Master Thesis
Computer Science
Thesis no: MCS-2010-33
September, 2010

Control of Articulated Robot Arm by Eye Tracking

Muhammad Imran Shahzad and Saqib Mehmood

School of Computing
Blekinge Institute of Technology
Box
SE – 37179 Karlsakrona
Sweden

Internet : www.bth.se/com
Phone : +46 457 38 50 00
Fax : + 46 457 271 25

This thesis is submitted to the School of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author(s):
Muhammad Imran Shahzad
19LGH 17 Folkparks Vägen 37240, Ronneby, Sweden
saramadeel@gmail.com

Saqib Mehmood
19LGH 17 Folkparks Vägen 37240, Ronneby, Sweden
saqib495@hotmail.com

University advisor(s):
Craig Lindley
Craig.lindley@bth.se
Game Systems and Interaction Research Laboratory

Department of
Interaction and System Design
Blekinge Institute of Technology
Box 520 SE- 37225
Karlsakrona, Sweden

Internet : www.bth.se/tek
Phone : +46 457 38 50 00
Fax : +46 457 102 45

ABSTRACT

Eye tracking has many comprehensive achievements in the field of human computer interaction. Uses of human eyes as an alternative of hands are an innovative way in the human computer interaction perspective. Many application of autonomous robot control has already been developed, but we developed two different interfaces to control the articulated robot manually. The first of these interfaces is controlled by mouse and the second is controlled by eye tracking. Main focus of our thesis is to facilitate the people with motor disabilities by using their eye as an input instead of a mouse. Eye gaze tracking technique is used to send commands to perform different tasks. Interfaces are divided into different active and inactive regions. Dwell time is a well known technique which is used to execute commands through eye gaze instead of using a mouse. When a user gazes in an active region for a specific dwell time, the command is executed and the robot performs a specific task. When inactive regions are gazed at, there no command execution and no function are performed. The difference between time of performing the task by mouse and Eyetracking is shown to be 40 ms, the mouse being faster. However, a mouse cannot be used for people with motor disabilities, so the Eyetracker in this case has a decisive advantage.

Keywords: Eyetracking, Interface, Articulated robot

ACKNOWLEDGEMENT

We are greatly thankful to our family for their encouragements, support, assistance and their prayers to ALLAH for our success.

We would like to thanks our Supervisor for this Master Thesis, Craig Lindley, Blekinge Institute of Technology, SWEDEN for his commitment and inspiration to help us achieving our research goals. Craig Lindley has provided us with an essential advice and extraordinary guidance and without his supervision it was not possible to do this research.

We further thank Blekinge Institute of Technology administration and staff who provide us an opportunity to have quality education and research exposure to be successful in our life.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

For many centuries, human beings have continuously made efforts to quest substitute that will imitate their role in numerous instances of interaction with environment. Several economical, social, scientific and philosophical principals have inspired this research. Artifacts to life are greatest ambition of human being. Czech playwright Karel Capek coined the term "***ROBOT***" – derive from Slav robota which means executive labor—this denoting a humanoid automation created by the Ficticious Rossum. [1]

In the field of robotics for manufacturing, there exist three types of automation:

1. Rigid Automation: In this kind of automation, the same types of machines are manufactured by automated assembly processes.
2. Programmable Automation: In this kind of automation, variable batches of different type of machines are manufactured

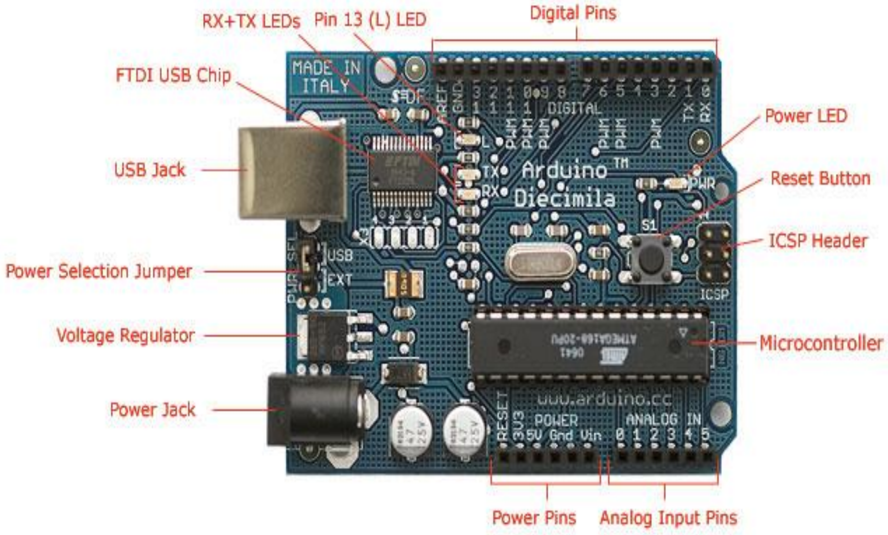
3. Flexible Automation: It deals with evaluation of programmable automation. [1]

With the advancement of robotics technology, robots are used for many purposes, such as painting, welding, sealing, assembly etc. Robots need some basic teaching for performing these tasks. These teaching operations are indispensible to bring in use as it is time taker activity. Also, there is worse accuracy in positioning of robots.

Position of articulated robot is calculated on angles of joints which are detected by rotary encoders. It could put bad affects on accuracy. This problem can be overcome by any board that will rotate the motors (servo) of robot. We have used board of "***Arduino Diecimila"*** company.

This is microcontroller board (ATmega168). There are 14 digital input/output pins (6 are PWM outputs), 1USB connection for communication, 1header (ICSP), 1 crystal oscillator (oscillator), 1 power jack. To start this board, we simply connect it with computer with USB cable or AC-to-DC adapter or battery.

There are servo loops that can rotate motors from 0 to 180 degree. It gives sufficient accuracy.



Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

Figure 1 Arduino board

Different methods are being used to make interaction with computer with passage of time, from long commands to mouse, recognition of voice to eye-tracking and punch card to switch boards.

Research on Eyetrcking has been conducted for almost 125 years [13]. Dodge and Cline (1901) used a method of light reflection from the cornea to develop eye tracking technique. It was a precise, non-invasive technique. They recorded position (horizontal) of eye onto photography photographic plate and the method required the participant's head to be motionless.

Later on, Judd, McAllister & Steel (1905) record "moving picture" in two dimensions. They recorded white spec material instead of cornea. They made more research during twentieth century on corneal reflection and motion picture techniques. [4]

Head-mounted eye tracking [3], [5], [6], is also used on large scale.

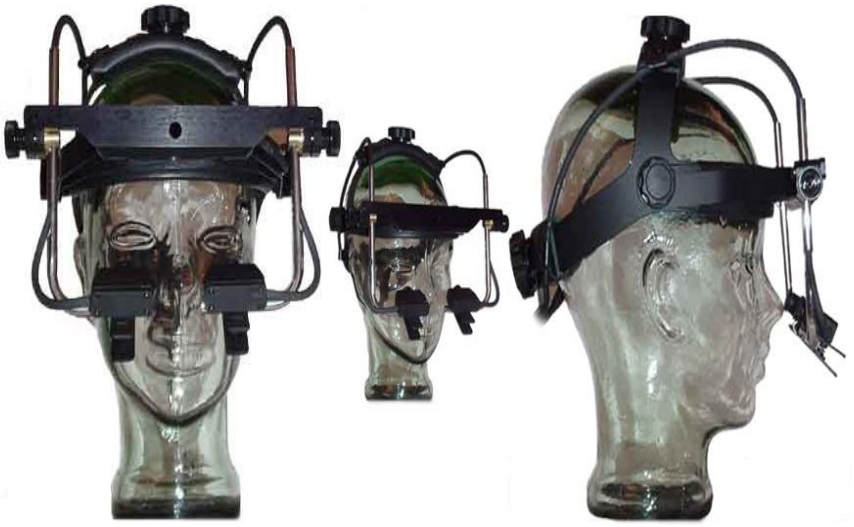


Figure 2 Head mounted Eyetracking system

Yarbus (a Russian psychologist) has also done practical work in gaze tracking world. In 1950s and 1960s, he did research in saccadic exploration of complex images. He made recordings of eye movements when observers view natural scenes [7]. Purpose of related work is to make technical improvement to enhance exactness and correctness of the eye trackers [8].

Researchers in the 1970s made technical improvements to increase accuracy and precision. This new research used multiple eye reflections to dissociate eye rotations from movement of the head (Cornsweet and Crane, 1973) which boosted tracking precision and provides the base for greater freedom of participant movement [4].

With the increasing use of computers in the 80's, there was increasing interest in gaze tracking and designing new interfaces and interaction with computer [9], [10],[11]. Levine was another pioneer who exposed the potential of eye tracking for interaction with applications [12].

In our project we have developed different interfaces for controlling an articulated robot arm. This development was done using JAVA. We have controlled these interfaces with both a mouse and an eye tracker device (Tobii (company)) .These interfaces send data to an arduino board that controls the robot. We send joint number and angles to the robot as specified on the interface.

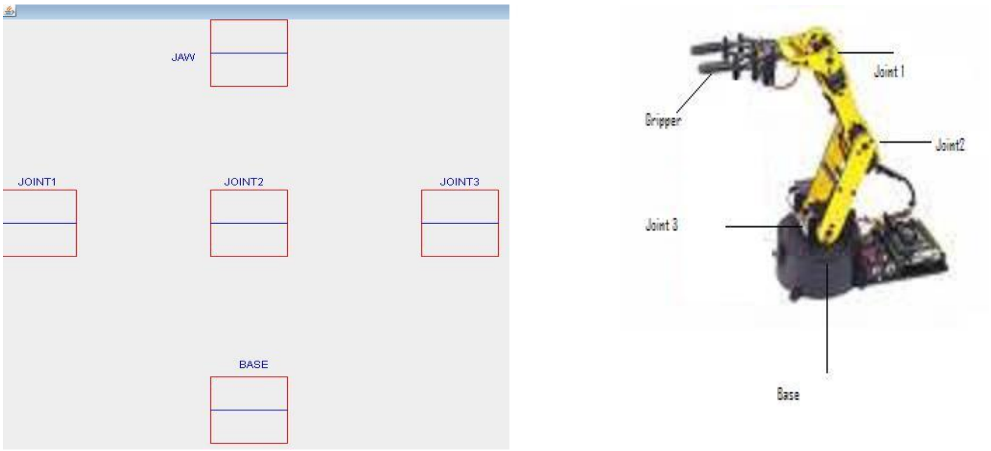


Figure 3 Interfaces and Robot

As Figure 3 shows, there are five rectangles namely, JAW, BASE, JOINTS (1, 2, 3) to control five different joints of the robot arm. Our robot arm can move from 0 to 180 degree. When we click on Jaw the gripper of the robotics moves. Joints 1, 2, 3 and the base work similarly. One noticeable thing is that each rectangle has two parts .When we click on the upper part then the joint moves from 0 to 90 degrees and when we would click on the lower part the joint moves from 90 to 180 degree.

2 BACKGROUND

2.1 What is eye Tracking Technology?

Eye trackers estimate direction of gaze of a user. Early eye tracking systems are surveyed by Young and Sheena [32].and recently by Duchowski [33].

Eye-tracking systems can track the eye-in-head gaze direction or the combined eye-head gaze direction. These systems could be intrusive (physical contact between the sensor apparatus and the user) [37] [34] [35] [36] or non-intrusive (camera-based techniques) [37] [38] [39] [40] [41] [42] [43].
Researchers in field of eye-movements have been pursued for almost 125 years [13]. Dodge and Cline (1901) used the method of light reflection from the cornea to develop eye tracking. It was a precise, non-invasive technique. They recorded the position (horizontal) of the eye onto a photographic plate, which required the participant's head to be motionless.
Later on, Judd, McAllister & Steel (1905) record ***"moving picture"*** in two dimensions. They did recording of white spec material instead of cornea. They made more research during the twentieth century on corneal reflection and motion picture techniques. (See Mackworth & Mackworth, 1958, for a review)

In the 1930s, Miles Tinker and his colleagues began to apply photographic techniques to study eye movements in reading (see Tinker, 1963 for a thorough review of this work).Miles Tinker and his colleagues studied eye movements in reading by applying photographic techniques in the 1930s.They observed effects on speed of reading, eye patterns by changing print size, page layout etc. Paul Fitts and his colleagues started "motion picture cameras" to observe pilots' eyes during landing in cockpit in 1947[4].
Impressive major innovation in eye tracking was the development of head-mounted Eyetracking ([3], [5], [6]), which is still widely used. Yarbus (a Russian psychologist) has made valuable contributions to gaze tracking, exploring complex images in the 1950s and 1960s. He recorded the movements of eyes of observers when looking at different scenes [7].
Electro-oculography is a method based on electrostatic fields that exist when eyes rotate. The position of the eye can be estimated by recording skin potential around the eye. Since this technique requires the close contact of electrodes, it is wearisome [14].
Three different approaches exist for detection and measurement of eye gaze:

- ➢ The glint
- ➢ 3D-model
- ➢ Local linear map network

In the glint approach, we calculate angle of the visual axis and fixation point by tracking the angle of the pupil and light reflection point from cornea (glint).
(Study of similar system can be found in [15]).

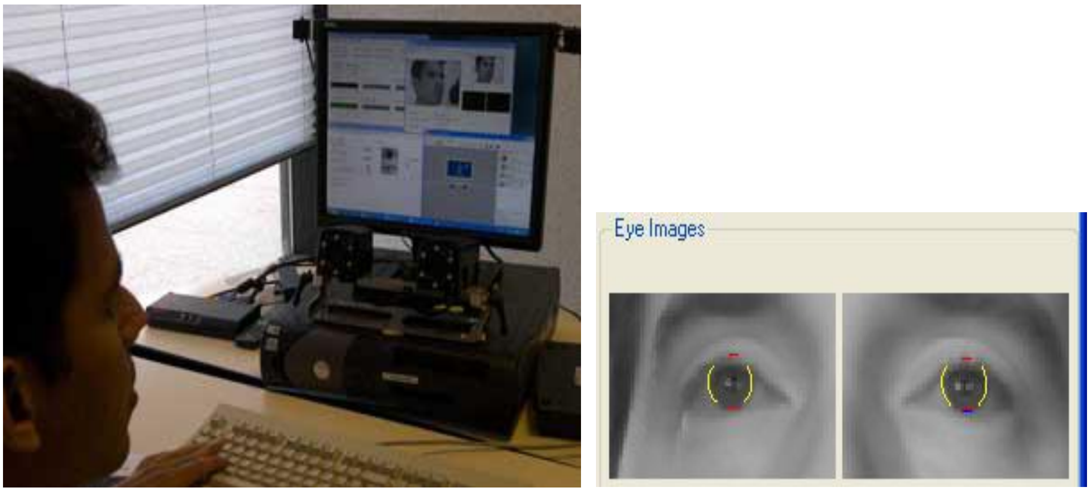


Figure 4 Eye Tracking System of early age

In 3D-model, we can detect the mouth, pupils, and face by using serialized image processing. With a 3D model [17], we evaluate the face orientation and eye images can be used to estimate gaze direction. (Fresh research is found in [16]).

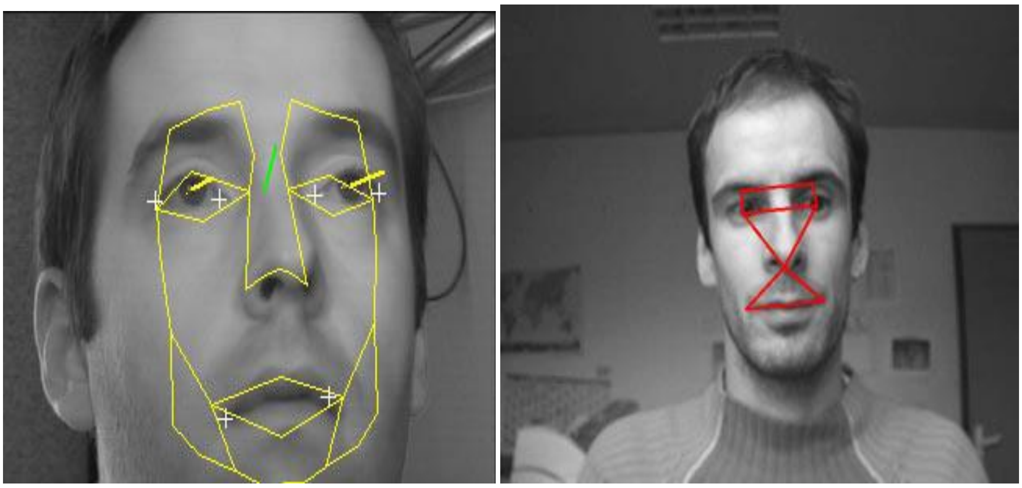


Figure 5 Head model for detection of eye and gaze orientation

Local linear map network is more trivial [18], it recognize the head orientation of a user with the help of infrared light.

2.2 Eye and gaze tracking applications

When we review history, we found that the first example of eye tracking systems is user interface design. We can design devices, cockpits, cars, etc. by tracking "where people are looking". [31]

In this field, a number of publications can be found [19] [20] [21].This whole study depends on the eye-mind hypothesis.

Eye mind hypothesis shows that what a person is looking at and what are the indications of assuming the thought on top of the stack of cognitive processes. The use of eye-mind hypothesis is meaningful and visible and improves design of interface. [22] [31].According to Jacob, the basic reason behind the dawdling appearance of interfaces [23] [31] is the Midas touch problem: the application must not respond every time the goal of the gaze changes.

We focus on aspect of eye and gaze tracking in game environment. But it is reported in 2005 [24] [31] that there is no existence of information about eye tracking and games of computers. Nevertheless, it is very motivating to reveal from previous work about foundations of players eye behaviors [25] [31].
Research reveals that action game playing is a constructive means for rehabilitation of visually impaired people [26]. Eye or gaze tracking is not a standard input for commercial games. The first contribution to handle this problem was made by [27] [31] in which a system was developed for eye motion disability rehabilitation. Joystick was used to play a game in environment of multimedia.

Figure 6 User points to objects

Laurent Itti Lab also made comprehensive contribution relating realistic avatar eye and neurobiological model of visual attention [28] [31].
This team has worked in field of interactive visual environments and mechanism of gaze direction [29] and S. Yan and N. Seif El-Nasr by [30] [31] has reinforced the work investigating visual attention patterns within 3D games.

2.2.1 **Midas touch, Dwell time & WPF Custom controls**

"How do you make a distinction between users glancing at objects and fixations with the intention to start a function”? This is the well known problem usually referred to as the "Midas touch problem". The interfaces that depends on gaze as input should carry such a interaction and must be able of making difference between glances (just looking around) and fixations (objective to start a function). The most common solution to this problem is "dwell-time" where you can make active the functions by prolonging fixation on an icon, button or image. Its range varies from 4 to 500 ms. the problem of people who is suffering from Amyotrophic lateral sclerosis (ALS) can be solved by this way.
Amyotrophic lateral sclerosis (ALS) is severe type of neurological disease which affects nerve cells and causes muscular weakness.
But dwell time has also some problems, e.g. prolonged fixation results in slow interaction because user has to sit through the dwell-time. When the user makes fixations on any object an event is triggered. This event creates new threads and it decide that if there is enough fixations on area then continue it, otherwise finish this thread. At the end it measures if fixations have resided within the area for more than 70% of the time, in that case, it activates the function.
Windows Presentation Foundation (WPF) is another progress in development of user interfaces. The Microsoft Expression Blend is a method for easing graphical design of components. It can be broken into different projects and included as a DLL file [52].

2.3 **Tobii Technology System**

Tobii Technology has camera-based eye tracker systems in which camera and light source are affixed to a monitor on a permanent basis. T/X-series and 50-series eyetrackers use infrared pupil-center/corneal-reflection eye tracking [37] [44] and respond via a Tobii Eye Tracker (TET) server.

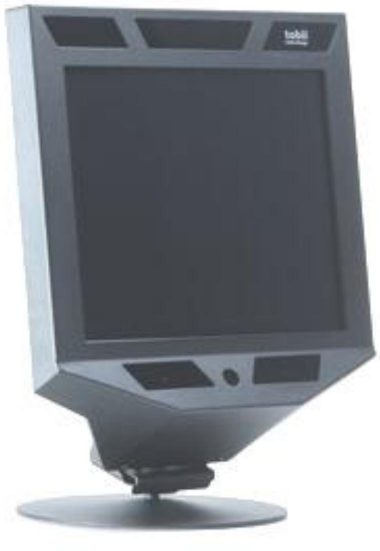


Figure 7 Tobii T 60 Eyetracking

The TET server does image processing on incoming data from a video camera and maps image parameters to screen coordinates. The server can exist on a separate machine from the machine that receives the reports of tracker and it communicates over TCP/IP [37].

2.3.1 **Model T60**

The T60 processes binocular tracking at 60Hz, and its head movement tolerance is 44x22x30cm volume centered 70cm from the camera. The T60 combines both bright and dark pupil tracking. Latency rate is 33ms between camera and gaze [37] [44].

The monitor (LCD) of the T60 is 17 inches. The system has an accuracy of 0.5 degree with fewer than 0.3 degree of drift over time and fewer than 1degree error due to head movement [37] [44]. Its processing rate of binocular tracking is 60Hz and the movement of the head is 44x22x30cm volume centered 70cm from the camera. Both bright and dark pupil tracking is combined.

During the tracking process the Tobii eye tracker uses infrared diodes that produces reflection pattern on the corneas in the user’s eye after it sensors come into action and it collect reflection patterns and other visual information. Different complex image processing algorithms detect features of corneal reflection patterns. After some mathematical calculations gives us gaze point on screen, which shows where user looking [45].

2.4 Eyetracking Overview /Methodology

2.4.1 Eyetracking

The term eye tracking is used here as way the estimation of direction of the user's gaze.
The estimation of the gaze direction describes the identification of the objects upon which the gaze falls. Eyetracking as standard computer machine identifies the object of gaze by the coordinates on the screen. To find out gaze direction in 3D virtual worlds is difficult and becomes hard when the real world interact [33].

2.4.2 Eyetracking Methodology

There are 3 broad categories of eye movement measurements methodologies.

1. Electro –Oculography
2. Scleral Contact Lens
3. Video / Photo –Oculography

- **Electro –Oculography**

It based on recordings of differences of skin surrounding the ocular cavity. It is basically used for point of regard measurement. Point of regard measurement totally based on reflection of corneal. Contact lens was developed for improvement of accuracy in 1950. Devices like Small mirror, coils of wire are used with contact for improvement of accuracy.
These rely on features of eyes like e.g., the pupil, iris-sclera boundary, corneal reflection of light source. Recorded potential is in range of 15- 200 v with sensitivity of 20 v. This technique is not suitable for point of regard .It measures eye movement related to head movement [33].

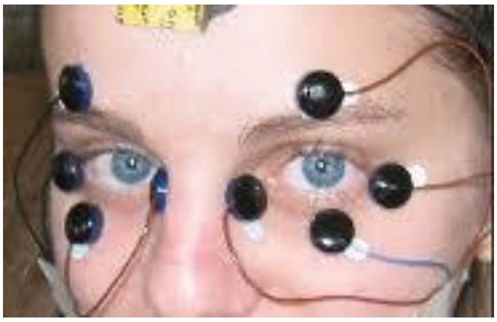


Figure 8 Sensors attached at the skin around the eyes

- **Scleral Contact Lens**

In this technique contact lens is worn on eyes. Optical reference object or mechanical object is mounted on contact lens. On modern contact lens mounting stalk is attached. This contact lens is extended over sclera and cornea. Reflecting phosphors, line diagram, wire coals are attached on mounting stalk. This is intrusive method. This technique has special consideration like, practice and care is required for insertion of lens. User can feel discomfort when he uses it. This is not suitable for point of regard measurement. The scleral search coil is made of a medical-grade silicone rubber suction ring .It has a suitable shape to stick on to the limbus of the eye [33].

Below mentioned 3 figures depicts a clear view of this process [65].

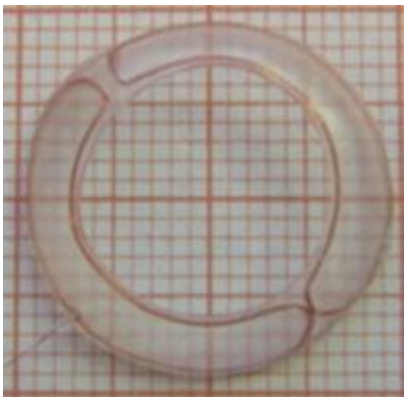


Figure 9 Scleral Contact Lens Process step 1

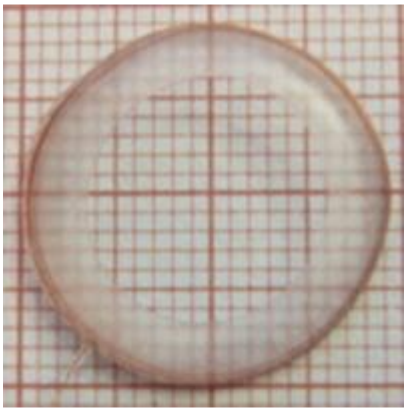


Figure 10 Scleral Contact Lens Process step 2

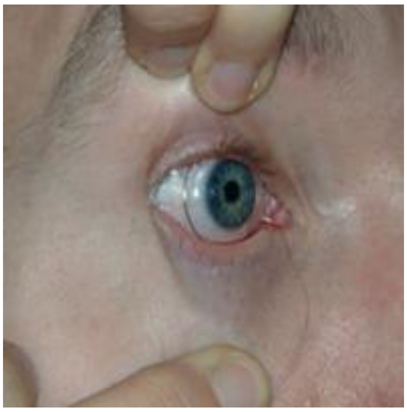


Figure 11 Scleral Contact Lens Process step 3

- **Video/Photo – Oculography**

This techniques measure to unique features of eye under rotation e.g., position of limbus, shape of pupil, corneal reflection of light source. This is not suitable for point of regard measurement. It involves visual inspection of recorded eye movement and measurement of ocular features is automatically. Visual measurement or assessment is performed by video tape frame by frame. This technique could be error free and tedious [33].

At the picture you can see the result of intelligent eye recognition: the software automatically locates the pupil and outlines the region of interest [33].

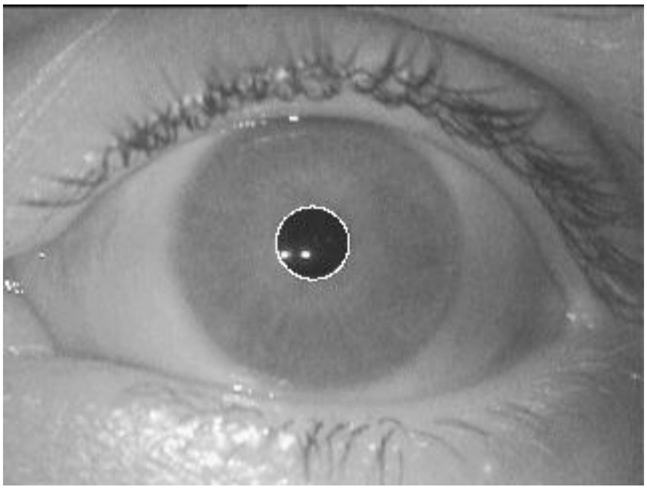


Figure 12 showing position of eye in video oculography

2.5 Motivation

Most of Eyetracking systems are based on video-based pupil detection and a reflection of an infrared LED. One reason is that Video cameras are cheaper and price for an LED is minor. Many devices like mobile phones, laptops come with built-in cameras. New machines with latest powerful processors can process the video stream necessary to do eye tracking. New head-tracking systems are also video-based. With passage of time ,due to cheaper available circuits a future standard eye tracker should not cost much more than an optical mouse or a webcam today Many people suffer from overstressing particular parts of their hands, as most interaction is done with keyboard and mouse by using the hands , that can result in carpal tunnel syndrome. The eyes are a good candidate because they move anyway when interacting with computers. New eye tracking based interface techniques could use benefits of eye movements and giving the users some benefit in terms of computer interaction [65].

2.6 Robot

Robots were introduced in 1921 by Czech Playwright Karel Capek first time by resembling it with human being in all aspects except tiredness. Robot is Czech word which means “Worker”. After the word Robot come into existence then it is needed which form of robot is required and which form of function it has to perform [51].

2.6.1 Articulated Robot

It is that types of robot which have a series of joints resembling a human arm.

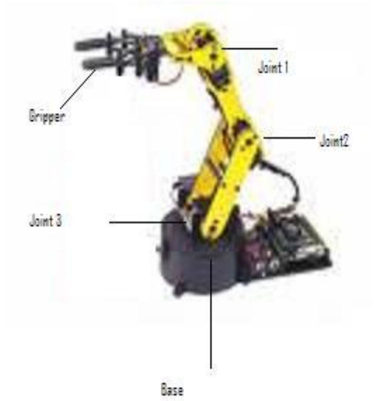


Figure 13 Articulated robot arm

The robot used for this project has 5 moving joints.

2.7 Related Work

This is fourth era of eye tracking search [46] and research is in progress to design more rapid, more trustworthy and easier to use eye tracking systems [47], [48]. Consequently, the realm of eye tracking applications is growing swiftly including both problem-solving and interactive applications. Application developers use inputs from the eyes to develop many graphical user interfaces for human computer interaction (HCI) as part of the development of interactive applications [49]. Consequently, a small number of attempts of eye tracking for human-robot interaction are existed [50].

2.8 Proposed Work

We have used Eye tracker of Tobii Company (T60). We used its built in APIs to take inputs of eyes in the form of X-AXIS and Y-AXIS. We then pass these parameters to a java application that controls the robot through an Arduino board. While robot is controlled by programming (JAVA APPLICATION). This robot can rotate from 0 degree to 180.It has5 different joints. Joints are controlled with pins number (2-10) mounted on the Arduino board. Gripper of robot can grip any object.

3 PROBLEM DEFINITION

The main focus of this study is to analyze the control of the robot. The first step involved is to control the robot with mouse by making some useful interfaces. Then take this process through eye tracking utilization.

3.1 Aims and Objectives

The aim and objective of this thesis is making an interface between articulated robot and eye tracking in the form of a prototype. The aim will be completed by addressing the following objectives.

- To control the articulated robot using gaze interaction by eye tracking.
- To find out the most effective interface to support this driving function.
- To explore eye tracking in handling articulated robotic functionality.
- To gain a deeper understanding of the degree of automation that may be applied to such a system, and how to balance or transfer between automated and manual functions.

3.2 Research Questions

There are three research questions in this study on the basis of above mentioned aims and objectives.

1. Can an articulated robot be controlled gaze interaction?
2. What kind of interface can be most effective for gaze interaction?
3. How does the best gaze-based interface compare with the mouse-based interface?

3.3 Expected Outcomes

On the basis of the Research questions literature review and experimental results, the following outcomes are possible.

1. Software for interfacing a GUI with robot control operators.
2. Software link to a Tobii T60 eye tracking system, to support gaze-directed robot interaction.
3. Several alternative interaction design implementations for gaze-directed robot control.
4. An evaluation of the different interaction models implemented, and comparison with a mouse-driven interface.

3.4 The Structure of the Thesis

This section will guide about the Structure of the thesis by chapters. The contents of each chapter are shown in this table.

No of Chapter	Title of the Chapter	Description of the Chapter
2	Backgrounds	This chapter is about background of Eye Tracking and Robot.
3	Problem Definition	This Chapter shows problems, Research questions and objectives.
4	Research Methodology	This chapter highlights the research methodology used
5	Theoretical Work / Study	This chapter shows results got by Experiments and systematic review.
6	Experiment Result and Analysis	This chapter describes the analysis of authors after conducting the experiments and research. Each research question is answered through the experiment is analyzed.
7	Conclusion / Future Works	This chapter is consisted of complete research work, conclusion and outcomes of experimental results.
8	References	This chapter consists of all used references in this research.
9	Appendix	

Table 1.Sructure of Thesis

4 RESEARCH METHODOLOGY

This chapter presents different approaches and methods used in our research work. Research methodology defines the area of research which is going to be addressed and what is the process of this research and how it leads to success. We adopted mixed methodology (Qualitative and Quantitative) to complete the thesis and research work. Quantitative research methods are capable providing numerically precise solution, which is based on facts and figures [53].

“Mixed method research in which the researcher uses qualitative research paradigm for one phase and the quantitative research paradigm for a different phase of the study” [54].

In this research thesis most emphasis is on the quantitative research methodology. In the first part of the thesis we conduct a literature review. After completing the literature review we implement a prototype application framework providing a GUI for robot control. The study of robotics control and implementing the eye tracker as a gaze input helped us a lot to make graphical user interface. Some specific interface designs will be completed. A series of experiments will then be conducted to evaluate those interface designs and for comparison with a mouse-based interface. Both interfaces are evaluated by using a mouse and then by using Eyetracking.
We designed a closed questionnaire to evaluate the interface. This is discussed in 4.7.

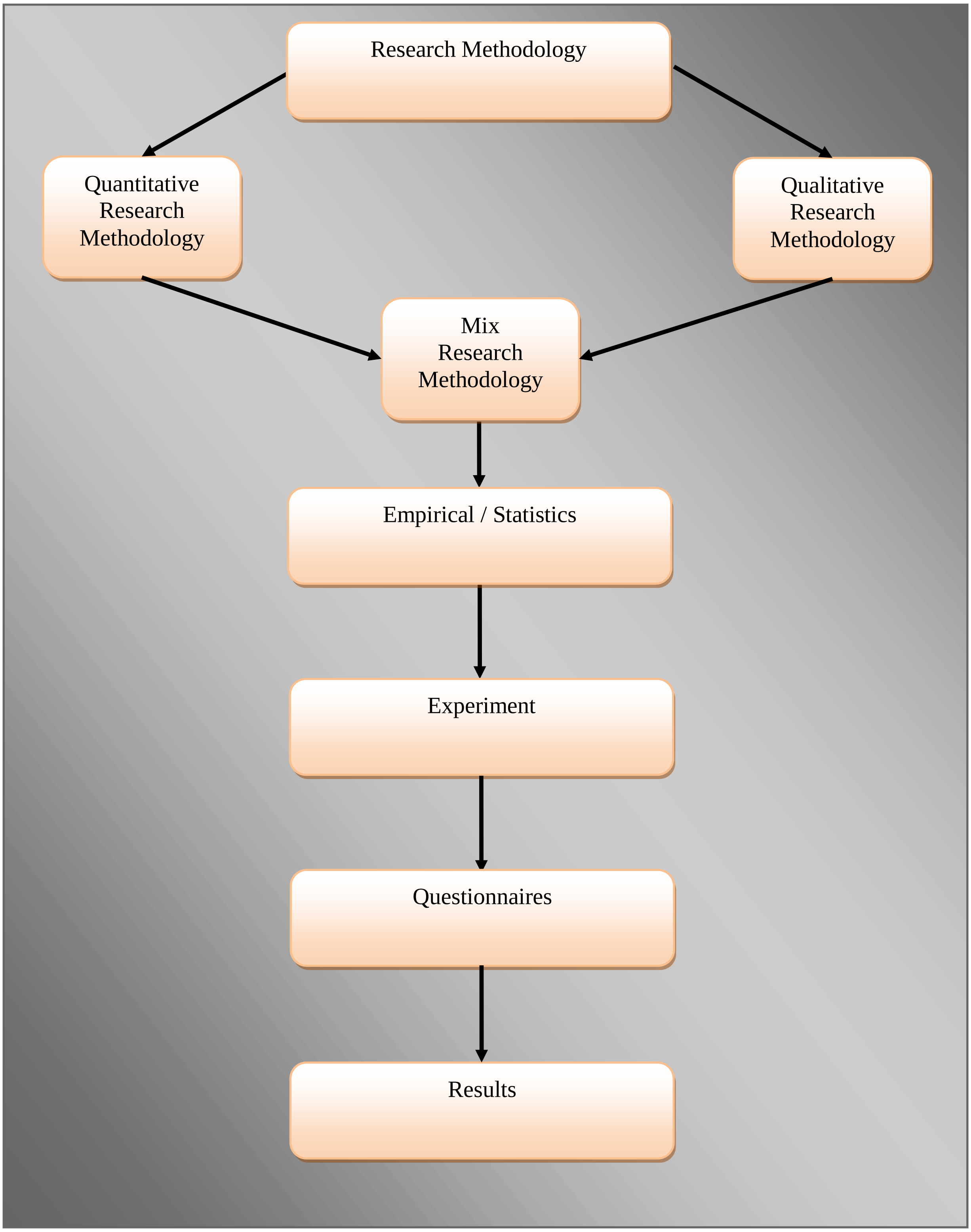


Figure 14 showing the structure of the research methodology

4.1 Literature Review

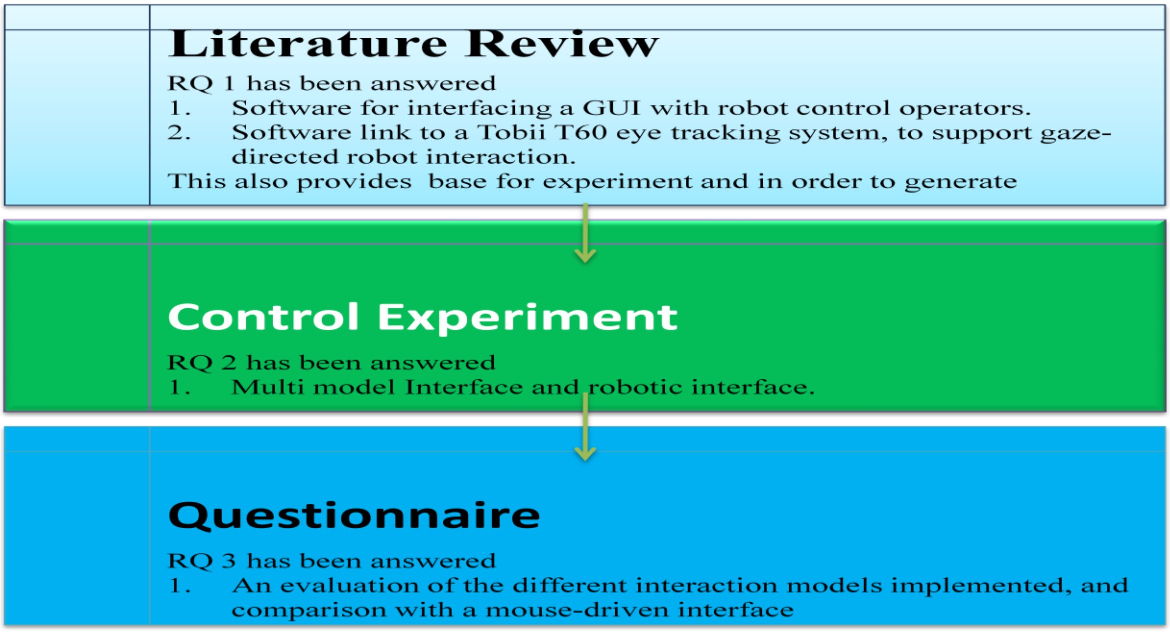
The role of a literature review is very important to know what the state of art is, what existing research there is and what other researchers says about any specific topic. A deep literature review is done to know the existing materials and to know the answers of our research questions as they have been developed so far. Our main focus was to find research papers, articles, books, journals, and websites related to our research area. For this different research engines, Google search engine, Engineering Village, IEEE Explore, ACM Digital library, BTH Library, and Zotero were used to find information and knowledge. We also used some websites to get information about our research area. A literature review process helps to understand the system and its functioning and where its need more research.

4.2 Research Questions/ Study:

For conducting research in any field of, we need to develop research questions that maximize possible solutions. Research questions are necessarily to include in our research study because we are developing a new interface design for the mouse and eye tracking system in order to control articulated robot. To answer the research questions means to design an interface for the said application. To use hypothesis is irrelevant at this stage, but once a prototype is developed and functioning we can use the hypothesis for the interface design to check for its efficiency, correctness/accuracy, simplicity, user friendliness, etc., in terms of task completion during an experiment (usability study).

Our thesis contains three important research questions, first and second questions are
Primary while the third is secondary. The significance of the research
Questions are concluded based on the following study domains.

4.3 Research Design



Literature Review

RQ 1 has been answered

1. Software for interfacing a GUI with robot control operators.
2. Software link to a Tobii T60 eye tracking system, to support gaze-directed robot interaction.

This also provides base for experiment and in order to generate

Control Experiment

RQ 2 has been answered

1. Multi model Interface and robotic interface.

Questionnaire

RQ 3 has been answered

1. An evaluation of the different interaction models implemented, and comparison with a mouse-driven interface

Figure 15 Process overview of Research work

4.4 Related Work

This is fourth era of Eyetracking research [46] and research is on its way to develop more speedy, more reliable and easier to use eye tracking systems [47], [48]. As a result, the area of eye tracking applications is on the rise rapidly including both problem-solving and interactive applications. Gaze inputs are being used to develop many demanding graphical user interfaces for HCI as part of the development of interactive applications [49]. As a result, a small number of examples of eye tracking for human-robot interaction exist [50]. Human robot interaction is study of multiple disciplines [55]. Different applications of Eye tracking has started to grow up in diagnostic and interactive fields. Eyes inputs have been used in many Human computer interaction user interfaces [58].

There are some attempts of developing interfaces for human robot interaction using eye tracking [58].There are a number of ways which determine gaze direction has been developed in last 50 years, like reflection of light, electric skin potential, and contact lens. Mason was the first who give the idea of calculating eye gaze by using an IR LED. This Paper reports a robotic arm controlled by experimental algorithm of eye tracking. A wheel chair controlling interface was designed that uses human eye gaze; this interface consisted of five non-active regions and four active regions. Another very close work related to our proposed graphical user interface was done by [56].This graphical user interface was based on buttons .This graphical user interface has not enough accuracy for small menu buttons so the author decided to go with large menu buttons. Every button was responsible for controlling some specific joints. These buttons are for controlling and commanding the all joints of the robot. The user of this interface just sees the button to control the joint which was specific to said button. The authors of [56] used two windows systems, one for the control system and the other to see the state of the robot. The divided interfaces were referred to as the feedback region and the commanding region. Another robotic toy was on a similar basis. [57]

4.5 Interface Design

The major part of our thesis work is to develop an interface design to control the robot which will suitable and user friendly for people with motor disabilities. Final output of the research is in the form of a proposed user interface design for controlling an articulated robot arm using mouse and eye tracking system and comparison between these two. The design relates the information obtained by following modules;

- Experiment Results
- Research Study guiding principle

An articulated robot is that type of robot which is consisted on rotary joints to access its work space. These joint are attached each other for supporting them.

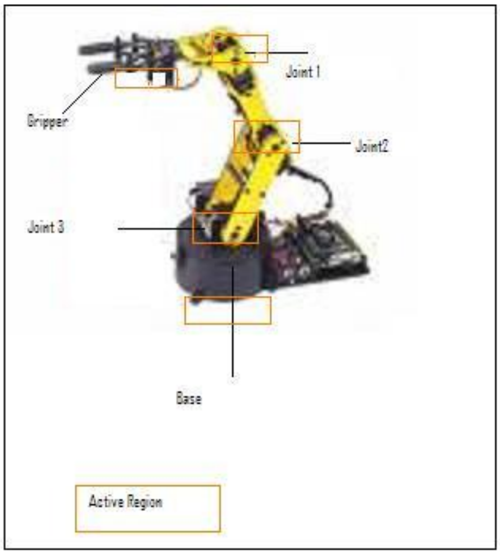


Figure 16 Articulated robot arm showing joints

4.5.1 **Robot Description**

Articulated Robot arm use used in our thesis project is consisted on five joints and each is controlled by servo motors attached as showing in figure 2.

4.5.2 **Arduino Micro controller Board for Robot**

A method is required to control the robot. In our case, we used an Arduino Diecimila microcontroller board. An ATmega 168 is used in this board. The board has 14 digital input/output pins, 6 of them are used as analog inputs and 6 are used as Pulse width Modulation (PWM) outputs, one 16 MHz crystal oscillator, one USB connection , one power jack , one ICSP header and one reset button. We simply use a USB cable to connect it with the computer.

“Diecimila” is an Italian word .The meaning of word “Diecimila” is 10000. This is the latest version of Arduino in USB series of board. It has many advantages over the previous versions.

Here is the total summary of the board

Microcontroller	ATmega168
Operating Voltage	5V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	16 KB (of which 2 KB used by boot loader)
SRAM	1 KB
EEPROM	512 bytes
Clock Speed	16 MHz

Table 2 Summary of Arduino board

- **Power**

There are two ways to give power to this board, by USB connection and another method is by Non USB (AC to DC adapter or battery). In our case, we used USB connection method to give power to the microcontroller board. PWR_SEL jumper is source of sending power.

- **Memory**

The Arduino board ATmega 168 has 16 KB of flash memory which is used for storing the code.2 Kb is required for a boot loader. SRAM requires 1 KB and EEPROM needs 512 bytes.

- **Input and Output**

The board has 14 digital pins for purposes of input or output. Pin Mode, Digital write, and Digital read functions are three different functions. 5 volts are needed to operate these functions. A maximum of 40 mA can be sent or received by each pin using a pull up resistor of 20-50 k Ohms. Each of these digital pins has its own special functions. The board has 6 analogue input pins. Resolution of each pin is 10 bits.

- **Digital Pins**

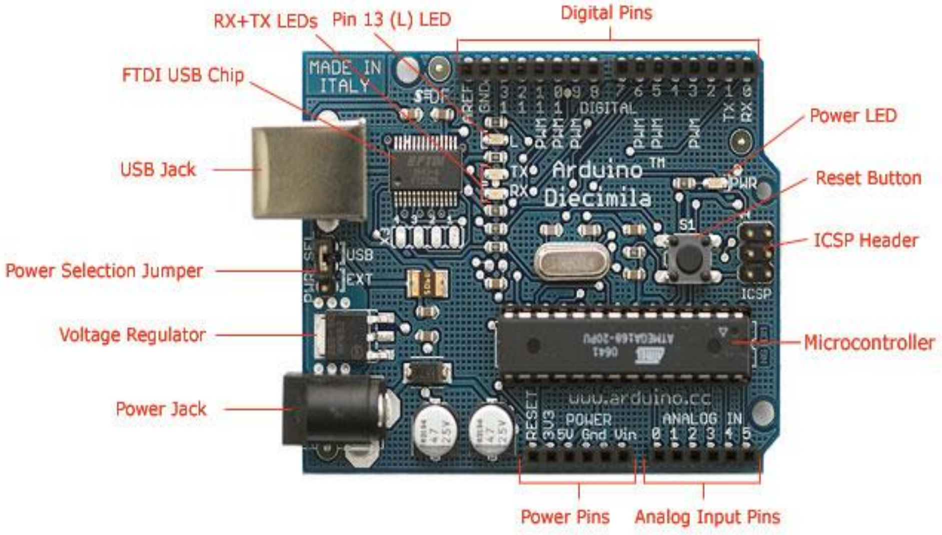
Serial 0 and 1: These pins are used for receiving and sending data (0(RX) for receiving data and 1 (TX) for sending data).
External Input 2 and 3: Value is changed from low to high by these pins.
Serial 3, 5, 6, 9, 10 and 11: For pulse width modulation, these six pins are used.
Serial 10 (SS), 11(MOSI), 12(MISO), and 13(SCK): To handle SPI communication, these pins are used.
LED 13: If value is high than LED is on and if value is low than LED is off.

Serial 10 (SS), 11(MOSI), 12(MISO), and 13(SCK): For handling SPI communication, these pins are used.

- **Analog Pins**

Serial I2C 4(SDA) and 5(SCI): For handling TWI communication, this pin is used.

Reset: For adding reset button, this pin is used



Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

Figure 17 Arduino board configuration [61]

4.5.3 Pulse width Modulation (PWM)

PWM is a technique that is used to provide the intermediate value of power between fully off and fully on. Different voltages can be obtained by varying the portion of the time that is used for on verses the portion of the time it uses for off. The duration of the on time is called pulse width. The proportion of the on time with the off time is called the duty cycle of the pulse. A 100% duty cycle is the pulse in full on. The microcontroller board generates PWM output voltages between 0 to 5V. A sample PWM signal with the Frequency of 50Hz is shown in the figure below [61].

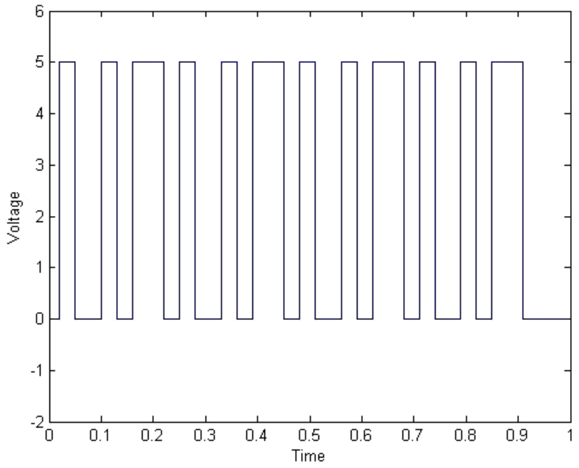


Figure 18 showing graph for pulse width modulation [61]

4.6 Validation

Validation of the research involves evaluating the effectiveness of the technology solution of using of gaze to control the articulated robot, via the proposed "User Interface Design". Validation is accomplished using a questionnaire given to the students who tested the system using both a mouse and Eyetracking. On the basis of answers collected from students we validate our result.

4.7 Questionnaires

Questionnaires are a well-established techniques for the collection of data and user's opinions. They are almost similar to interviews and are of two types [63].

- Closed Questionnaires
- Open Questionnaires

4.7.1 Closed Questionnaires

Questionnaire which consists of closed-ended questions is known as closed questionnaires. These questions have predefined answers, e.g. "YES", "NO" or multiple choice questions [63].

4.7.2 Open Questionnaires

Open questionnaires permit the user to respond in their own words [63]. In open-ended questions, different users provide different points of view and different styles of thinking according to their own choice and will.

We used (forced choice questionnaire) a type of closed questionnaire to evaluate results of our experiment which is shown in appendix 9.1.

4.7.3 Forced Choice Questionnaire

There exist many methods of forming survey questions, which help us to gather more action able data and forced choice questionnaire is also a type of closed questionnaire. Forced choice questions help to make the survey most respondent. In this way a definite opinion is shown by the user by selection a response option. When we form this type of questionnaire then neutral and do not know option is eradicated. This questionnaire is designed to force the respondent to show his / her attitude about said question, these question are usually written in the form of agree / disagree. Most of the survey research studies show that excluding of do not know and neutral options does not alter the percentage of specific response scale[66], [67].

5 THEORETICAL WORK/STUDY

5.1 What does Interface mean?

User interface is defined as way of communication between the users and the system. The purpose of the interface is to create ease of use and accessibility to human computer interaction. Interface design consists of seven different principals. Many interfaces existed and these interfaces are completed for user experiences. There are many types of interfaces with features like intelligence, adaptively and some interfaces are specific for interaction like command, graphical multimedia. Relative types of interfaces to our thesis are multimodal and robotic [62].

5.2 Multimodal Interfaces.

Multimodal interfaces are multimedia interfaces based on a “more is more” principle .They are controlled by different modalities, i.e. touch, sight, sound, speech. Interface techniques that have been combined for this purpose include speech and gesture, eye, pen input and speech. Multimodal interfaces can support more flexible, efficient and expressive means of human computer interaction, that are more akin to the multimodal human experiences in the physical world [62].

5.3 Robotic Interfaces

Robots have been with us for some time, most notably as characters in science fiction movies, but also playing an important role as part of manufacturing assembly lines, as remote investigators of hazardous locations such as nuclear power plants and bomb disposal and as search and rescue helpers in disasters [62].

5.3.1 Consol Interface

A console interface allows a human to control and navigate robots in remote environments using a combination of joystick and keyboard. This combination of joystick and keyboard controls acts together with camera and sensor based interaction. The focus has been on designing interfaces that enable users to effectively steer and move a remote robot with the aid of live video and dynamic maps. Domestic robots are appearing in our homes as helpers. Robots are being developed to help elderly and disabled with certain activities. Such as picking up and placing objects [62].

5.4 User Interface Design Process

User interface development involves different phases, which are used to define the interface [30] .Each interface depends upon its type of application. It can be described as below.

5.4.1 Requirements of Functionality

Functionality requirements include requirements for software, the whole system and its components. User input and output behavior is also mentioned in this part. It contains processing, relevant calculations, technical details of the system. “What system is expected to accomplish” is founding question answered by functional requirements.

5.4.2 User Analysis:

User analysis is a very essential aspect for success of an interface and concerns analyzing the following problems.

When the product is totally dependent on the user, its role becomes very important, while if the product is specific user independent, it reduces complexity in the user analysis.

5.4.3 Prototype

Prototyping is the development of an entity having a logical or physical existence based on an idea or concept. A prototype is a complete product, developed at an intermediate stage in the system development life cycle. A prototype helps to confirm the “does” or “don’ts” for use, cause, functionality, user expectation and usability tests. An interface prototype allows users to experience the interface directly [63]. It is vital to include this in product development to analyze the conceptual model working in reality and its limitations.

5.4.4 Usability Testing

The purpose of usability testing is to confirm the product development verification and validation through performing user tests. The main motto and objective of usability testing is to achieve user satisfaction and confidence. Usability evaluation is carried out through simple methods, i.e. questionnaires, video recording, code testing and provisioning [63].

5.4.5 Graphical User Interface Design (GUI)

Mostly graphical user interface is described as the better options to command line interface usability. GUI interface designers give more control to users, functionality, design options, length and size of the screen, attracting users understanding and skills, to give the user adaptation and learning [63].

.

5.5 Programming Language used

We used Java to control the articulated robot. There is some motivation behind choosing Java Platform.

5.5.1 Java is Simple

Java has a rich feature set. Syntactic sugar or unnecessary features are removed. There are only about eighty pages for the Java language specification. Java has more functionality than C. A learner can become fluent quickly because it is small.

5.5.2 Java is Platform Independent

Byte-code is an intermediate form in the compilation of java code. A Java interpreter (special program) reads the byte code and executes the corresponding native machine instructions. Byte codes remain the same on all platforms. If you want to run a program on new platform, run it with compatible interpreter.

5.5.3 Java is Safe

Java provides secure execution of code across a network even when source is UN trusted and possibly malicious. Expected and unexpected errors are handled by robust exception handling mechanisms of java.

5.5.4 Java is High Performance

Java byte code is compiled using a "just-in-time compiler" very speedily. Several companies are also doing research on native-machine-architecture compilers for Java.

5.5.5 Java is Multi-Threaded

Java is multi-threaded. A single Java program can have many different processes executing independently and continuously. Different threads of a single java program can execute independently and continuously. Different threads can get equal time from the CPU with little effort [64].

6 EXPERIMENT RESULT / ANALYSIS

We develop two different interfaces.

6.1 Interface 1 Using Eyetracking

There are 3 stages of Eye tracker based interface.

6.1.1 First Stage

In first stage there is implementation of eye tracking. The *Tobii Eye Tracker Components API* (*TetComp*) is a programming environment that provides real-time high-precision gaze from Tobii eye tracker hardware. It provides a high level of abstraction.
The eye tracker machine consists of a client and a server (local server) .There is a monitor that is used to take input. A user can rotate his eyes, while sensors on the Eyetracker monitor record eye movements .This data (recorded movements of eyes) is sent to the server machine that determine the X-AXIS and Y-AXIS position of eyes and convert it into floating point numbers. It also writes this data in a text file.

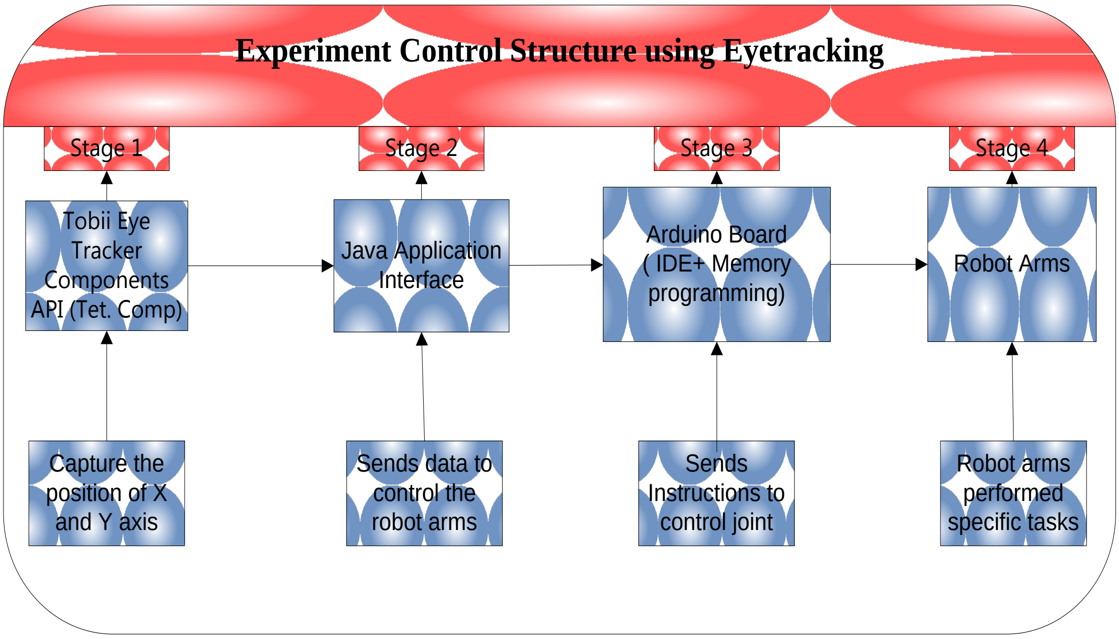
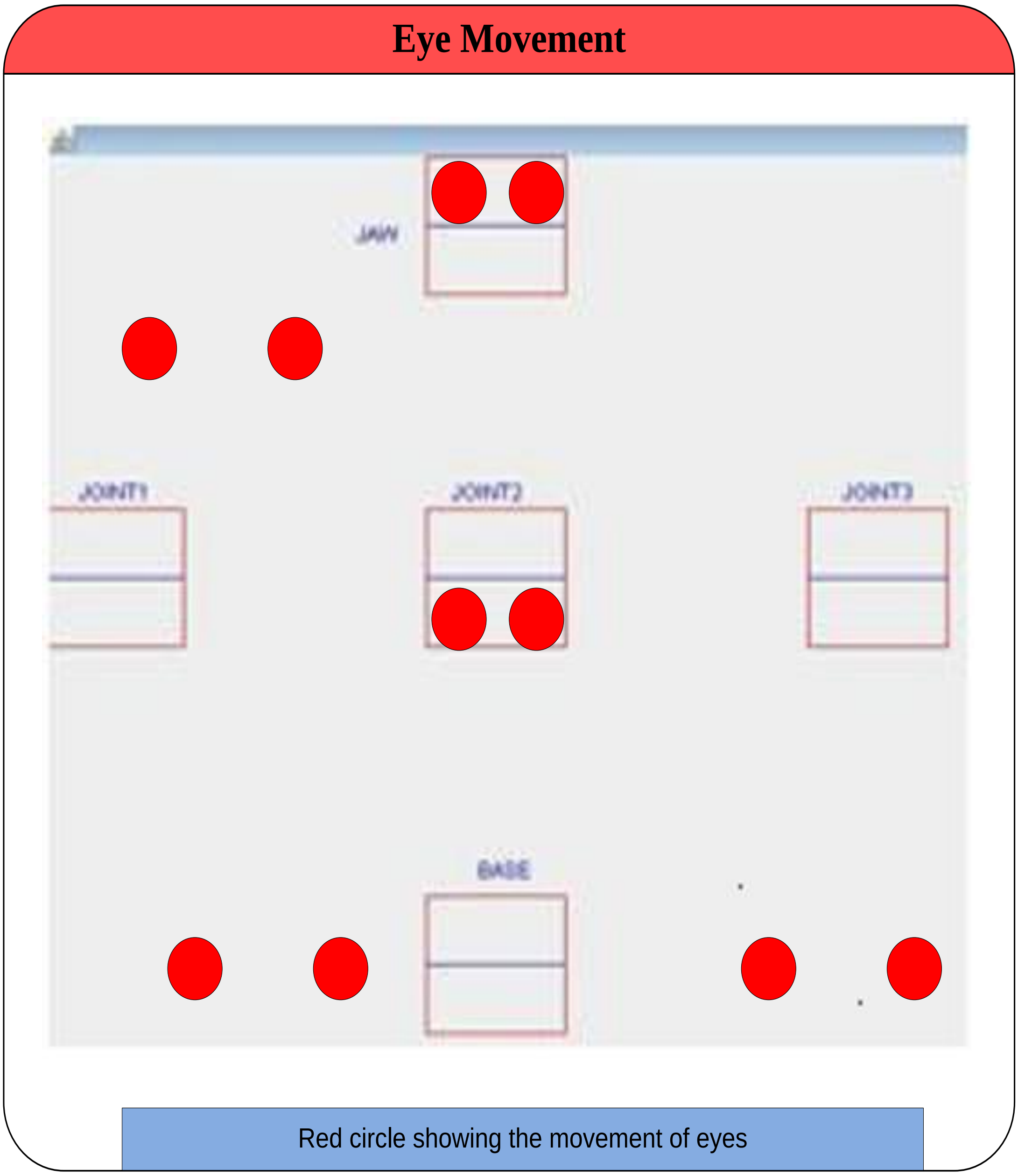

Experiment Control Structure using Eyetracking
Stage 1
Stage 2
Stage 3
Stage 4
Tobii Eye Tracker Components API (Tet. Comp)
Java Application Interface
Arduino Board (IDE+ Memory programming)
Robot Arms
Capture the position of X and Y axis
Sends data to control the robot arms
Sends Instructions to control joint
Robot arms performed specific tasks

Figure 19 Flow of data of three stages

6.1.2 Second Stage

In a second stage, a java application reads data from the text file and converts this data into integer form. The Java application uses this data (written in text) for two purposes.
Java application uses java APIs (draw Circle) to draw the position of the eyes on different positions at different interval of time (as shown in Figure 6.2) on the basis of the data from the text file (X-Axis and Y-Axis). Secondly, it determines the movement of joints of the robot according to the position of the eyes.


Eye Movement

Red circle showing the movement of eyes

Figure 20 Graphical movements of eyes of users

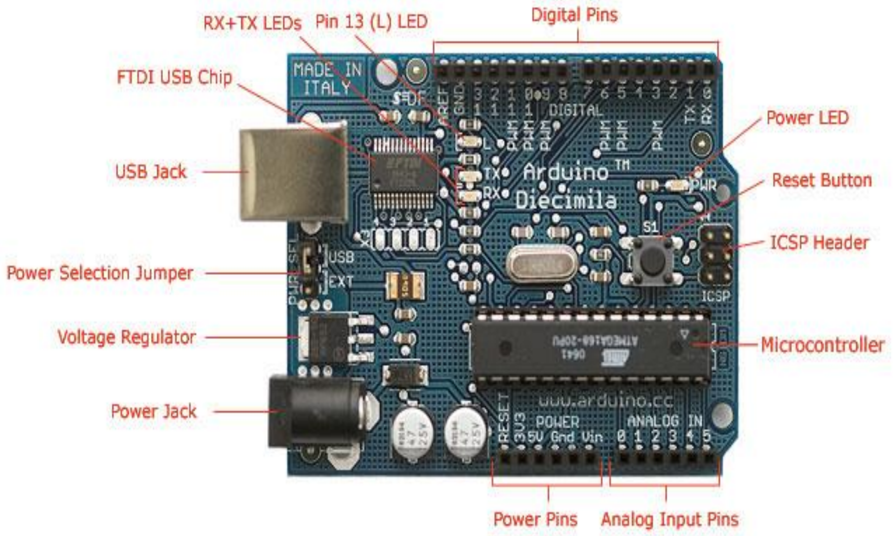
Because the robot has five different joints (figure 19) we have drawn five different rectangles on the interface (figure 23). Each rectangle is associated with a different joint of the robot. The name of a rectangle and the corresponding joint of the robot are the same (Base, Gripper, Joint 1, 2, and 3) as shown in figures 19 and 23.

We have done programming in such a way that if the eyes of the user focus on the upper part of a rectangle, then the relevant joint can move from 0 to a 90 degree angle. Likewise if the eyes of the user focus on the lower part of a rectangle, then the relevant joint can move from 90 to 180 degrees.

The java application maintains byte b [] of size 3. First byte b[0] holds a joint number to be moved (0 for Base , 1 for Gripper , 2 ,3,4 for Joint 1,2,3). When the eyes of the user focus on any relevant rectangle among 5 different rectangles, then digit (0 to 4 as a joint number) will be assigned to b [0]. Similarly, if the eyes of the user focus on the upper part of a rectangle, then angle (0) is assigned to b [1] and the last angle (90) is assign to b [2]. If the eyes of the user focus on the lower part of a rectangle, then angle (90) is assigned to b [1] and the last angle (180) is assigned to b [2]. This byte is sent to the memory of the Arduino board.

6.1.3 **Third Stage**

All input data is stored in a "Positioning Controller", that is the memory for a specific servo.



Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

Figure 21 Specification of arduino board

As the picture shows, the Arduino board has digital pins. Different joints of the robot are given power by these pins. We have controlled these pins according to our needs by java application and Arduino board IDE. In our case we have given high voltage to a pin on basis of data provide by b [0] using serial communication.

We have also done coding in the IDE of the Arduino board (3rd stage). Programming in this stage has while () loop and waits for data from the java application through the serial port. When it receives data from the java application (stage 2nd), this data has the form of a byte array b [] of size 3.The first byte tells the joint number to be moved. A programming instruction sends PWM signal to the pin of the arduino board (through attach (pin number) method) that moves to specific joint. Second byte b [1] and third byte b [2] specify the first and last angle to which a specific joint is to be moved. Programming instruction use this data for movement.

6.2 2nd Interface Using Eyetracking

Its first and third stage is similar to first interface.

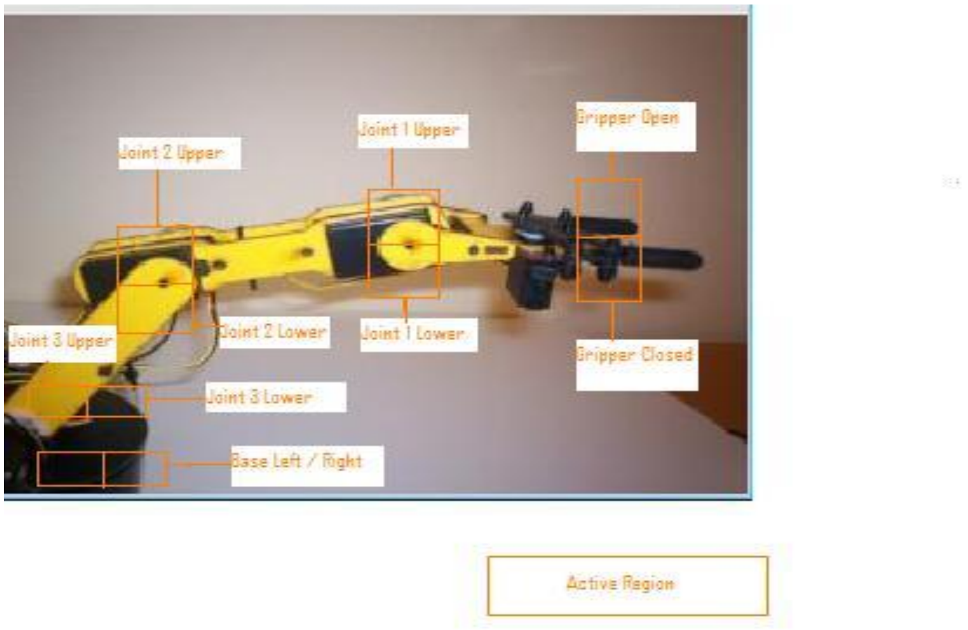


Figure 22 2nd interface

The difference between Figure 21 and Figure 19 is that it has a picture with rectangles showing the active regions of the interface. Because the robot has five different joints, we have drawn five different rectangles on the interface .Each rectangle is associated with a different joint of the robot. The name of the rectangle and the joints of robot are the same (Base, Gripper, Joint 1, 2, 3).

We have programmed the application in such a way that if the eyes of the user dwell in the upper part of the rectangle, then relevant joint can move from 0 to 90 degrees. Likewise, if the eyes of the user dwell in the lower part of the rectangle then the relevant joint can move from 90 to 180 degree.

The java application maintains **byte** *b[] of size 3*.The first byte *b[0]* holds the joint number (0 for Base , 1 for Gripper , 2 ,3,4 for Joint 1,2,3). When the eyes of the user dwell on any relevant joint among the 5 different joints then digit (0 to 4 as a joint number) will be assigned to b *[0]*. Similarly, if the eyes of the user dwell on the upper part of the rectangle, then angle (0) is assigned to *b [1] and last* angle (90) is assigned to *b [2]*. If the eyes of the user dwell on the lower part of the rectangle then angle (90) is assigned to *b [1] and last* angle (180) is assign to *b [2]*.

6.3 Working of Interface1 using Mouse.

This interface has 3 stages .Its second stage is similar to the 3rd stage of the eye tracker base interface shown in figure 6.5, so we will not describes it again.

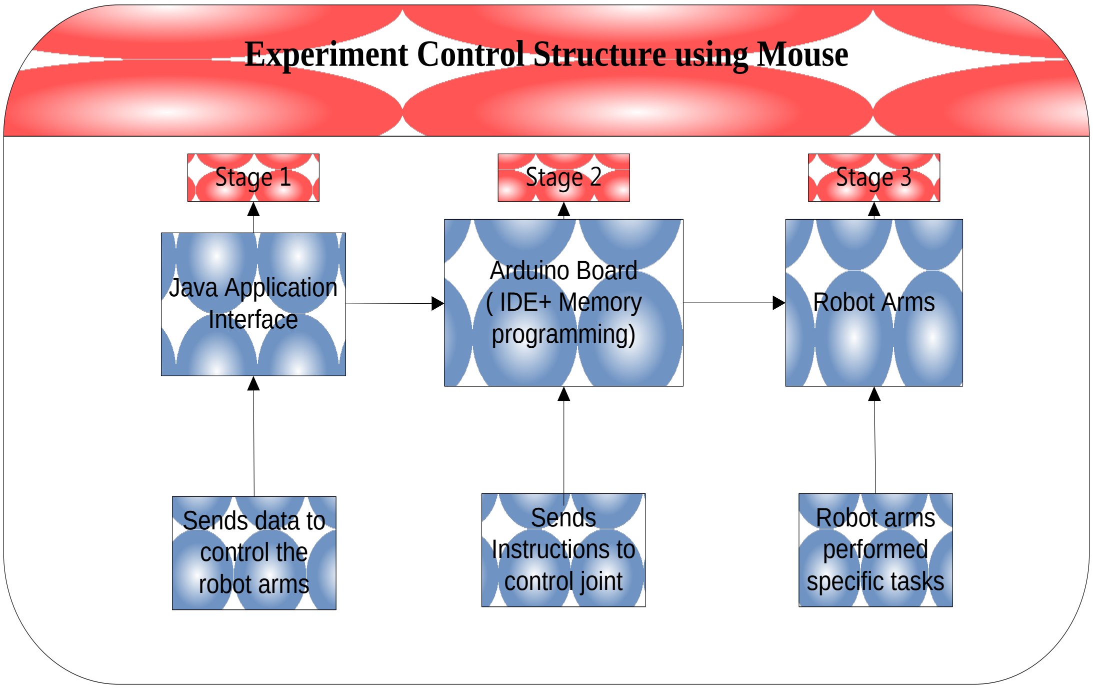

Experiment Control Structure using Mouse
Stage 1
Stage 2
Stage 3
Java Application Interface
Arduino Board (IDE+ Memory programming)
Robot Arms
Sends data to control the robot arms
Sends Instructions to control joint
Robot arms performed specific tasks

Figure 23 Mouse based interface

Its first stage is different so here is its description. The Java application draws the following graphical user interface. This application uses the following methods to draw the GUI.

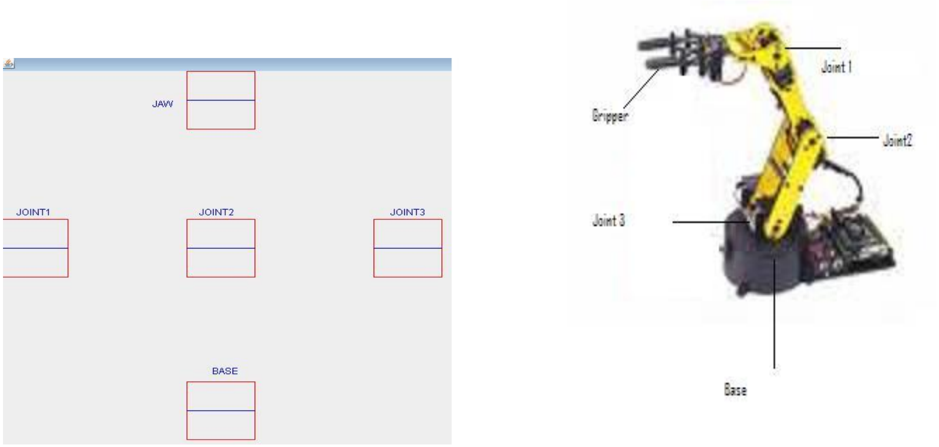


Figure 24 Mouse Based GUI and Robot.

SetColor (Color.red);
DrawRect(x, y, width, length);
SetColor (Color);
DrawLine(x, y, ending x, ending y);

As shown in Figure 23, there are 5 different rectangles, namely (Gripper, Base, joint 1, joint2, and joint3). These rectangles have two parts. The upper part is used to move the relevant joint from 0 to 90, while the lower part is used to move the relevant joint from 90 to 180. As shown in the picture, there are 5 different joints of the robot, namely (Gripper, Base, joint 1, joint2, and joint3). These joints can move from 0 to 180 degree as needed. The gripper can grip any solid object as it moves from 0 to 180 degrees. The other joints can move likewise.

Later on, this java application sends data to the arduino board that controls the robot. Details of data are given below.

Public void write (byte *b []*) throws IOException

Parameters *b* the data.

Description

A specified byte array (b.length bytes) is written to this output stream. Actually, the java application maintains byte *b[] of size 3*.First byte *b[0]* holds joint number(0 for Base , 1 for Gripper , 2 ,3,4 for Joint 1,2,3). When the user clicks on any relevant joint among 5 different joints then digit (0 to 4 as a joint number) will be assigned to b *[0]*. Similarly, if the user clicks on the upper part of a rectangle then angle (0) is assigned to *b [1] and last* angle (90) is assigned to *b [2]*. If the user clicks on the lower part of the rectangle, then angle (90) is assigned to *b [1] and last* angle (180) is assigned to *b [2]*. Later on this data will be passed to the second stage (the second stage is similar to the third stage of the first interface and its description is already given).

6.4 Working of the Interface 2 Using a Mouse

This is another interface shows in figure 21 used to control the robot.

Through java programming, we have included a picture of the robot (as shown in Figure 19).White parts show the active regions .By clicking these active regions, we can control the relevant joint .Whenever we click the mouse the click event returns X and Y Axis position. If the user clicks on the upper part of a rectangle then the relevant joint can move from 0 to 90 degrees. Likewise, if the user clicks on the lower part of a rectangle then the relevant joint can move from 90 to 180 degrees.

The java application maintains byte b [] of size 3. The first byte b[0] holds the joint number(0 for Base , 1 for Gripper , 2 ,3,4 for Joint 1,2,3). When the user clicks on any relevant joint among the 5 different joints then digit (0 to 4 as a joint number) will be assigned to b [0]. Similarly, if the user clicks on the upper part of a rectangle then angle (0) is assigned to b [1] and last angle (90) is assigned to b [2]. If the user clicks on the lower part of a rectangle then angle (90) is assigned to b [1] and the last angle (180) is assigned to b [2].Later on this data will be moved to the arduino board (2nd stage) through java serial communication with the write (byte b []) method.

6.5 Comparison of Mouse-Based and Eye Tracker Interfaces

Time which we mentioned here is time to complete a task. Servo motors (used in all joints of robot) can move from 0 to 180 degree. Task is that base moves from 0 to 180 degree. Robot picks an object from 0 degree and places it at 180 degree.

Users have used both interfaces. In mouse based interface, when he clicks on any part of rectangle (either lower or upper) than robot would move. Movement of robot depends on which part, user has clicked. It will take time to complete this process. (Time is given in table 3). In eye tracking based interface, when eyes of users focus on any part of rectangle (either lower or upper) than robot would move. Movement of robot depends on which part, user has focused. It will take time to complete this process (time is given in table 4).

Ten Different students have used both interfaces and performance data is given below.

6.5.1 Reading with Mouse

User	Time(second)
A	9.00
B	10.00
C	11.00
D	9.00
E	11.00
F	11.00
G	9.00
H	10.00
I	9.00
J	11.00

Table 3 Number of Users and their time Consumption in seconds

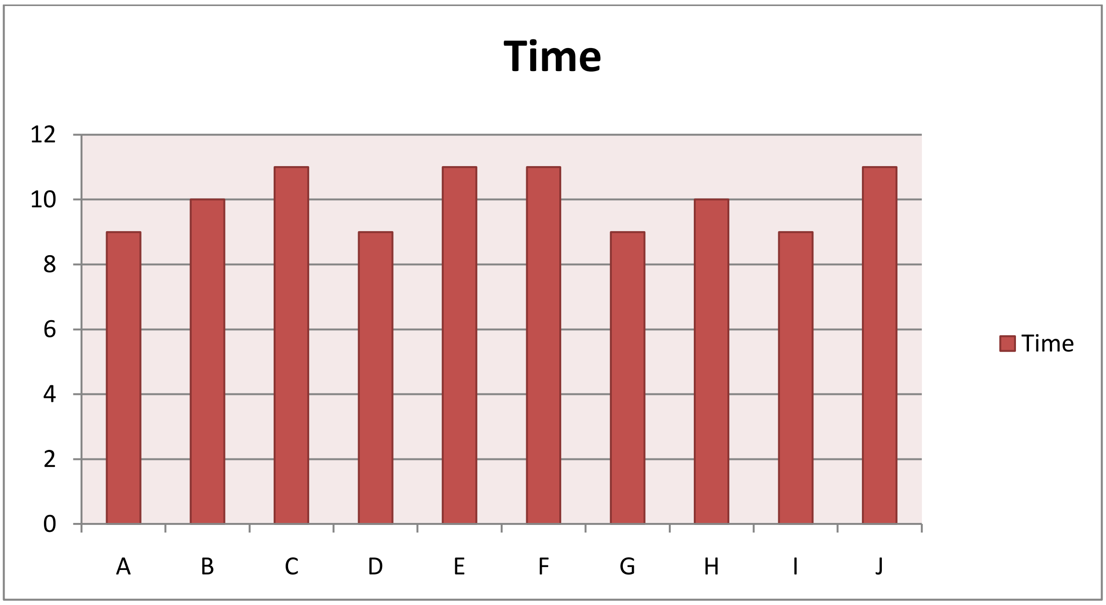


Figure 25 graph showing time using mouse

We have collected reading from 10 students. They have used our implementation. They have used mouse bases implementations .When all users click on interface than relevant joints take varying time(seconds) to complete its movement .Different users have different time interval when they were using it as shown in figure 19 and 23.

6.5.2 **Reading using Eye Tracker**

User	Time + Delay Time
A	9.04
B	10.04
C	11.04
D	9.04
E	11.04
F	11.04
G	9.04
H	10.04
I	9.04
J	11.04

Table 4 Time used in seconds to complete task by Eyetracking

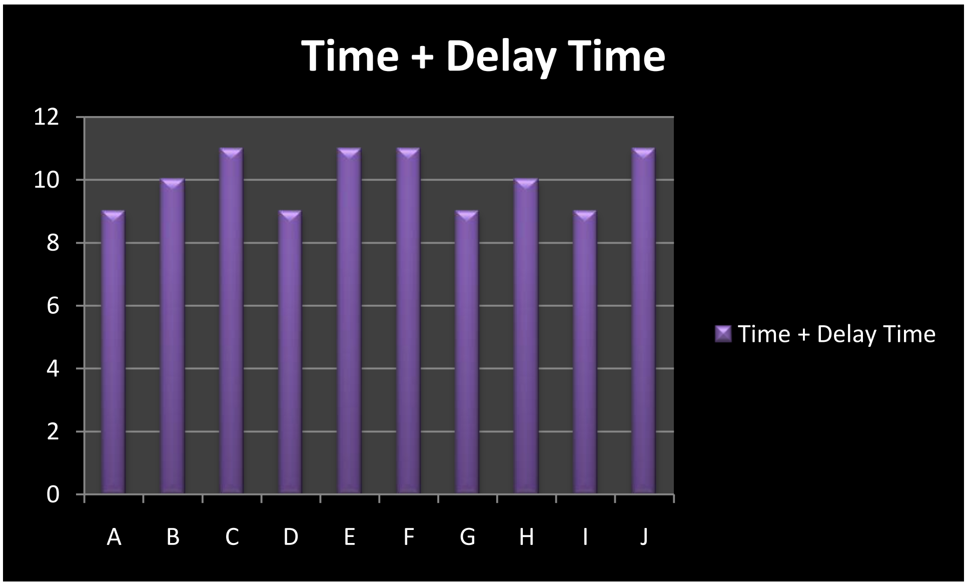


Figure 26 graph showing time Eyetracking

6.6 Comparison between Mouse and Eyetracking

To compare the results of experiments we select statistical analysis. The motivation behind taking mean and standards deviation was to evaluate the result of both Eyetracking and mouse. Mean is a statistical analysis which measures the average values or average errors of certain experiments. In our case we have to compare the time of both Eyetracking and mouse, so we used this statistical analysis and then to modify our calculations more precisely effective we take the standard deviation which minimizes the square of the error. Hence our result of standard deviation of using mouse and Eyetracking is approximately same, which strengthen our objective of replacing Eyetracking instead of mouse.

User	Time for Eyetracking + Delay Time	Time for Mouse
A	9.04	9.00
B	10.04	10.00
C	11.04	11.00
D	9.04	9.00
E	11.04	11.00
F	11.04	11.00
G	9.04	9.00
H	10.04	10.00
I	9.04	9.00
J	11.04	11.00

Table 5 showing time difference between Eyetracking and Mouse

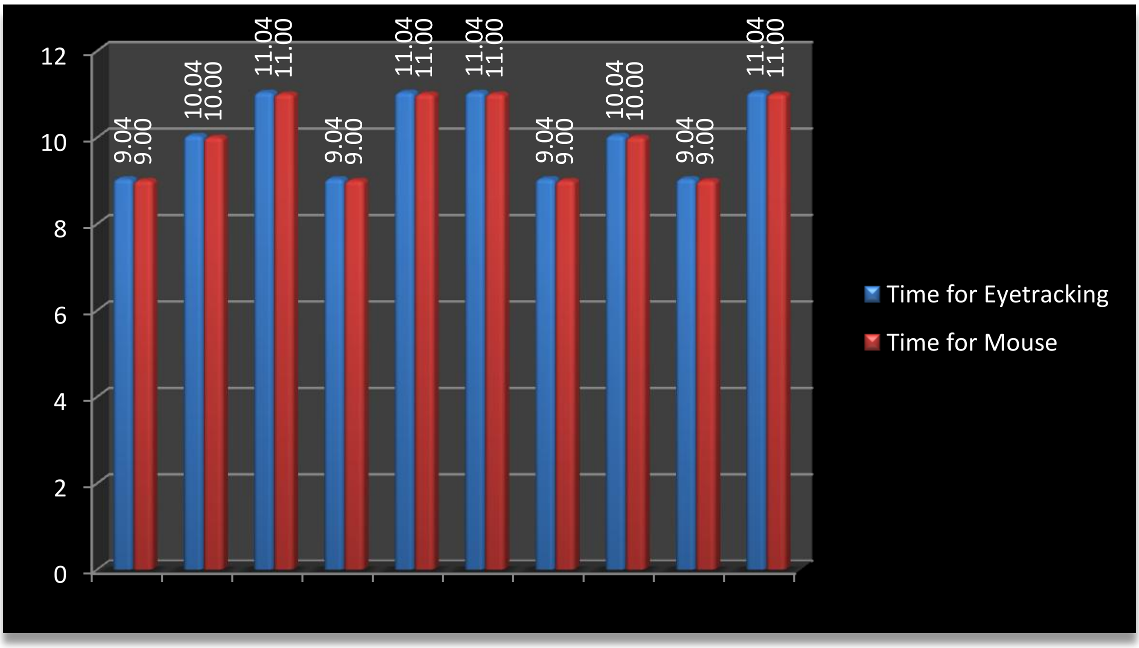


Figure 27 Showing difference of time between mouse and Eyetracking

We calculated the mean of Mouse based interface and mean of Eyetracking interface

Mean of Mouse	10.00
Mean of Eyetracking	10.04

Table 6 showing mean difference of time between Mouse and Eyetracking

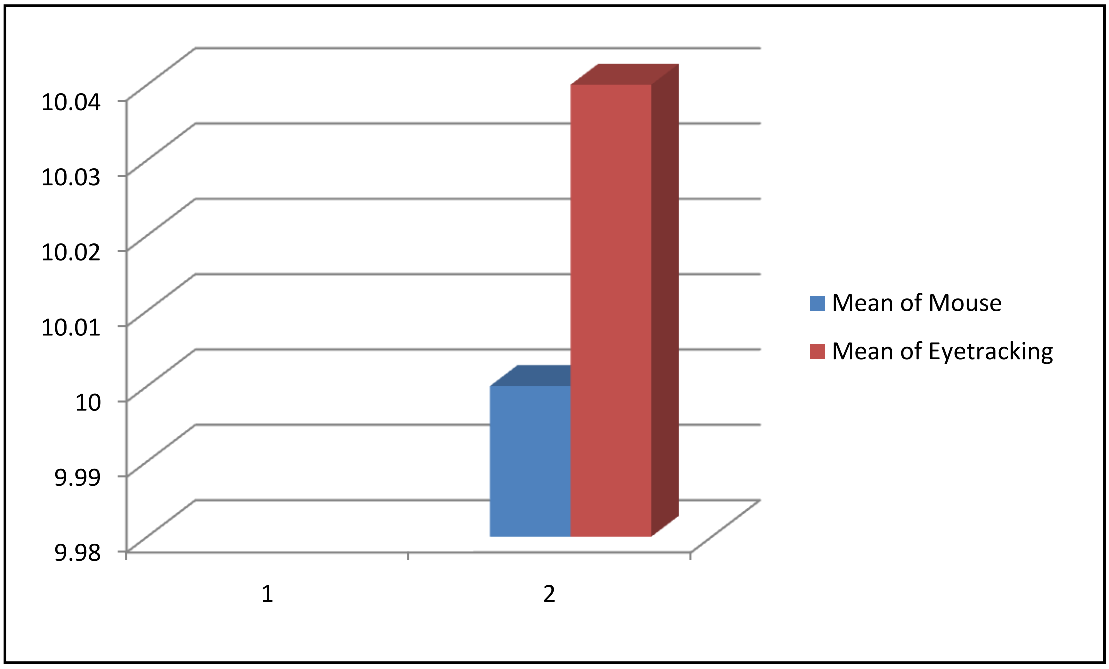


Figure 28 Showing mean difference of time between mouse and Eyetracking

We calculated the Standard Deviation of Mouse based interface and mean of Eyetracking interface

Standard Deviation of Mouse	0.942809042
Standard Deviation of Eyetracking	0.942809042

Table 7 Showing standard deviation difference of time between Mouse and Eyetracking

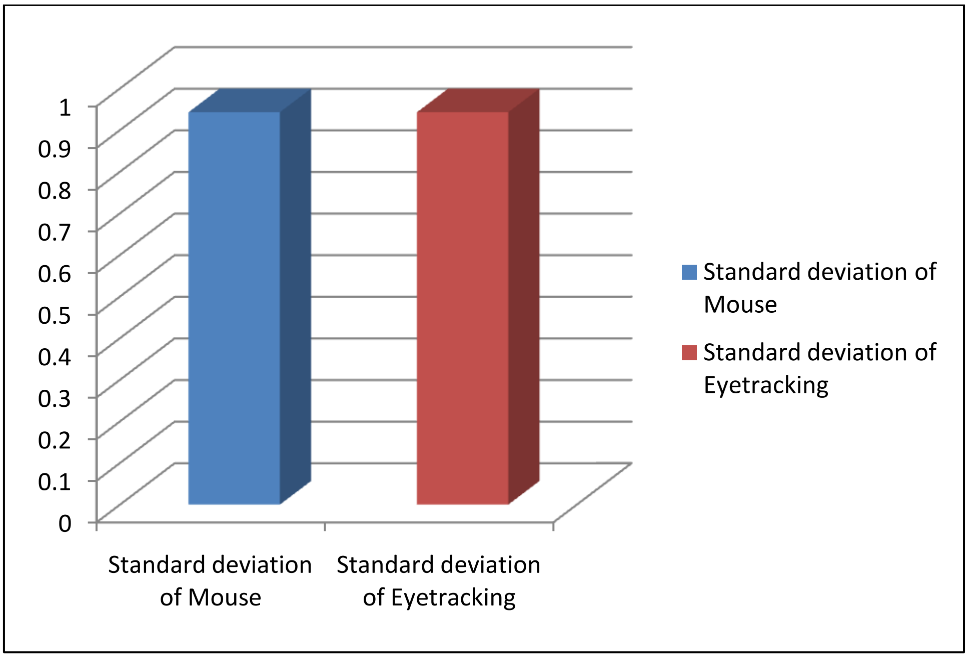


Figure 29 Showing Standard deviation difference of time between mouse and Eyetracking

We have collected readings from 10 different students. The entire student are studying different course at master level in Blekinge Institute of Technology. They have used the Eye Tracker based implementation .When they used it, it takes varying time (in seconds) as shown in figures 25 and 26.

There is delay time of 0.04 seconds. The Eyetracker consists of a client and a server machine (local server). It has a monitor with sensors that is used to track the eye movements of a user .This data is moved to the server machine that calculates the X-AXIS and the Y-AXIS positions of the eyes of users. This process takes 0.01 seconds. We have used built-in APIs of Tobii Company (it has its own IDE). We have used a text file for interaction between the java application and the built-in APIs from Tobii. This process takes 0.03 seconds. We have also used the JAVA application for interaction with Arduino Board that controls the robot.

	Mean	Standard Deviation
Mouse	10.00	0.942809042
Eyetracking	10.04	0.942809042

Table 8 Showing comparison between mouse and Eyetracking for mean and standard deviation

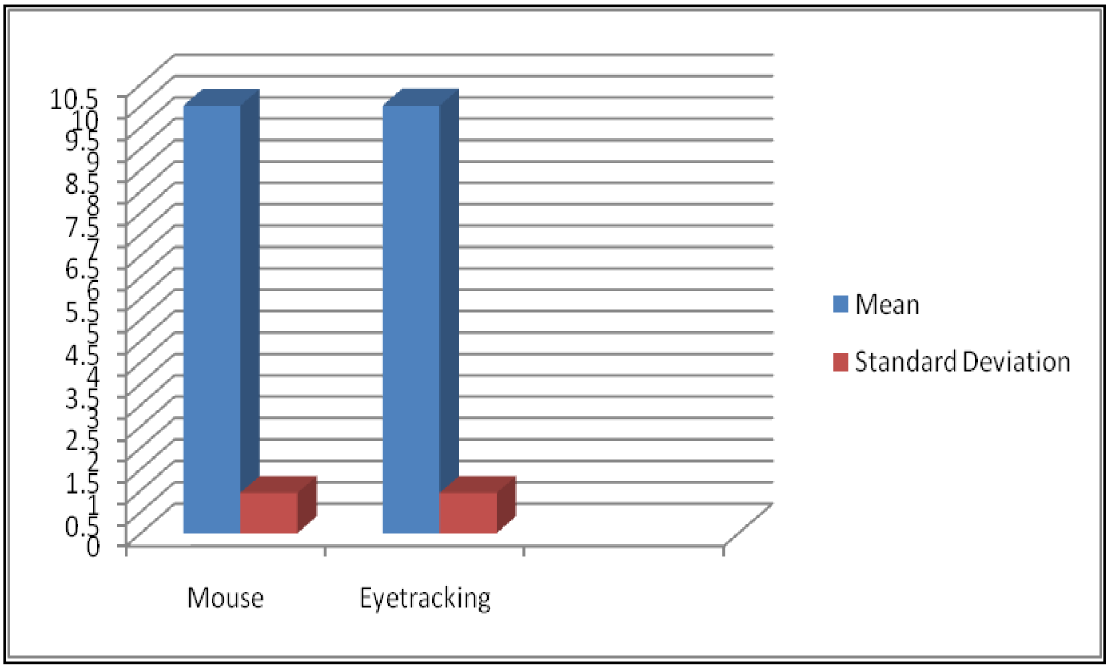


Figure 30 Showing comparison of mean and Standard deviation for both Mouse nad Eyetracking

Which is better?

The experimental result shows that working and control of a mouse is little better than an eye tracker on these interfaces. One most noticeable difficulty in the application of eye tracking on these interfaces is keeping focus on different rectangles.

According to the calculations from experimental results, when we used mouse the mean difference is 10.00 seconds where as the mean difference is 10.04 seconds when we used the Eyetracker instead of mouse. Finally we take standard deviation of both mouse and Eyetracking which is approximately equal and reached at the result that we can replace Mouse with eye tracking

7 CONCLUSION AND FUTURE WORK

7.1 Conclusion

The major purpose and objective of our thesis is to develop interfaces to control the articulated robot arms using an eye gaze tracking system, to allow users to use their arms for other tasks or to facilitate telerobotic manipulation for users without free use of their arms. After developing these interfaces the main focus was to control the robot arms by using a Tobii Eyetracker provided by BTH. First we apply these interfaces to control an articulated robot arm with a mouse. After that we modified these interfaces to be driven by Eyetracking. The interfaces that rely on gaze as the only means of input must support this style of interaction, and be capable of making a distinction between glances when the user is just looking around and fixations with the intention to start a function. There is a problem in gaze-directed control known as MIDAS TOUCH. To overcome this problem, the concept of DWELL-TIME is used, which is a common solution to deal with MIDAS TOUCH. We controlled the articulated robot arm using dwell time to facilitate motor disabled people who cannot perform specific tasks with their hands. We compared the performance time of basic control tasks using the mouse and gaze-directed control. It is obvious that the mouse is bit faster than eye tracking in performing the tasks, but on the other hand eye tracking has clear and decisive advantages over the mouse for those people who are unable to use a mouse since they can perform different tasks using their eyes as an input instead of the mouse.

To validate our research, 10 students operate these two interfaces using both mouse and Eyetracking and found that it is a milestone for motor disable users by using their eyes as an input instead of hands. We performed graphical comparison for time consumption of mouse and Eyetracking.

7.2 Future Work

We have controlled only robot arm. We can easily extend it to implementation of full body robot. Full body robot can perform all functionality like walking, gripping the solid objects etc like human being. We can implement a robot that can see, talk and listen by ***java APIs.*** Such robot can detect the edges, can change their direction, can respond, can listen instruction etc.

In nut shell, since robotics field is so vast so number of new ideas can be introduced.

8 REFRENCES

[1] Lorenzo Sciavicco, Bruno Siciliano “Modeling and control of robot manipulators” 2nd Edition. pp. 2-3.

[2] Jones R.E. Fitts, P.M. and J.L Milton. Eye movements of aircraft pilots during instrument landing approaches. Aeronautical Engineering Review, 1950.

[3] H. Hartridge and L.C. Thompson. Methods of investigating eye movements. British Journal of Ophthalmology, vol.32, pp.581-591, 1948.

[4] Robert J.K. Jacob, Keith S. Karn. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises, pp.2.

[5] Moran T.P. Card, S.K. and A.P. Newell. The Psychology of Human-Computer Interaction. Erlbaum, Hillsdale NJ, 1983.

[6] B. Shackel. Note on mobile eye viewpoint recording. Journal of the Optical Society of America, vol.50, pp.763-768, Aug 1960.

[7] A. L. Yarbus. Eye Movements and Vision, Plenum Press, 1967.

[8] L.R. Young and D. Sheena, Survey of eye movement recording methods, Behavior research methods, instruments and computers, 7(5), , pp.397-429, 1975.

[9] J.F. Mackworth and N.H. Mackworth. Eye _xations recorded on changing visual scenes by the television eye-marker. Journal of the Optical Society of America, 48(7), 1958, pp.439-445.

[10] N.H. Mackworth and E.L. Thomas. Head-mounted eye-marker camera, Journal of the Optical Society of America, vol.52, pp.713-716, June1958.

[11] C. Ware and H. T. Mikaelian. An evaluation of an eye tracker as a device for computer input. In Proceedings of the ACM CHI + GI-87 Human Factors in Computing Systems Conference, pages 183-188.

[12] J.L. Levine. An eye-controlled computer, Technical report, IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y., 1981.

[13] Javal, E., (1878/79) Essai sur la physiologie de la lecture. Annales d'Oculistique.

[14] L. Mazo M. Barea, R. Boquete and E. Lopez. System for assisted mobility using eye movements based on electrooculography. IEEE Trans on Neural Sys. and Rehab. Eng., 10(4):209{218, December 2002.

[15] J. Villanueva A. Cabeza R. Goni, S. Echeto. Robust algorithm for pupil-glint vector detection in a video-oculography Eyetracking system. In ICPR. Proceedings of the 17th International Conference on Pattern Recognition, volume 4, pages 941{944, 23-26 Aug 2004.

[16] M. Chaumont and W. Puech. 3d face model tracking based on a multi resolution active search. In VCIP'2007, Visual Communications and Image Processing, volume 6508, January 2007.

[17] D. DeMenthon and L.S. Davis. Model-based object pose in 25 lines of code. IJCV, International Journal of Computer Vision, pages 335-343, 1995.

[18] H.J. Rae, R.and Ritter. Recognition of human head orientation based on artificial neural networks, IEEE Transactions on Neural Networks, 1998.

[19] H. Hartridge and L.C. Thompson. Methods of investigating eye movements, British Journal of Ophthalmology, 32:581{591, 1948.

[20] T. Sengupta and O.J. Jeng. Eye and mouse movements for user interface design, Proceedings of IEEE 29th Annual Bioengineering Conference, pages 1{2, March 2003.

[21] R. Jacob. Eye tracking in advanced interface design, Oxford University Press, 1995.

[22] M. A. Just and P. A. Carpenter, Eye Fixations and cognitive processes, Cognitive Psychology, 1976.

[23] R.J.K. Jacob. The use of eye movements in human-computer interaction techniques: what you look at is what you get. ACM Transactions on Information Systems - Special issue on computer - human interaction, pages 11-18, 1991.

[24] E. Jonsson. If Looks Could Kill An Evaluation of Eye Tracking in Computer Games, PhD thesis, Department of Numerical Analysis and Computer Science, 2005.
[25] C. Sennersten. Eye movements in an Action Game Tutorial, PhD thesis, Department of Cognitive Science, 2004.
[26] D. Bavelier and S. Green. Action video games modify visual selective attention. Nature, 423:534{537, 29 May 2003.
[27] C. Chan C. Yeh M. Lin, C. Huan and C. Chiu, Design of a computer game using an eye-tracking device for eyes activity rehabilitation. Optics and Lasers in Engineering, 42(1):91108, July 2004.

[28] N. Itti, L. Dhavale and F. Pighin. Realistic avatar eye and head animation using a neurobiological model of visual attention. In B. Bosacchi, D. B. Fogel, and J. C. Bezdek, editors, Proc. SPIE 48th Annual Inter-national Symposium on Optical Science and Technology, volume 5200, pages 64-78, Bellingham, WA, August 2003. SPIE Press.
[29] R. J. Peters and L.Itti. Computational mechanisms for gaze direction in interactive visual environments. In Proc. ACM Eye Tracking Research and Applications, pages 27{32, March 2006.
[30] S. Yan and N. Seif El-Nasr. Visual attention in 3d video games. In Proceedings of the 2006 ACM SIGCHI international conference on Advancesin computer entertainment technology, number Article n 22, Hollywood, California, 2006.

[31] Abdallahi Ould Mohamed and Matthieu Perreira Da Silva. A history of eye gaze tracking. Vincent Courboulay. December 18, 2007
[32]. L. Young and D. Sheena, “Methods & designs: survey of eye movement recording methods,” Behav. Res. Methods Instrum. 7(5), pp. 397–429, 1975.
[33] A.T. Duchowski, *Eye tracking methodology [Elektronisk resurs]: theory and practice*, London: Springer, 2007.
[34]. D. Robinson, “A method of measuring eye movements using a sclera search coil in a magnetic field,” IEEE Transactions on Biomedical Engineering 10, pp. 137–145, 1963.
[35].. A. Kaufman, A. Bandopadhay, and B. Shaviv, “An eye tracking computer user interface,” in Proceedings of the Research Frontier in Virtual Reality Workshop, pp. 78–84, IEEE Computer Society Press, 1993.
[36].. J. Reulen, J. Marcus, D. Koops, F. de Vries, G. Tiesinga, K. Boshuizen, and J. Bos, “Precise recording of eye movement: the iris technique, part 1,” Medical and Biological Engineering and Computing 26(1), pp. 20–26, 1988.
[37]Chris Weigle and David C. Banks.,” Analysis of Eye-Tracking Experiments performed on a Tobii T60”. University of Tennessee/Oak Ridge National Lab, Joint Institute for Computational Sciences.
[38]. Y. Ebisawa and S. Satoh, “Effectiveness of pupil area detection technique using two light sources and image difference method,” in Proceedings of the 15th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, A. Szeto and R. Rangayan, eds., pp. 1268–1269, (San Diego, CA), 1993.
[39]. A. Tomono, M. Iida, and Y. Kobayashi, “A TV camera system which extracts feature points for non-contact eye movement detection,” in Proceedings of the SPIE Optics, Illumination, and Image Sensing for Machine Vision IV,1194, pp. 2–12, 1989.
[40]. K. Kim and R. Ramakrishna, “Vision based eye gaze tracking for human computer interface,” in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, (Tokyo, Japan), 1999.
[41]. Y. Ebisawa, “Unconstrained pupil detection technique using two light sources and the image difference method,” Visualization and Intelligent Design in Engineering and Architect II 15, pp. 79–89, 1995.

[42]. Y. Ebisawa, M. Ohtani, and A. Sugioka, “Proposal of a zoom and focus control method using an ultrasonic distance meter for video-based eye-gaze detection under free-hand condition,” in Proceedings of the 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1996.

[43]. D. Yoo, J. Kim, B. Lee, and M. Chung, “Non contact eye gaze tracking system by mapping of corneal reflections,” in Proceedings of the International Conference on Automatic Face and Gesture Recognition, pp. 94–99, (Washington,DC), 2002.

[44]. Tobii Technology Inc., Falls Church, VA, USA, Product description: Tobii T/X Series eye trackers, May 2007.

[45]Tobii Technology. http://www.tobii.com. Accessed on 06-05-2010.

[46]Hemin Omer Latif, *Student Member, IEEE,* Nasser Sherkat and Ahmad Lotfi, “Teleoperation through Eye Gaze (TeleGaze): A Multimodal Approach “*Senior Member, IEEE*

[47] C. Hennessey, B. Noureddin, and P. Lawrence,”Fixation Precision in High-Speed Noncontact Eye-Gaze Tracking”, *IEEE Transactions on Systems, Man, and Cybernetics: Part B (Cybernetics)*, Vol. 38, pp. 289-98, 2008

[48] A. Villanueva and R. Cabeza,”A Novel Gaze Estimation System with One Calibration Point”, *IEEE Transactions on Systems, Man, and Cybernetics: Part B (Cybernetics)*, Vol. 38, pp. 1123-1138, 2008

[49] C. H. Morimoto, D. Koons, A. Amit, M. Flickner, and S. Zhai,”Keeping an Eye for HCI”, *XIII Brazilian Symposium on Computer Graphics and Image Processing*, pp. 171-6, 1999

[50] M. A. Bhuiyan, V. Ampornaramveth, S. Muto, and H. Ueno,”On Tracking of Eye for Human-Robot Interface”, *International Journal of Robotics and Automation*, Vol. 19, pp. 42-54, 2004

[51] R.M. Murray, Z. Li, S. Sastry, and S.S. Sastry, A mathematical introduction to robotic manipulation, CRC Press, 1994.

[53] Narayanan, N.H.; Chandrasekaran, B., "Integration of qualitative and quantitative methods invisual reasoning," AI, Simulation and Planning in High Autonomy Systems, 1991. Integrating Qualitative and Quantitative System Knowledge, Proceedings of the Second Annual Conference on , vol., no., pp.272-278, 1-2 Apr 1999.

[54] Johnson B., Christian L., Educational Research Quantitative, Qualitative, and Mixed Approaches,http://www.southalabama.edu/coe/bset/johnson/dr_johnson/2textbook.htm,accessed on 3 March 2010.

[55] K. Dautenhahn, “Methodology themes of human-robot interaction: A growing research field,” *International Journal of Advanced Robotic Systems*, vol. 4, 2007, pp. 103-108.

[56] Dong Hyun Yoo, Jae Heon Kim, Do Hyung Kim, and Myung Jin Chung, “A human-robot interface using vision-based eye gaze estimation system,” *IROS 2002: IEEE/RSJ International Conference on Intelligent Robots and Systems, 30 Sept.-5 Oct. 2002*, Piscataway, NJ, USA: IEEE, 2002, pp. 1196-201.

[57] M. Bhuiyan, V. Ampornaramveth, S. Muto, and H. Ueno, “On tracking of eye for human-robot interface,” *International Journal of Robotics & Automation*, vol. 19, 2004, pp. 42-54.

[58]H. Latif, N. Sherkat, and A. Lotfi, “Teleoperation through eye gaze (TeleGaze): a multimodal approach,” *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO 2009), 19-23 Dec. 2009*, Piscataway, NJ, USA: IEEE, 2009, pp. 711-16.

[59] http://www.best-microcontroller-projects.com/servo-motor.html accessed on 12-05-2010.

[60] http://www.baldor.com/pdf/manuals/1205-394,pdf accessed on 16-04-2010.

[61] Timothy *Hirzel, "Arduino. Internet:* http://www.arduino.cc/en, [July.10, 2010].
[62] Jim Foley. *Interaction Design.* Atlana, GA: John Wiley & Sons Ltd, 2007, pp. 270-284
[63] I. Somerville, *Software Engineering* (9th Edition),
[64] http://www.ibiblio.org/java/books/jdr/chapters/01.html
[65] Heiko Drewes "Eye Gaze Tracking for Human Computer Interaction "A. thesis, University of Calgary, Canada, 2000. Dissertation and der LFE Medien-Informatik der Ludwig-Maximilians-Universität München
[66] L. V. Gordon, "Validities of the forced-choice and questionnaire methods of personality measurement," Journal of Applied Psychology, vol. 35, no. 6, pp. 407-412, Dec. 1951.
[67] C Vent, http://survey.cvent.com/blog/market-research-design-tips-2/0/0/forced-choice-survey-questions [September. 17, 2010].

9 APPENDIX

9.1 Questionnaire

1. How do you find the use of Eyetracking?

Easy □

Tough □

2. Which interface do you find more user-friendly?

Interface 1 □

Interface 2 □

3. Which one is best to control articulated robot arm?

Mouse □

Eyetracking □

4. Is Eyetracking helpful for motor disable people to control the robot arm?

Yes □

No □

5. Do you feel any drawback of using Eyetracking?

Yes □

No □

9.2 RESULTS OF THE FORCE CHOICE CLOSED ENDED QUESTIONS

Question No	**Question**	**Option 1**	**Option 2**
1	Use of Eyetracking	100%	Nil
2	Interface	100	Nil
3	Which is best	60%	40%
4	Eyetracking helpful	100%	Nil
5	Draw back	20%	80%

Table 9 Result of Questionnaire

.