it has a wide operating range i.e. 5-28 Volts. It provide 3Ampere current for each channel with 5A peak stall. Its input are compatible with a 5V or a 3.3V system like the controller Arduino Mega 2560. It can Pulse Width Modulate up to 20 KHz.

Fig.3.1 Pololu MC33926 Dual Motor Driver: [15]

3.4 <u>CONTROLLER</u>

Considering all the necessities of controlling the actuators of RHINO XR4, a controller was selected. Firstly the controlling was done with Arduino UNO [16] for the two actuators. It was found that Arduino was completely capable for fulfilling the requirements of controlling the actuators. Incremental encoder mounted on the shaft of every motor are quadrature encoders and they have two output signals namely A and B, generate pulses that were read using the Arduino. Interrupt Service Routine (ISR) algorithm was used to count the pulses from the encoders. There are basically two ways of doing this, one is the Polling and the other is Interrupt Service Routine. You may say that the Polling is the continuous monitoring of any device by the controller and the ISR is the servicing by the controller upon request by the device. Once it was found that Arduino UNO can efficiently control the motors i.e. their position, direction and speed in collaboration with the motor driver shields, it was tested for the serial communication. Chapter 05 describes the interfacing of hardware and the software in detail. After finding that Arduino can communicate serially according to the requirements of the project, it was decided to move on to Arduino Mega 2560 Figure 3.2[17]. The reason for moving towards Arduino Mega are that it has more Digital I/O pins, Analog pins and more interrupts than Arduino UNO. In fact Arduino Mega is the most appropriate choice for the controller of the project, as it has six interrupts one for serving each of the two channels of incremental encoders. But it imparts a limitation to the designed controller in regard to the number of encoded motors controlled by the controller. The developed controller will be a dedicated controller for RHINO XR-4 and RHINO XR-3.

Arduino Mega has been successfully tested for;

- Simultaneously reading encoder counts from every axis motor.
- Simultaneously monitoring the limit switches for each joint.
- Sending and receiving data serially at the baud rate of 115200bps, from the Arduino Mega to the operating system and from the operating system to the controller through the USB port.
- Writing and executing the code for HOME position of RHINO XR4.

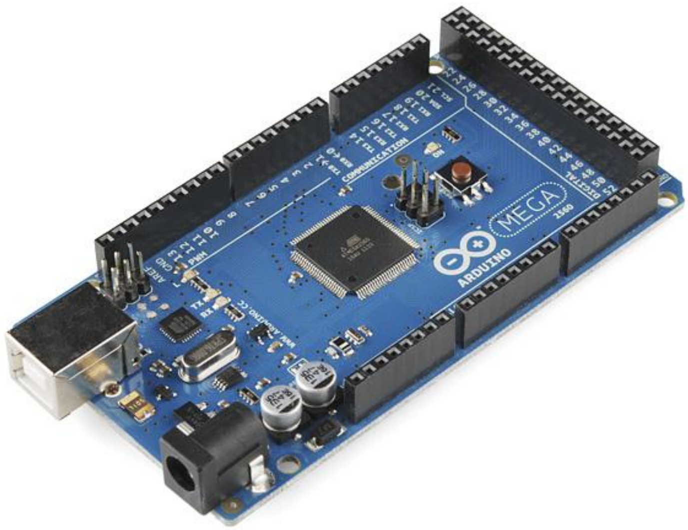


Fig. 3.2 Arduino Mega 2560 [18]

3.5 THE OPERATING SYSTEM:

With the objectives of making the controller advanced, efficient, and interactive and user friendly, the Graphical User Interface has been developed. Its development will be described in Chapter 04 along with its features. The operating system with Windows 7 houses the GUI. The GUI is deliberately developed for the interactive controlling of the robot. It is serving as the teach pendant in this case, as the controls are there in the GUI for controlling individual motors and more than one at a time. The Operating System along with its GUI will be helpful for students in learning the basics of Robotics

3.6 WORKING OF THE CONTROLLER WITH ITS GUI:

As the controller is turned on RHINO XR4 comes to its HOME position wherever it would be. Once it is at its HOME position now it can be used for manipulation tasks. For example it is desired to move the motor F through 30 degrees, how this is programmed and developed it is explained below;

- The pulses generated during the complete motion of a particular joint which is 350 degrees for motor F.
- Pulses are counted and then normalized with their respective movements in degrees.
- The GUI controls take inputs in degrees.
- The input angle in degrees is then converted into the pulses.
- These pulses are serially transported through the USB port to the Arduino Mega.
- Arduino Mega then forward these pulses to the motor driver shields.
- Motor driver shields then generate the PWM accordingly.
- The generated PWM is then transferred through the IDC cables to the motor.
- Motor in turn move to the proportional number of degrees.
- Closed loop mechanism is implemented for efficient control.
- As the motor moves it generates the pulses accordingly.
- These pulses through IDC cable are sent to the controller and are being monitored by ISR.

- As soon as the desired position is reached the controller stops the motor driver shields from supplying more PWM.

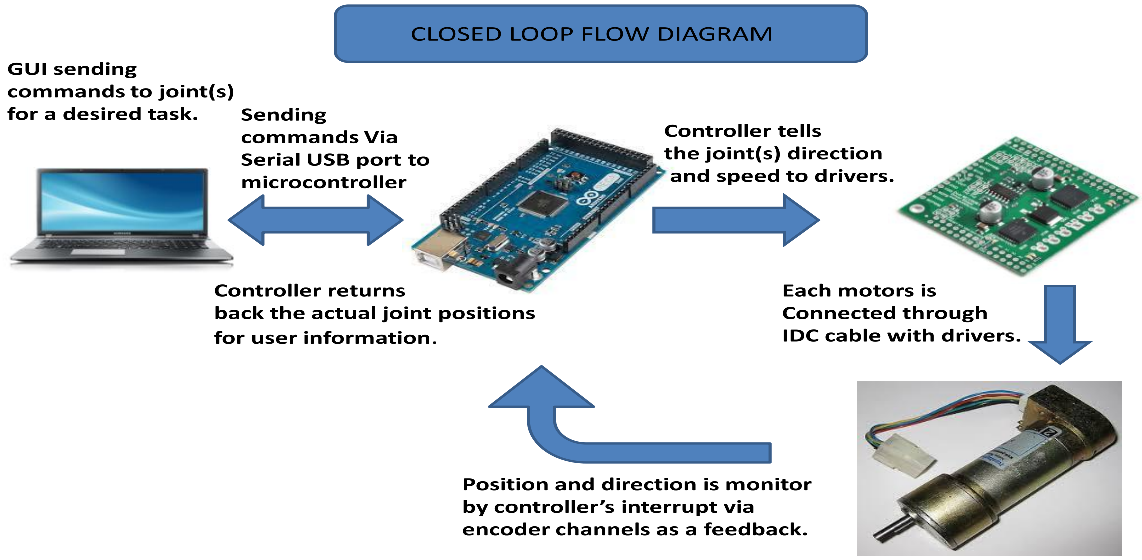
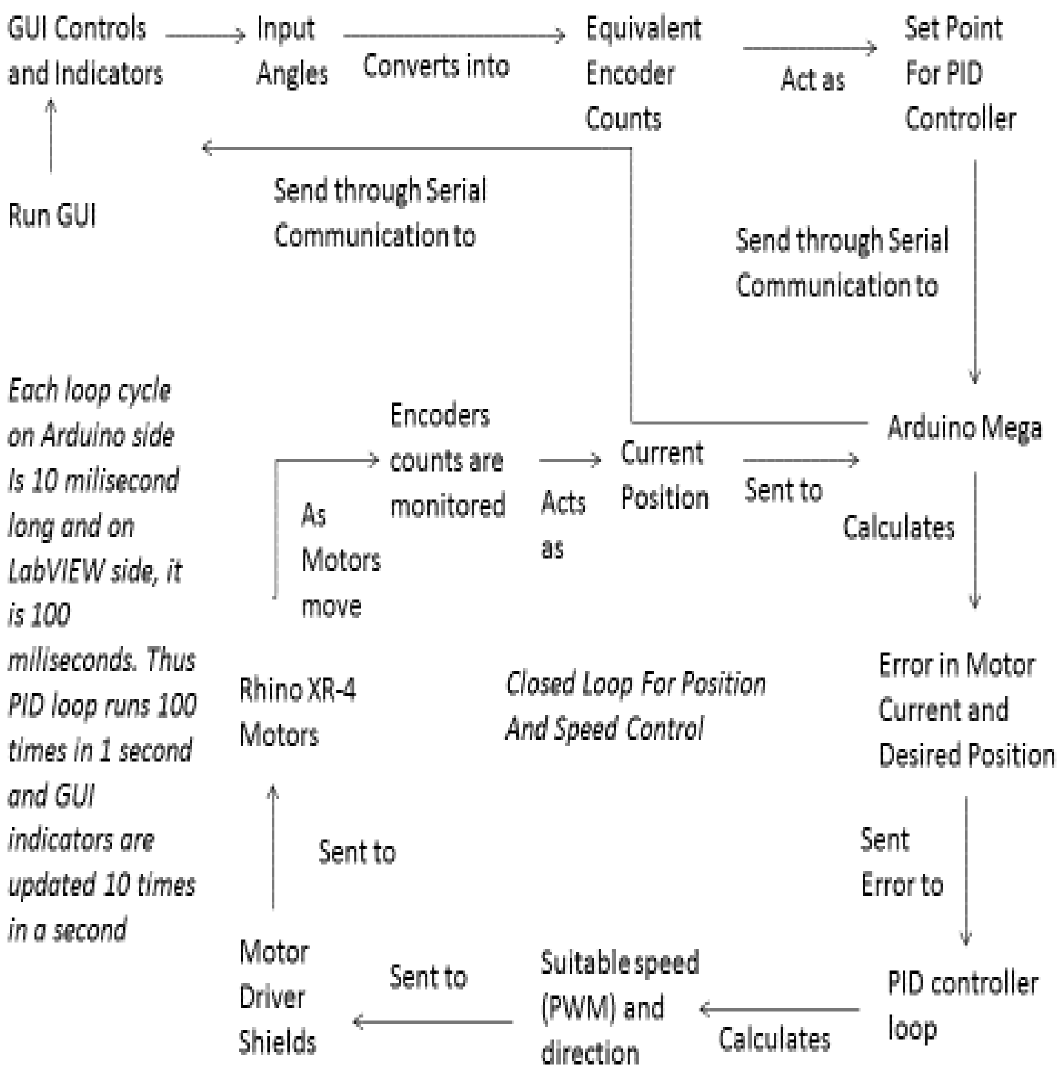


Figure 3.3 Flow Diagram


GUI Controls and Indicators
Input Angles
Converts into
Equivalent Encoder Counts
Act as
Set Point For PID Controller
Run GUI
Send through Serial Communication to
Send through Serial Communication to
Each loop cycle on Arduino side Is 10 milisecond long and on LabVIEW side, it is 100 miliseconds. Thus PID loop runs 100 times in 1 second and GUI indicators are updated 10 times in a second
Encoders counts are monitored
As Motors move
Acts as
Current Position
Sent to
Arduino Mega
Calculates
Rhino XR-4 Motors
Closed Loop For Position And Speed Control
Error in Motor Current and Desired Position
Sent to
Sent Error to
Motor Driver Shields
Sent to
Suitable speed (PWM) and direction
Calculates
PID controller loop

CHAPTER NO 04

SOFTWARES UTILIZED

4.1 INTRODUCTION

Last chapter provided the detailed description of the hardware integrated in the custom controller, their selection criteria, applications in the project, advantages and limitations. This chapter describes the software utilized in the development of the custom controller, particularly their usage in this development, their advantages and their limitations.

4.2 SOFTWARE EMPLOYED

Software remained the crucial element for the development of the custom controller. Software were used for the development of the Graphical User Interface, programming, development of the Printed Circuit Board, Developing transfer functions and Simulation. These software are;

1. LabVIEW 2013.
2. Arduino IDE 1.6.4
3. Proteus 7.
4. MATLAB 2012b
5. RoboAnalyzer.

4.2.1 LabVIEW:

LabVIEW stands for Laboratory Virtual Instrumentation Engineering Workbench. It is basically a system design platform and development environment for a visual programming language by National Instruments [12]. LabVIEW is not that popular among students of the department of Biomedical but it is utilized for the development of the Graphical User Interface. LabVIEW was chosen for the development of the GUI due to the following reasons;

- LabVIEW uses Graphical Programming which is easy to learn but not that easy.
- Dragging, dropping and proper wiring is just needed for the development.
- It is more attractive and interactive
- It has a large support on the Internet for development and interfacing it with controllers.
- It has numerous flexibilities for the development

Fig. 4.1 LabVIEW

LabVIEW had to be learnt for this development, video tutorials, LabVIEW MakerHub[19] LabVIEW help were utilized. The GUI developed using the LabVIEW Figure 4.1a and Figure 4.1b, has the following Controls and Indicators;

1. Direct control of individual motor.
2. Co-ordinated control of multiple motors
3. Soft Home Control.
4. 3D Simulator.
5. Speed Control.
6. Stop control.
7. Homogenous Transformation Matrix Indicator.

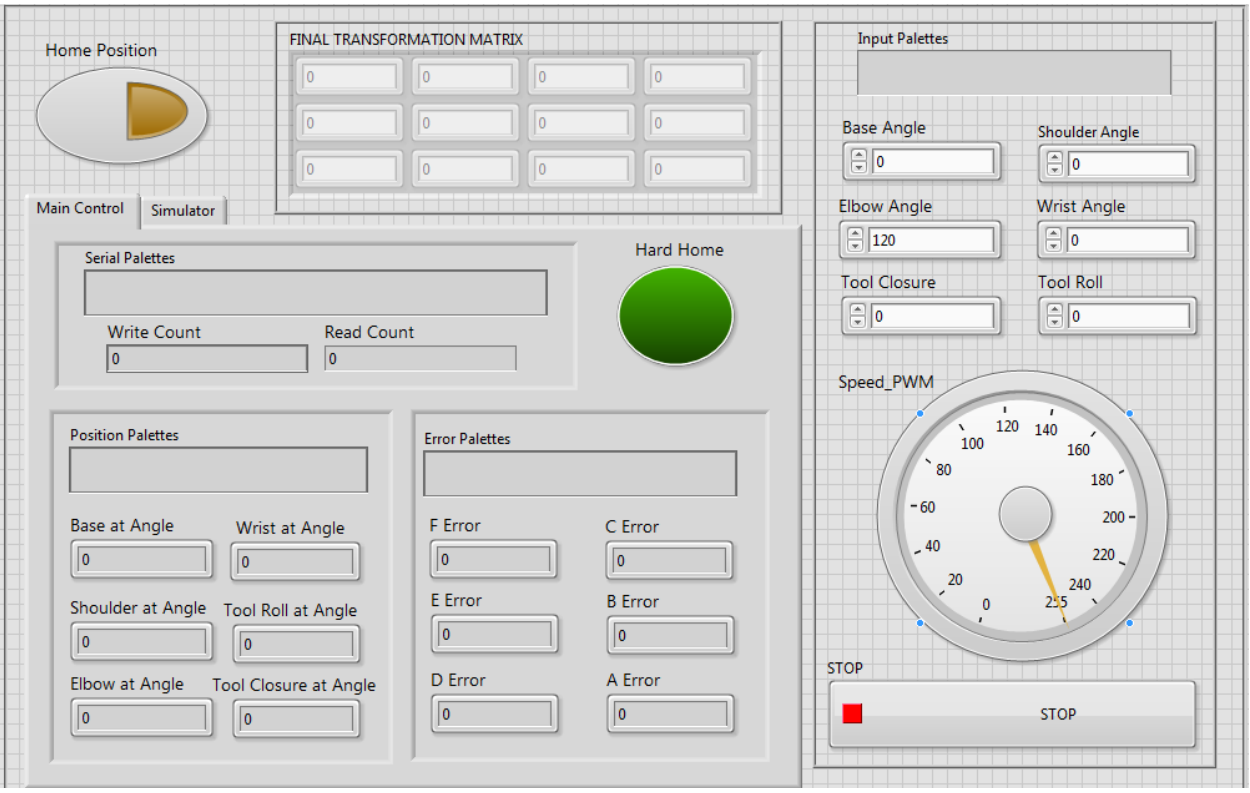


Fig 4.1a GUI with its controls and indicators:

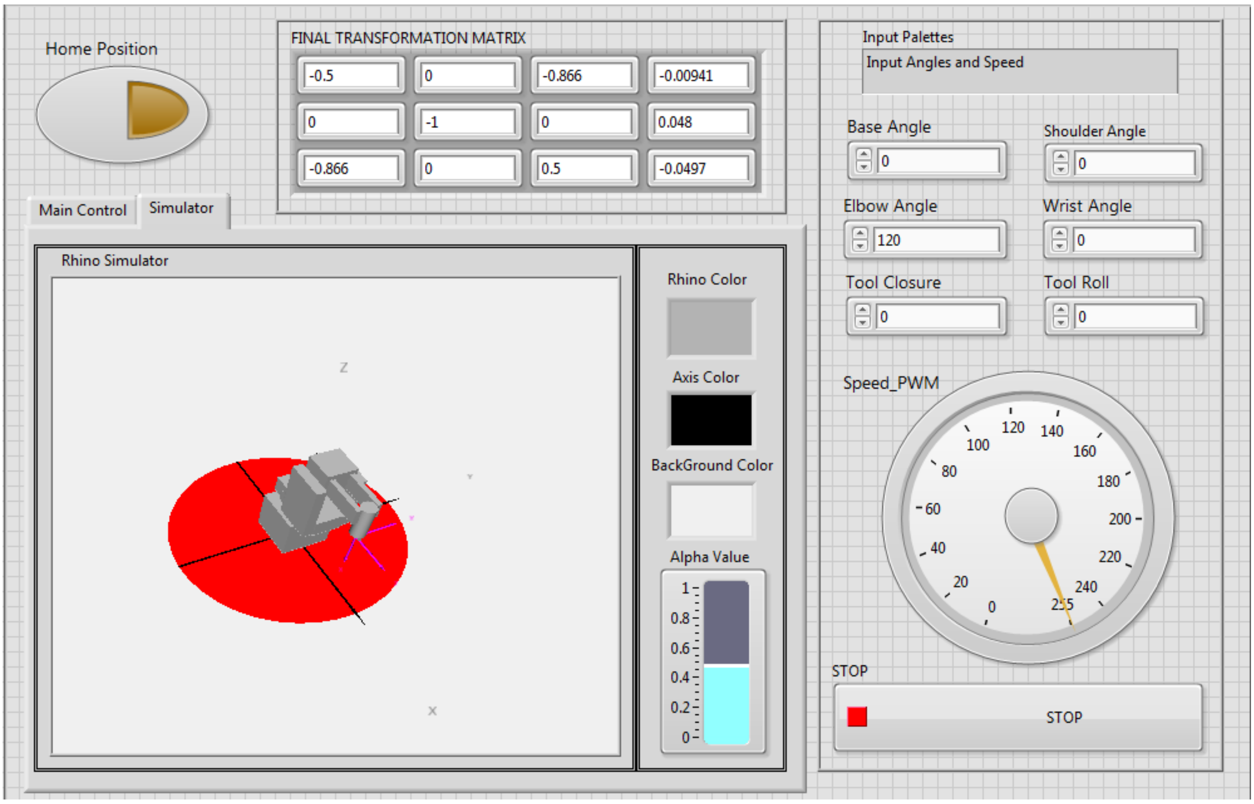


Fig 4.1b GUI with its simulator:

Controls in the LabVIEW are for the purpose of controlling some parameters and Indicators are for indicating the values of some parameters.

4.2.1.1 Functions of Various Controls in the GUI:

I. Motor Controls; they are for controlling individual motor and more than one at a time. Controlling in the developed GUI has been made easy, the user just have to input the number of degrees for a particular motor to be moved. Alternatively this can be done with the arrow keys. Each motor can be moved to a minimum of one degree precisely. For controlling multiple motors at a time user has to input the desired angles in the respective input box and then hit the RUN button. The coordinated motion for the motors can be repeated as well by just pressing the HOME button in the GUI.

II. Speed Control; this is for the purpose of controlling the speed of all the motors simultaneously, this basically does Pulse Width Modulation. The dial has values from 0 to 255.

III. Stop Control; as the name indicates stops the running programming.

IV. Error Indicator; it indicates the difference between the desired and the actual position. It eventually turns to zero or a number near it when the desired position is reached.

V. Serial Communication Indicator; it indicates the number of data byes, sent by the controller as the Write count and the number of data bytes received by the controller as the Read count.

VI. Homogenous Transformation Matrix Indicator; it indicates the orientation and position of the end effector and it is continuously updated as the actuators move. This HTM Indicator is developed by employing forward kinematics equations of RHINO XR4.

VII. 3D Simulator; a 3D model of RHINO XR4 is displayed in the simulator and this model move synchronously with the real robot. A frame is attached for the better understanding the transformations.

This GUI can also be used for the verification of forward kinematics and as the HTM tells the end effector configuration, the position of the end effector can be measured and then can be verified from the HTM. GUI needs much more to be incorporated.

The interfacing between GUI and the controller is done using VISA, a chapter is dedicated for the description of the interfacing of GUI with Arduino Mega

.

4.2.2 Arduino IDE 1.6.4:

Having the previous exposure of programming with C language, a programming environment was needed that uses the syntax of C language, as Arduino Mega has been used as a controller and its programming is done with its own Integrated Development Environment so one of the major reason for choosing Arduino was the ease for programming.

Fig.4.2 Arduino IDE

Arduino IDE has been utilized for writing all the programs that are to be stored in the controller. Some are;

a. Encoder counts.
b. Home Position.
c. Serial Communication.
d. Movement of a single motor.
e. PID

The baud rate for the serial communication was set to be 115200 bps. At this rate serial communication was done smoothly.

4.2.3 Proteus 7

It was needed that all the hardware components and modules must be on a circuit board. Proteus [20] is a software for simulation of circuits and designing PCB. A PCB was designed housing all the modules of the controller. In fact a PCB was designed in an efficient way that has reduced the size of the controller. Figure 4.4a b and c shows the designed PCB using Proteus.

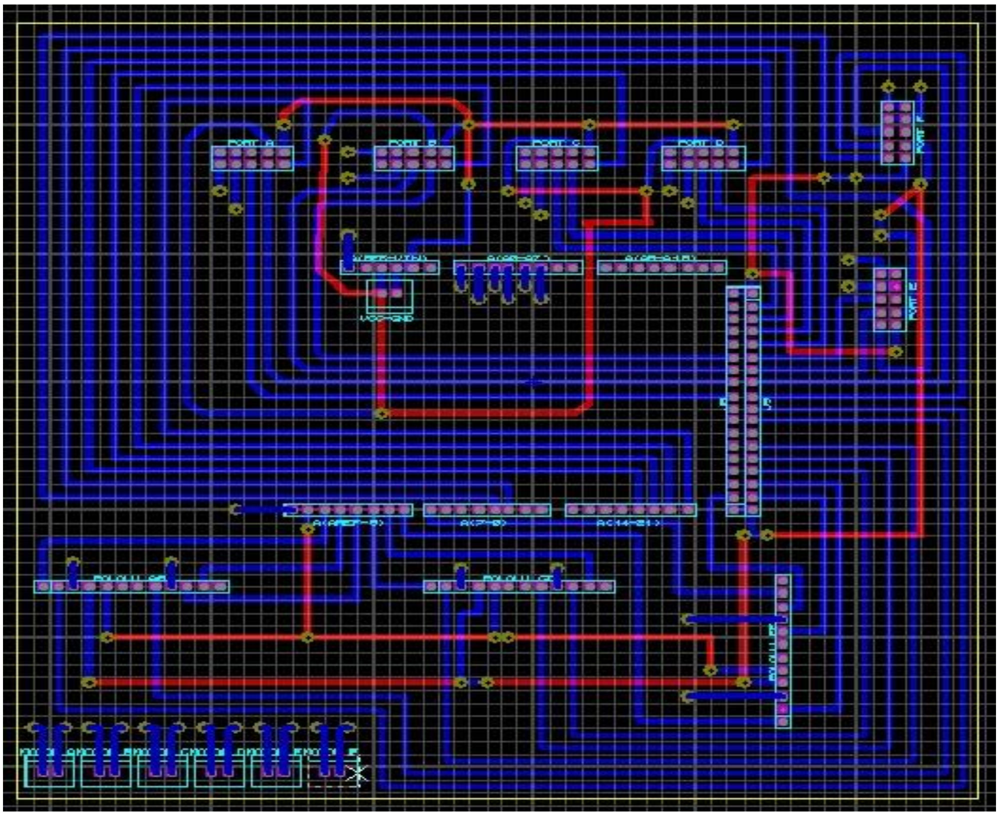


Fig 4.3a PCB layout:

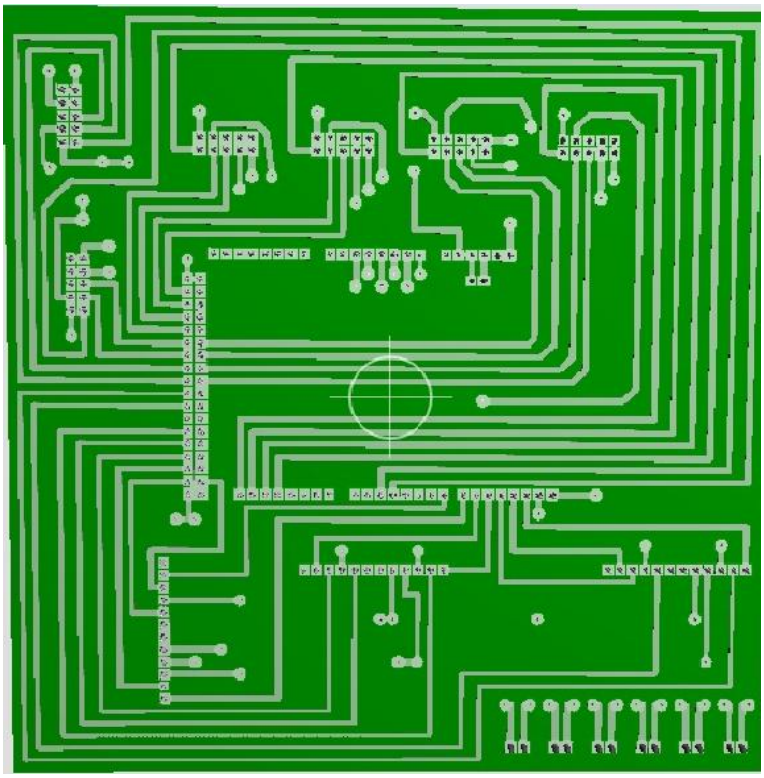


Fig.4.3b PCB Layout:

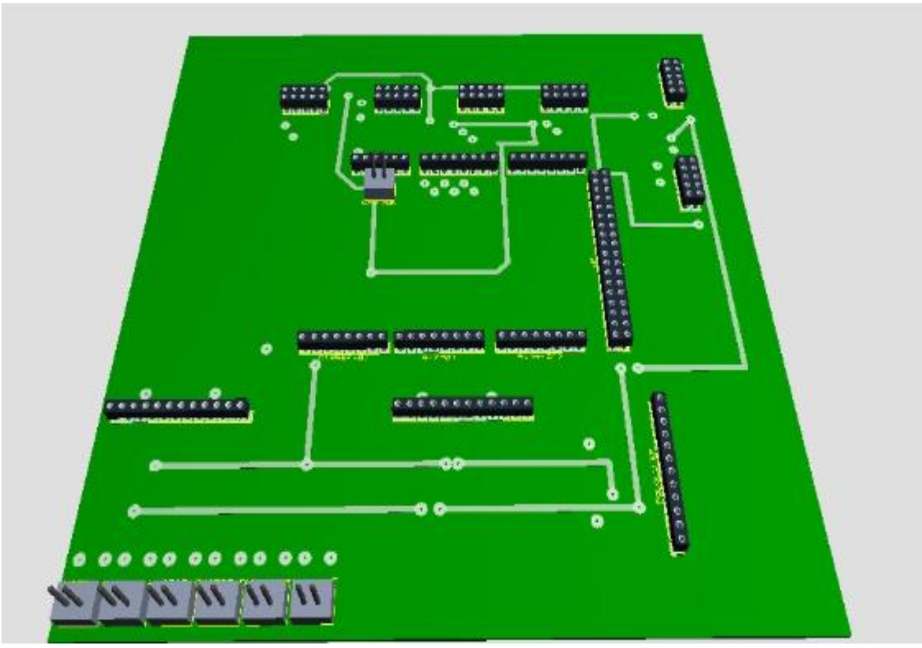


Fig.4.3c PCB 3D layout:

4.2.4 <u>MATLAB</u>:

MATLAB requires no introduction every engineering student is familiar with the MATLAB environment. It was used to form the transfer functions of the motors to find the Kp, Kd and Ki values for the purpose of writing PID (Proportional, Integral and Derivative). In the early stage of the development MATLAB Robotics Module was utilized for working on the forward kinematics and inverse kinematics of RHINO XR4 but later on, it was done with LabVIEW due to the flexibility of programming. MATLAB was also utilized for the serial interfacing of Arduino.

4.2.5 <u>RoboAnalyzer:</u>

Robotics is difficult to understand without 3D visualization. There is a freeware named RoboAnalyzer[21] that aids the understanding of students in Robotics. It was utilized for the verification of the Forward Kinematics of RHINO XR4. Using the D-H parameters (Denavit Hartenberg) of RHINO XR4 a simulator was built. It was then simulated for various inputs of joint variables. Figure 4.4 shows the simulator built using the actual parameters of the robot.

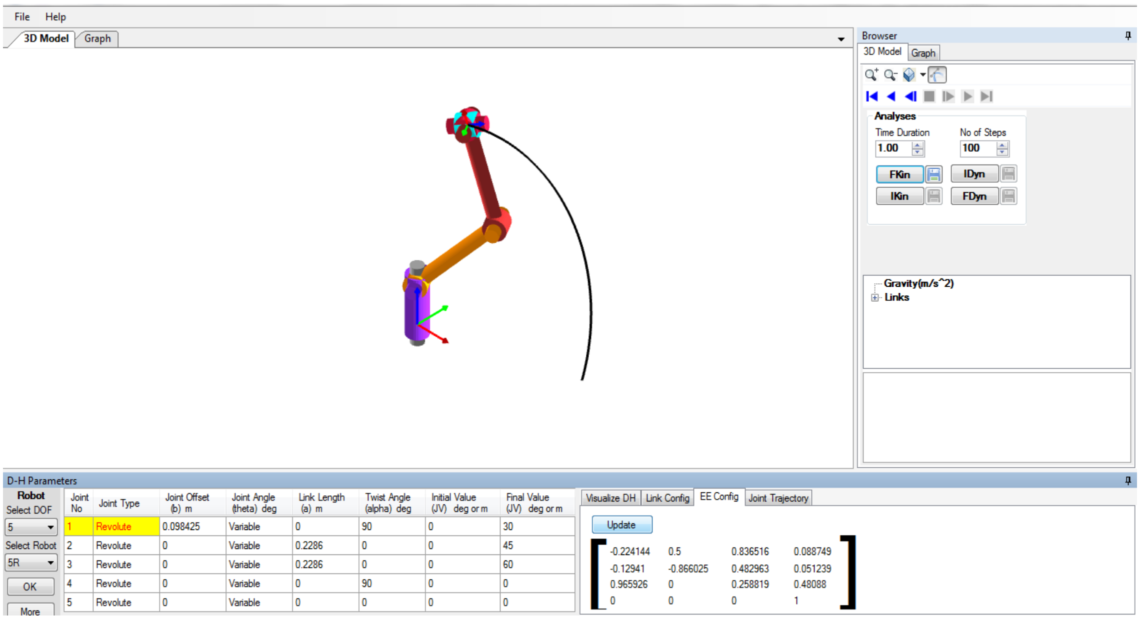


Fig. 4.4 Simulator of RHINO XR4 built using RoboAnalyzer.

After running the simulation for the forward kinematics of the robot, the position of the end effector was noted and it was then verified by measuring the actual position of the end effector using the measuring tape. Results obtained from the simulation in RoboAnalyzer were then compared with the one in the GUI of the LabVIEW and MATLAB. These measurements are repeated time and again for various input of joint variables.

CHAPTER NO. 5

HARDWARE SOFTWARE INTERFACE

5.1 INTRODUCTION

Chapter 03 and Chapter 04 described in detail the hardware and the software employed in the development of the custom controller. This chapter will illustrate the hardware-software interface in detail. As this interfacing is crucial for the GUI to be used effectively.

5.2 INTERFACING OF ARDUINO MEGA WITH THE LABVIEW

Interfacing between the controller and the operating system is the key of controlling the movement of the robot through the GUI in the operating system. There must be a protocol that houses the synchronization procedure. There are some software architectures that provide the protocols for the serial communication between Arduino and LabVIEW. They are listed below;

1. LINX
2. FIRMATA
3. VISA [22]

5.2.1 LINX

LINX delivers easy to use LabVIEW (VIs) for interacting with common implanted platforms like Arduino, “myRIO” and “chipKIT”. It Uses the built-in sensor VIs to start getting data to the PC in seconds or uses the peripheral VIs to access your devices "digital I/O, analog I/O, SPI, I2C, UART, PWM and more".

5.2.2 FIRMATA

The **Firmata** library kits the Firmata Protocol for collaborating with software on the host computer. This permits you to inscribe custom firmware without having to create one own protocols and objects for the programming environment that is being used.

5.3. VISA

VISA The Virtual Instrument Software Architecture (VISA) is a customary for organizing, encoding, and troubleshooting instrumentation systems involving GPIB, VXI, PXI, Serial, Ethernet, and/or USB interfaces. It delivers the programming interface between the hardware and development environments such as LabVIEW, Lab Windows/CVI, and Measurement Studio for Microsoft Visual Studio. NI-VISA is the National Instruments employment of the VISA I/O standard. It includes software libraries, collaborative utilities such as NI I/O Trace and the VISA Interactive Control, and conformation programs through Measurement & Automation Explorer for all the development needs. VISA is the most appropriate protocol for the hardware interface as it has the ability to evolve with time.

5.3.1 Benefits Of VISA:

One of VISA's benefits is that it uses numerous of the similar operations to link with instruments irrespective of the interface kind. For instance, the VISA command to code an ASCII string to a message-based instrument is the identical whether the instrument is "Serial, GPIB, or USB". Thus, it offers interface freedom. This can make it easy to shift interfaces and also provides the users who must program gadgets for dissimilar interfaces a lone language they can learn.

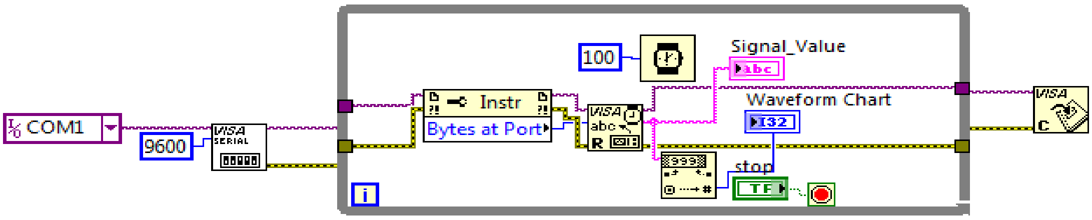


Fig 5.1 Sample VISA Serial Communication Program

It is also developed so that programs written using VISA function calls are simply transferrable from one platform to another. VISA performs this by using its own data types. This inhibits the problem of moving from one platform to the other. In other words, a LabVIEW application written with VISA commands can be simply ported to another platform that supports LabVIEW. VISA support various operating system.

VISA's utmost gain is that it is a tremendously easy language to learn. VISA delivers a very simple-to-use API that has bus autonomous purposes for most of its I/O functionality. It delivers the most frequently used functionality for instrumentation in a solid command set, eradicating the necessity to learn low level communication protocols for numerous bus categories.

5.3.2 VISA Vocabulary:

The most significant objects in the VISA language are recognized as resources. “A VISA Resource is any instrument in your system (this includes serial and parallel ports)”.

Synonym for resource is VISA descriptor. It states the interface kind “(GPIB, serial, USB)”. “A VISA session is a path of communication to a VISA Resource”, so one must open a VISA Session any time want to do VISA communication to an instrument.

5.3.3 A Typical VISA Application:

A typical VISA application runs with the following steps:

1. Open a Session to a given Resource.
2. Do any configuration on the given resource (setting baud rates, termination character, etc...).
3. Perform writes and reads to the device.
4. Close the Session to the Resource.
5. Handle any errors that may have occurred.

“This exact same format would be used in a text based languages, or any of the other buses VISA supports. All you would have to change is the Instrument Descriptor connected to the VISA Open. This code would run on any operating system that supports LabVIEW and NI-VISA.”

5.4 SUMMARY

NI-VISA is a software API that significantly decreases the development time of test and measurement systems. It provides developers the skill to effortlessly develop code to communicate with any instrument, over any bus, on most operating systems in use today. It also permits the generation of code that can be relocated from one platform to another or from one bus type to another with minute or no overhead in the change. This opens the door for producing “hybrid systems” that create the finest and most cost effective use of multiple bus types all used in one test system.

NI-VISA is also reserved up to date with innovative technologies so any test system developed with NI-VISA will be capable to make use of the latest and greatest bus technologies existing, while still leveraging earlier technologies to deliver a comprehensive communication protocol for today's test and measurement system.

All of these software architectures have been tried. But VISA proved to be the most appropriate.

CHAPTER NO 06

CLOSED LOOP CONTROL

6.1 INTRODUCTION

The drawbacks of open-loop systems, namely sensitivity to disorders and failure to correct for these disorders, may be overcome in closed-loop systems. A closed-loop control has been executed while the custom controller for RHINO XR4 was being developed. This chapter describes in detail the closed-loop mechanism for the custom controller.

6.2 PID CONTROLLER

Proportional-Integral-Derivative (PID) control is the most common control algorithm used in industry and has been unanimously acknowledged in industrial control. The admiration of PID controllers can be accredited partially to their stout performance in a broad range of operating situations and partially to their purposeful ease, which lets engineers to control them in a simple, direct manner. As the name proposes, PID algorithm involves of three basic coefficients; proportional, integral and derivative which are diverse to acquire optimum response.

The basic idea after a PID controller is to read a sensor, then calculate the wanted actuator output by computing proportional, integral, and derivative responses and adding those three components to compute the output.

6.3 CLOSED LOOP SYSTEM

In a representative control system, the process variable is the system factor that requests to be controlled. A sensor is utilized to measure the process variable and deliver feedback to the control system. “The set point is the desired or command value for the process variable”. At any given instant, the difference between the process variable and the set point is used by the control system code, to find the wanted actuator output to drive the system. This is named as closed loop control system, because the process of reading sensors to provide constant feedback and calculating the desired actuator output is repeated continuously and at a fixed loop rate.

In many cases, the actuator output is not the only signal that has an effect on the system. There are many other disturbances and a control system is designed to minimize the effect of these disturbances on the process variable.

After utilizing one or all of these parameters to describe the performance necessities for a control system, it is beneficial to determine the worst case conditions in which the control system will be probable to encounter these design requirements. Sometimes, there is a disturbance in the system that disturbs the process variable or the measurement of the process variable. It is significant to develop a control system that accomplishes reasonably during worst case conditions. The degree of how good the control system is capable to overcome the effects of disturbances is known as the disturbance rejection of the control system.

It may also happen that the response of the system to a given control output may vary over time or in relation to some other variable. “A nonlinear system is a system in which the control parameters that produce a desired response at one operating point might not produce a satisfactory response at another operating point”. “The measure of how well the control system will tolerate disturbances and nonlinearities is referred to as the robustness of the control system.”

“Some systems exhibit an undesirable behavior called deadtime”. “Deadtime is a delay between when a process variable changes, and when that change can be observed”. Deadtime may also be produced by a system or output actuator that is sluggish to respond to the control command.

Loop cycle is also an important parameter of a closed loop system. "The interval of time between calls to a control algorithm is the loop cycle time".

6.4 PROPORTIONAL RESPONSE:

"The proportional component depends only on the difference between the set point and the process variable". This difference is called as the Error term. The proportional gain (Kc) signifies the ratio of output response to the error signal. In general, growing the proportional gain will enhance the speed of the control system response. However, "if Kc is too large, the process variable will start oscillating. If it is increased further, the oscillations will become larger and the system will become unstable and may even oscillate out of control."

6.5 INTEGRAL RESPONSE

"The integral component sums the error term over time". The outcome is that even a minute error term will make the integral part to grow slowly. The integral response will constantly grow over time lest the error is zero, so the final effect is to take the Steady-State error to 0. "Steady-State error is the final difference between the process variable and set point". "A phenomenon called integral windup results when integral action saturates a controller without the controller driving the error signal toward zero."

6.6 DERIVATIVE RESPONSE

"The derivative component causes the output to decrease if the process variable is increasing rapidly". The derivative response is related to the rate of change of the process variable. "Increasing the derivative time (Td) parameter will cause the control system to react more strongly to changes in the error term and will increase the speed of the overall control system response". Most practical control systems use very small derivative time (Td). The Derivative Response is highly sensitive to noise in the process variable signal. "If the sensor feedback signal is noisy or if the control loop rate is too slow, the derivative response can make the control system unstable."

6.7 PID IN CUSTOM CONTROLLER

PID control is implemented in traditional way. The set point i.e. the desired position in terms of encoder pulses is input from GUI. The current position of the motor is continuously monitored by the Arduino Mega interrupts. These two values are compared and the error in the position to be desired is calculated. PID works to decrease this error in minimum time. The output of the PID is the speed of the motor in terms of PWM signal. This position error is directly proportional to the PID output. The more the error in position, the greater will be the speed of the motor to achieve the desired position in minimum time. As soon as the motor moves towards the desired position, the encoder also rotates and the increase in encoder signal is monitored and saved in a variable, this is the current position record of the motor. The difference in motor position error will be reduced as the motor moves towards its destination, so is the speed of the motor. At the desired position, the motor stops as the position error will be zero, so is the motor's speed.

The values of PID constants, Ki, Kp, Kd, were found through hit and trial method. The value of integral windup was set to 10,000. Integral windup was used as in case where motor is externally kept at halt, the error in loop cycle keeps on increasing as the motor is not moving. This integral windup keeps the integral term in calculated limits.

CHAPTER NO. 07

FORWARD KINEMATICS OF RHINO XR4

7.1 INTRODUCTION:

Precise mathematical modeling of motion helps us to control the motion of robot accurately. Forward Kinematics of any robotic mechanism is to model the motion of robot such that having joint variable(s) information, the location of end effector can be found. By location here means both position and orientation. This chapter describes the development of the forward kinematics of RHINO XR4.

7.2 DEVELOPMENT OF THE FORWARD KINEMATICS OF RHINO XR4:

A manipulator is composed of serial links which are affixed to each other through revolute or prismatic joints from the base frame through the end-effector. Calculating the position and orientation of the end-effector in terms of the joint variables is called as forward kinematics. In order to have forward kinematics for a robot mechanism in a systematic manner, one should use a suitable kinematics model. Denavit-Hartenberg method that uses four parameters for the development of forward kinematics and it is the most common method for describing the robot kinematics. These four parameters are $\theta_i, d_i, a_i \; and \; \alpha_i$.

- θ_{i+1} is the rotation about the previous z-axis to make the x-axes of frames $i_n and \; i_{n+1}$ parallel or coincident.
- d_{i+1} is the translation of the origin of previous frame along Z_n to make them coincident.
- a_{i+1} is the translation of the origin of previous frame along X_n.
- α_{i+1} is the rotation about previous x-axis to make both frames exactly conincident.

This is basically a set of parameters that allows us produce the mathematical model of any robotic mechanism such that our work is just reduce to providing an input and the produced mathematical model based on its form gives us particular. Input in this case is joint variable(s) as all the joints of RHINO XR4 are revolute and the output is robot's end effector's position and orientation.

For the development of forward kinematics of RHINO XR4 following steps were followed [23];

- Considering the Universal frame of reference in the workspace of RHINO XR4.
- Assigning local frames to each joint or axis of motion. Assigning the Z-axis along the axis of rotation of a particular joint and then assigning X-axis according to the pre-defined rules of D-H parameters.
- Calculating the D-H parameters. i.e. $\theta_i, d_i, a_i \; and \; \alpha_i$
- Completing D-H parameters table.
- Forming transformation matrices for each transformation between two frames.
- Multiplying transformation matrices to get the final transformation matrix.
- Final transformation matrix has been used in the GUI for showing the current configuration of the End Effector of the robot. And joint variables have been utilized in the GUI as controls.

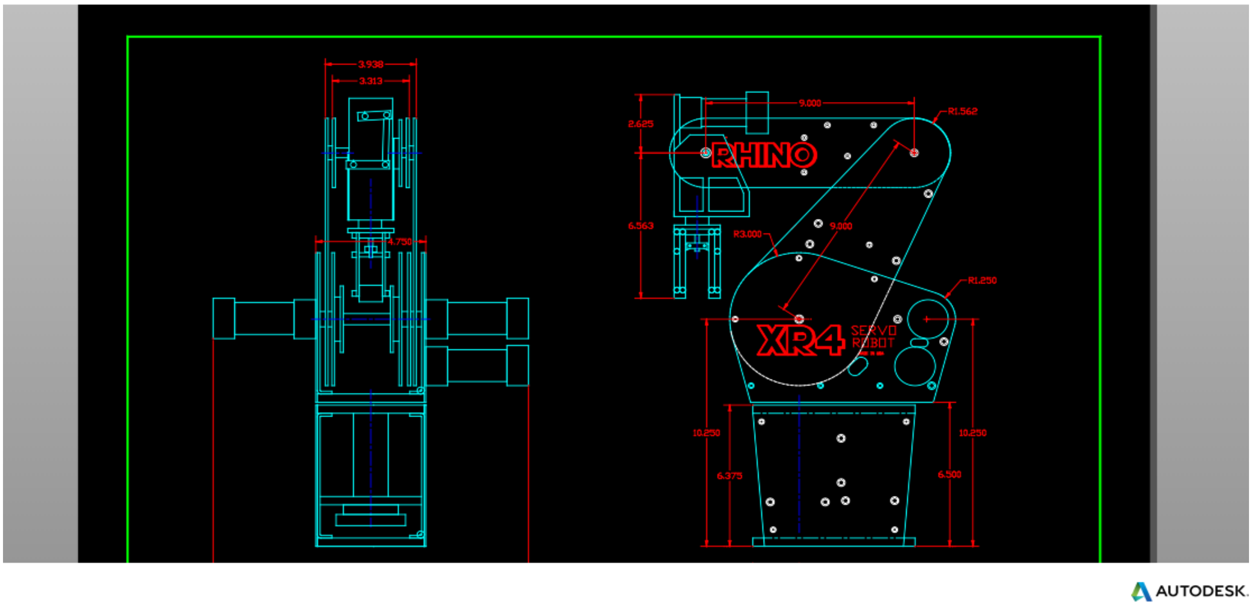


Figure 7.1 RHINO XR4 Dimensional Diagram [24] :

Considering the original mechanical structure and measurements from the above figure D-H parameters for RHINO XR4 have been calculated and they are shown in the table 7.1.

Table 7.1: D-H parameters Table

#	θ_i	d_i (cm)	a_i(cm)	α_i
0-1	θ_1	d_1=9.8425	a_1 = 0	$\pi/2$
1-2	θ_2	d_2 = 0	a_2=22.86	0
2-3	θ_3	d_3 = 0	a_3=22.86	0
3-4	θ_4	d_4 = 0	a_4 = 0	$\pi/2$

After having D-H parameters of the robot transformation matrices have been formed and they are shown below;

$$\boldsymbol{A}_0 = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{A}_1 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{A}_2 = \begin{bmatrix} C_3 & -S_3 & 0 & a_3C_3 \\ S_3 & C_3 & 0 & a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{A}_3 = \begin{bmatrix} C_4 & 0 & -S_4 & a_4C_4 \\ S_4 & 0 & C_4 & a_4S_4 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{A}_4 = \begin{bmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The final transformation matrix which is obtained by the multiplication of these transformation matrices is shown below;

$${}^{0}_{5}T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$n_x = C_1C_{234}C_5 + S_1S_5$$

$$n_y = S_1C_{234}C_5 - C_1S_5$$

$$n_z = S_{234}C_5$$

$$o_x = -C_1C_{234}S_5 + S_1C_5$$

$$o_y = -C_5C_5 - C_{234}S_1S_5$$

$$o_z = -S_5S_{234}$$

$$a_x = -C_1S_{234}$$

$$a_y = -S_1S_{234}$$

$$a_z = -C_{234}$$

$$p_x = C_1(a_3C_{23} + a_2C_2)$$

$$p_y = S_1(a_3C_{23} + a_2C_2)$$

$$p_z = d_1 + a_3S_{23} + a_2S_2$$

Forward Kinematics equations are incorporated in the graphical code of the GUI which is shown in the Fig. 7.2a with the help of these equations Homogenous Transformation Matrix has been displayed in the GUI which is constantly updated in the GUI. This Homogenous Transformation Matrix has been obtained through two other soft wares MATLAB and RoboAnalyzer. They are also shown in the Fig.7.2b and 7.2c.

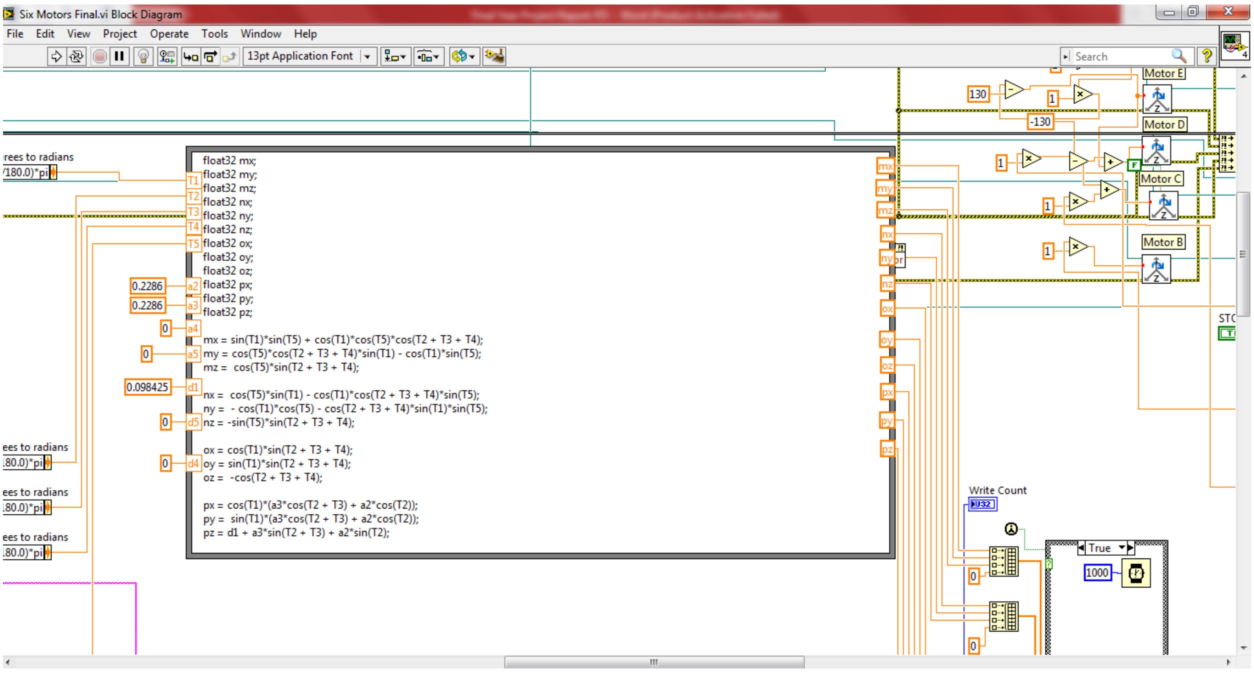


Fig.7.2a Forward Kinematics Equations Incorporated in the Graphical Code:

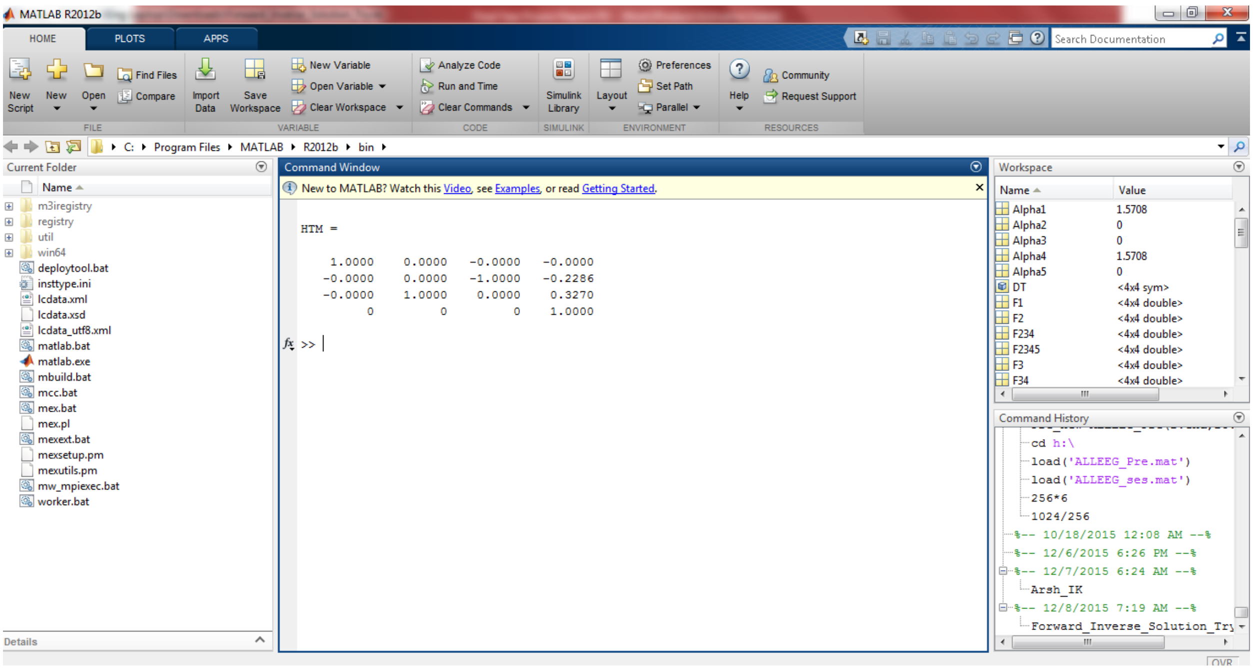


Fig. 7.2b Homogenous Transformation Matrix through MATLAB:

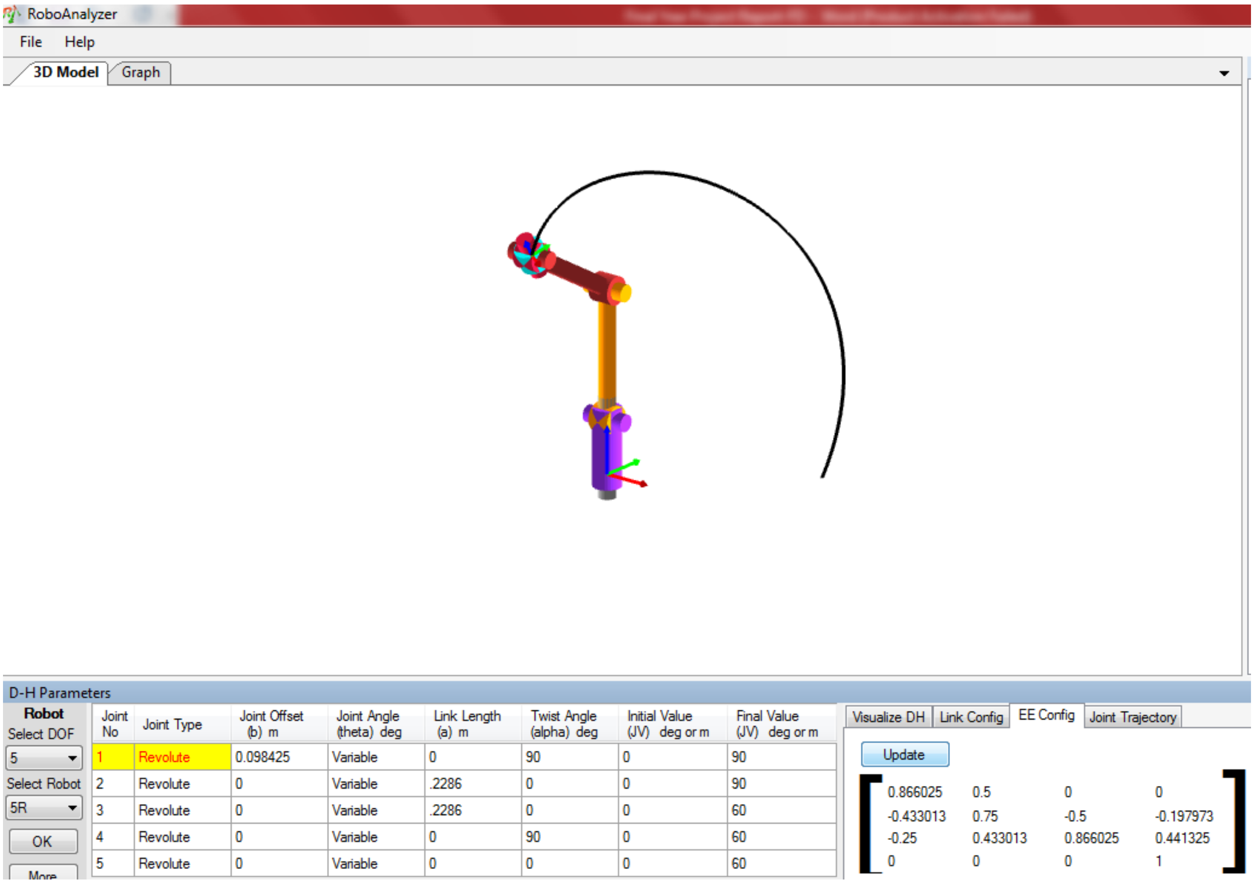


Fig. 7.2c Homogenous Transformation Matrix through RoboAnalyzer:

Using this homogenous transformation matrix forward kinematics can easily be verified by measuring the end effector position with measuring tape. Verified forward kinematics will lead to the accurate inverse kinematics solution i.e. determination of the values of joint variables while having particular position and orientation.

CONCLUSION

Development of the custom controller for RHINO XR4 robotic arm with only mechanical structure in hand was a challenge which was undertaken and has been accomplished successfully. Previous works such as [7] have a IEEE conference paper for their work on the educational workstation build around RHINO XR4 and they have done the same task but they have utilized MATLAB for their work which has been briefly described in the first chapter. All the modules of the controller have been developed and integrated into the custom controller and are interfaced together. The developed custom controller has successfully replaced MARK IV controller and it is much less than MARK IV in cost, weight and size. As Far as the functionality is concerned it has certain gains over MARK IV and it has some limitations as well. It has a robust PID algorithm implemented in the code in Arduino IDE which allows for precise, smooth and accurate control of the robot. The developed controller is a dedicated controller for RHINO XR4 but with certain modifications it can also control RHINO XR3 and SCARA as well. This controller can also control two un-encoded motors that may be used with some special purpose end effectors, moreover some sensors can also be used as the input sensors that will be aimed to improve the capabilities of RHINO XR4. The GUI developed for the controller has made the control of RHINO XR4 easy and interactive and it will be helpful for the students for learning the basics of Robotics with practical demonstrations using RHINO XR4. This project also under takes the development of the forward kinematics of the robot that gives accurate results for the homogenous transformation matrix same as obtained with a simulation software RoboAnalyzer developed in Mechatronics Lab, Department of Mechanical Engineering at IIT Delhi, New Delhi, India.[25]

LIMITATIONS

The custom controller has certain limitations of course it would have some as the RHINO ROBOTICS LTD is a world leader in Educational Robots [1] . MARK VI is an 8 axis Controller with 8 input lines, 8 output lines and 2 auxiliary ports. While the developed controller is a 6 axis controller it may have more auxiliary ports than MARK IV but it depends on the application since it is dedicated to the routine control of RHINO XR4, therefore it has no auxiliary ports and I/O lines but the can be easily available for some special applications of RHINO XR4.It has a microprocessor in it while the developed controller incorporates microcontroller Arduino Mega 2560. Only short in number of encoded motors to be controlled by the controller, the developed controller stands beside MARK IV in functionality.

FUTURE RECOMMENDATIONS

One year time period was in fact not sufficient for all the tasks that we wanted to accomplish with RHINO XR4, though we have done our part but a lot more could be done in this regard. Inverse Kinematics of RHINO XR4 has been solved but it is left to be tested so one could determine the complete inverse kinematics solution and can also determine the associated singularities and degeneracies. Having the complete inverse kinematics solution will open the door to many possibilities such as GPS control of RHIINO XR4 which can make it possibly a catching robot and many manipulating tasks such as in assembling. Force sensitive sensors can also be incorporated in RHINO XR4 end effector so that it could handle objects according to their strengths. Proximity sensors can also be added for adding the proficiency of RHINO XR4.

7.6 APPLICATIONS

RHINO XR4 with its controller finds applications mainly in education and research. It may be utilized in various manipulating task and in assembling units. With conveyor belt at its base and with a special end effector it can be used in the bomb diffusion process. But the work that was done for the development of the custom controller along with its GUI has laid the foundation for automation of manipulators in the assembling units and in laboratories.

REFERENCES

1. LTD., R.R. *Educational Robots*. Available from: http://www.rhinorobotics.com/.
2. Ltd., R.R. *RHINO XR4 Specification Table*. Available from: http://www.rhinorobotics.com/xr4_flyer.html.
3. Ltd., R.R. *MARK IV Specification Table.*; Available from: http://www.rhinorobotics.com/xr4_flyer.html.
4. ROBOTICS, R. *MARK IV Picture*. Available from: http://www.rhinorobotics.com/markiv.html.
5. ROBOTICS, R. *Teach Pendant.*; Available from: http://www.rhinorobotics.com/markiv.html.
6. LTD>, R.R., *Mark IV Owner's Manual for Windows.*, in *Mark IV controller Overview*.
7. Alcalde, L.R.S.J.a.V.H.C., *<An Educational Robotic Workstation based on the.pdf>.*
8. Cabré, T.P., et al., *Project-based learning example: Controlling an educational robotic arm with computer vision.* Tecnologias del Aprendizaje, IEEE Revista Iberoamericana de, 2013. **8**(3): p. 135-142.
9. MALEK, Z.A.B.A., *ARM ROBOT WITH SENSING INTELLIGENCE*. UNIVERSITI TEKNOLOGI MALAYSIA.
10. Mrad, F.T. and G. Deeb, *Extending the utility of the RHINO educational robot.* Education, IEEE Transactions on, 1997. **40**(3): p. 184-189.
11. SING, W.T., *PICK AND PLACE USING RHINO XR-4.*
12. Soares Jr, L.R. and V.H.C. Alcalde. *A laboratory platform for teaching computer vision control of robotic systems*. in *Frontiers in Education Conference, 2008. FIE 2008. 38th Annual*. 2008. IEEE.
13. Cook, A.M., et al., *School-based use of a robotic arm system by children with disabilities.* Neural Systems and Rehabilitation Engineering, IEEE Transactions on, 2005. **13**(4): p. 452-460.
14. PITTMAN. *RHINO XR4 actuators*. Available from: http://www.pittman-motors.com/.
15. Electronics., P.R.a. *Motor Driver Picture.*; Available from: https://www.pololu.com/product/2503.
16. Arduino.CC, *Arduino UNO*.
17. Arduino.CC. *Arduino Mega 2560*. Available from: https://www.arduino.cc/en/Main/ArduinoBoardMega2560.
18. Arduino.CC. *Arduino Mega Picture*. Available from: https://www.arduino.cc/en/Main/ArduinoBoardMega2560.
19. MakerHub, L. *LabVIEW MakerHub*

Available from: https://www.labviewmakerhub.com/.

20. Electronics, L. *Proteus 7*. Available from: http://www.labcenter.com/index.cfm.
21. RoboAnalyzer. *3D Model Based Robotics Learning Software*. Available from: http://www.roboanalyzer.com/.
22. Instrumentation, N. *NI-VISA*. Available from: https://www.ni.com/visa/

23. Niku, S.B., *Introduction to Robotics*.
24. AutoDesk. *RHINO Dimensional Diagram*. Available from: http://usa.autodesk.com/adsk/servlet/index?id=2787358&siteID=123112&wac=IL13872.
25. RoboAnalyzer. *Simulation Software for Robotics*. Available from: http://www.roboanalyzer.com/.