Automation and Expressions Achievement of iCub

Salman Zafar	034-BSCS-11
Shahmeer Arshad	022-BSCS-11
Daniyal Waris	110-BSCS-11

Session: 2011 – 2015

DEPARTMENT OF COMPUTER SCIENCE

GC UNIVERSITY LAHORE

Submitted to GC University Lahore

In partial fulfilment of the requirements

For the degree of

BACHELOR OF SCIENCE
IN

COMPUTER SCIENCE

BY

Name: Salman Zafar
Session: 2011-2015
Roll No: 022-BSCS-2011

Name: Shahmeer Arshad
Session: 2011-2015
Roll No: 034-BSCS-2011

Name: Daniyal Waris
Session: 2011-2015
Roll No: 110-BSCS-2011

Department of Computer Science

GC University Lahore

DECLARATION

We Salman Zafar Roll No. 022-BSCS-2011, Shahmeer Arshad Roll No. 034-BSCS-2011 and Daniyal Waris Roll No. 110-BSCS-2011 Students of Bachelor of Science in the subject of Computer Science Session 2011-2015, hereby declare that the matter printed in the project documentation titled Automation and Expressions achievement of iCub is our own work and has not been printed, published and submitted as research work, thesis or publication in any form in any University, Research Institution etc. in Pakistan or abroad.

Dated

Salman Zafar

Shahmeer Arshad

Daniyal Waris

ROJECT COMPLETION CERTIFICATE

I Syed Ali Raza certify that Project titled Automation and Expressions Achievement of iCub has been carried out and completed by Salman Zafar Roll No. 022-BSCS-11, Shahmeer Arshad Roll No. 034–BSCS–11, and Daniyal Waris Roll no. 110-BSCS-11 under my supervision.

Dated	Signature

Submitted Through:

Dr. Asad Raza Kazmi	**Controller of Examination**

Director

GC University Lahore

Department of Computer Science

GC University Lahore

DEDICATION

This work and Project is dedicated to our Parents, without their support, guidance during studies and help to solve certain problem by motivating, encouraging and knowing the capabilities that their children possess made it possible. Parents’ love and pray is the major key to success in every field of life. Other than that all those colleagues and friend who helped us out in our work and let us release the meaning of true friendship when we needed them in our hard time. We dedicate this of our work to them.

God’s tender guiding hand.

May ALLAH Bless Them

Amen!

ACKNOWLEDGEMENTS

All praises to Almighty Allah Who make us able to complete this task. Our words of special thanks to Dr. Asad Raza, Director, Department of Computer Science for his guidance and support. Next, we want to say special thanks to Mr. Wajahat M. Qazi, Deputy and Mr. Ali Raza, our research supervisor who played a major role in fulfilments of this project, without their valuable direction this might be an impossible for us.

We remember the whole BSCS 2011 batch, in accompany to whom we sat in an academic journey, which is being concluded now. Thank you for being there, our dear fellows.

We are out of justifiable words to stretch our thanks to our commendable parents and our friends who been tolerant during our hectic hours of study and research.

ABSTRACT

The development of robotic cognition and understanding the environment is one of the toughest things now-a-days. A human baby learn many skill by interacting with the environment and other human beings using its limbs and senses. The robot was designed to test this hypothesis by allowing cognitive learning scenarios to be acted out by an accurate reproduction of the perceptual system and articulation of a small child so that it could interact with the world in the same way that such a child does. The project “Automation and Expression Achievement of iCub” is using and iCub simulator on which various image processing techniques have been applied. Through that we will be capable of capturing an image through the lens of the iCub robot, process that image and stimulate different expressions upon detection of certain objects. The project is implemented on an open source platform. This final year project would include the testing and implementing the image processing techniques and getting a reaction from the iCub on the certain inputs.


بسم الله الرحمن الرحيم

Table of Contents

1. Introduction

1.1 Project Motivation

As students of computer sciences, we liked developing and we had great interest in artificial intelligence. We always believed to work on unique ideas and we had experience in developing various projects regarding environment analysis. Our experience on neural networks during our course work motivated us to work further on robotics and get to know about artificial intelligence and its applications in real time. During this project we explored how robots interact and learn from their environment.

Initially, our knowledge regarding the project was very much casual, we were aware of a few programming platforms for robotics. As this is the subject in which we are interested and we are going to make our career as “Robotics Developers”. We have explored how the robots perform when we teach them or they learn things from their environment.

The learning process during the development of this project was much steeper than our previous projects. Our previous knowledge of the Open CV did not cater for the scale of this work, and our skills in Linux were only of a basic level. During this project we learned everything that was necessary about these languages and how we can apply them in automation of robots.

1.2. Aims and Objectives:

The core objective of the project that are designated are as follows:

- Provide ability to robot in simulation to identify an object
- Provide ability to robot to recognize it in real time
- Enable robot to grab the desired object
- Achieve expressions in robot
- Get real time feedback from robot

1.3. Why the work is being done

The reason for our work is to improve the robotic cognition by helping them understand their surroundings and world. Previous work has been done on humanoids making them familiar with real world objects. But the difference that we have tried to bring through our work is to make our robot "iCub" familiar with objects in its simulated world making it capable of detecting tracking and probably following the objects in its simulated world using the iCub simulator.

1.4. Related Work:

A lot of research work has been done in this field but that work is done mostly at Masters and PhD level. Some of the recent research work is discussed below.

a. Robot CUB

The goal of Robot Cub was to study cognition through the implementation of a humanoid robot the size of a 3.5 year old child: the iCub. This is a fully open source & hardware project. This is the project that started the iCub both in hardware and software.

b. Codyco

CoDyCo aims at advancing the current control and cognitive understanding about robust, goal-directed whole-body motion interaction with multiple contacts. CoDyCo will go beyond traditional approaches:

(1) Proposing methodologies for performing coordinated interaction tasks with complex systems

(2) Combining planning and compliance to deal with predictable and unpredictable events and contacts

(3) Validating theoretical advances in real-world interaction scenarios

c. Xperience:

Xperience will demonstrate that state-of-the-art enactive systems can be significantly extended by using structural bootstrapping to generate new knowledge. This process is founded on explorative knowledge acquisition, and subsequently validated through experience-based generalization.

d. EFAA

The Experimental Functional Android Assistant (EFAA) project will contribute to the development of socially intelligent humanoids by advancing the state of the art in both single human-like social capabilities and in their integration in a consistent architecture.

1.5. Intended audience:

Intended audience for our project are those students who are willing to get their hands into robotics, this platform will provide them a chance to make a mark and with their dedication and proper guidance they can go on to continue researching on iCub

1.6. Methodology

Our method is divided into following main steps:

First step is setting up the environment for our iCub where the objects to be recognized are placed, it is possible to place the objects on different locations but that would be out of the scope of the iCub so we have used a table that is placed in on which all the objects are placed.

Next step is for the iCub to detect the position desired object. We have used image processing techniques that make iCub capable of distinguishing the required object.

Third step is of iCub simulation is to pick up the detected object and for picking it up we have used iCub motors and for controlling those motors we needed complete knowledge of mechanical functionalities and its relation with the simulator.

Last step is triggering the emotions interface to simulate expressions. After the successful detection of an object and grabbing, emotions of happiness will be triggered and expressions of happiness with be delivered by iCub. If there is no object detected in the camera range, emotions of failure will be triggered and robot will deliver expressions of sadness.

2. Background

In order to understand the requirements of expressions achievement in iCub, it is important to have details on the following:

- iCub
- Yarp
- Open Source Robotics
- Cognitive Robotics
- Humanoids and Android Robots
- An understanding of robotic simulations
- Interconnection between sensors, processors and actuators of robot with simulator

Looking at certain objects, one can usually guess the position and size of that object. Object detection and identification is related to the perception. One person looking at an object *A* may think it to be closer than the object *B* because of the frame of reference. Another person may disagree on the colour of the Object *A* and Object *B* due to colour blindness. This is the first fact to keep in mind that every individual have his or her own unique perception about objects and their position. Some of other factors include: write about the motion of objects and objects changing their place with respect to some frame of reference

i. ICub

ICub is a 1 metre high humanoid robot testbed for research into human cognition and artificial intelligence.

The ICUB is open-source, with the hardware design and software.

The motivation behind the strongly humanoid design is the embodied cognition hypothesis, that human-like manipulation plays a vital role in the development of human cognition. A baby learns many cognitive skills by interacting with its environment and other humans using its limbs and senses, and consequently its internal model of the world is largely determined by the form of the human body. The robot was designed to test this hypothesis by allowing cognitive learning scenarios to be acted out by an accurate reproduction of the perceptual system and articulation of a small child so that it could interact with the world in the same way that such a child does.

The dimensions of the iCub are similar to that of a 2.5 year old child. The robot is controlled by a controller which communicates with actuators and sensors using canbus.

It utilises tendon driven joints for the hand and shoulder. The Joint angles are measured using custom-designed sensors and the robot can be equipped with torque sensors. The finger tips can be equipped with tactile touch sensors.

The software library is largely written in C++ and uses YARP for external communication.

The robot was not designed for autonomous operation.

The head has stereo cameras, where eyes would be located on a human and microphones on the side. It also has lines of red LEDs representing mouth and eyebrows mounted behind the face panel for making facial expressions.

ii. Yarp

iii. Open Source Robotics

Robot Cub is Open Source both for software and hardware. While the phrase "Open Source software – OSS" is clear, "Open Source hardware" might sound strange, but in fact it is a plain transfer of the open source philosophy to the entire design of the Robot Cub platform. The design of the robot started from the preparation of specifications (e.g. estimation of torque, speed, etc.), a typical 3DCAD modelling, and eventually in the preparation of the executive files which are used to fabricate parts and for assembly. Without good documentation it is very complicated to build and assemble a full robot. This means that documentation (as for software) is particularly important.

The CAD files, in some sense, can be seen as the source code, since they are the "preferred form of the work for making modifications to it", in the language of the GPL. They get

"Compiled" into 2D drawings which represent the executive drawings that can be used by any professional and reasonably well-equipped machine shop either to program CNC machines or to manually prepare the mechanical parts. This compilation

Process is not fully automated and requires substantial human intervention. There is a clear dependency of the 2D drawings on the original 3D CAD model. To enable the same type of virtuous development cycle as occurs in open source software, the 3D CAD is required, since changes happen in 3D first and get propagated to 2D later. In addition, assembly diagrams, part lists, and all the material produced during the design stage should be included to guarantee that the same information is available to new

developers. The actual design of the robot had to incorporate manipulation by providing sophisticated hands, a flexible oculomotor system, and a reasonable bimanual workspace. On top of this, the robot has to
Support global body movements such as crawling, sitting, etc.

These many constraints were considered in preparing the specifications of the robot and later on during the whole design process. Both the iCub design and its software architecture are distributed as Open Source. This is not enough to guarantee success. Additional initiatives are required. Robot Cub is giving away six copies of the iCub to the winners of an Open Call for proposals to use the iCub (recently concluded). In addition a structure called The Research and Training Site (RTS) has being created to support visiting researchers to work on the iCub prototypes in Genoa.

iv. Cognitive Robotics Group

The Cognitive Robotics group is concerned with endowing robotic or software agents with higher level cognitive functions that involve reasoning, for example, about goals, perception, actions, the mental states of other agents, collaborative task execution, etc. To do this, it is necessary to describe, in a language suitable for automated reasoning, enough of the properties of the robot, its abilities, and its environment, to permit it to make high-level decisions about how to act. The group has developed effective methods for representing and reasoning about the prerequisites and effects of actions, perception and other knowledge-producing actions, and natural events and actions by other agents.

This presents an architecture for autonomous robots to generate behaviour in joint action tasks. To efficiently interact with another agent in solving a mutual task, a robot should be endowed with cognitive skills such as memory, decision making, action understanding and prediction. The proposed architecture is strongly inspired by our current understanding of the processing principles and the neuronal circuitry underlying these functionalities in the primate brain. As a mathematical framework, we use a coupled system of dynamic neural fields, each representing the basic functionality of neuronal populations in different brain areas. It implements goal-directed behaviour in joint action as a continuous process that builds on the interpretation of observed movements in terms of the partner's action goal. We validate the architecture in two experimental paradigms:
(1) A joint search task
(2) A reproduction of an observed or inferred end state of a grasping–placing sequence. We also review some of the mathematical results about dynamic neural fields that are important for the implementation work.

Cognitive robotics views animal cognition as a starting point for the development of robotic information processing, as opposed to more traditional Artificial Intelligence techniques. Target robotic cognitive capabilities include perception processing, attention allocation, anticipation, planning, and complex motor coordination, reasoning about other agents and perhaps even about their own mental states. Robotic cognition embodies the behaviour of intelligent agents in the physical world (or a virtual world, in the case of simulated cognitive robotics). Ultimately the robot must be able to act in the real world.

v. Humanoid Robots:

In human-to-human collaboration, cooperate gesture play a significant role in communication, to educate, to lead and to build a connection. In the early stages of human, child communicates cooperatively, to inform other individuals about things and share their interest through gestures as interaction. Thus, it seems reason to expect that, such gestural interaction to people collaborating with robots might be helpful.

A Humanoid Robot is a structure form that related to the Human body Structure and it is used to recognize the human interaction gesture by using different tools of the human body such hands, fingers, face and legs. The humanoid robots are built and used for various scientific experimentation. The research need to understand the human body structure and behaviour to build and structure humanoid robots. The basic aim to build humanoid robots is to build orthosis and prosthesis for human being.

Orthosis: an externally applied device used to modify the structure, design and function characteristics of the neuromuscular.

Prosthesis: is an artificial part of the body that been missed or lost by any disease, accident.

Today a lot of work is being done on building robotics that are capable of recognition of the human interacting gesture. When the human interact with each other, the adopt the feeling, motion and intention of other to achieve the goal. However it’s unclear how this interaction gesture interact with humanoid.
Our work will provide an approach or director for the development of interactive gesture that interact with humanoid robots. We will provide an interactive mode that would provide humanoid robot to interact with the

human, the human will interact and treat the humanoid robot as child. Our evaluation approach to measure the object interaction with humanoid robot.

vi. Sensors:

Close your eyes and touch your nose. If everything is working properly, this should be easy because your brain can sense your body, as well as its position and movement through space. This is called proprioception. But how does this "sixth sense" work — and what happens when it clashes with other senses?

As human body had five standard senses, that include the vision, hearing, smell, taste and touch. These Five senses allow human body to experience the world. It been saying that "Without it (Five senses), our brain are lost"

Sensors is a transducer that would, detect the thing from the outer environment and in return it would generate a corresponding output that would be generally the electrical or optical signal. The Sensor being the part of three primitives of robotics, so sensor played a vital and important role in Robotics Paradigm. Sensor are classified according to the physical process with which they perform their work .The Sensor are further divided to the Following:

1. Proprioceptive Sensors
2. Exteroceptive Sensors

i. Proprioceptive Sensors:

Is the type of sensor, relative to the position of the body part and strength need to show movement? In human body these proprioceptors are present in muscle, tendons and joints. As brain will gather the information from the proprioceptors and will be sense to the whole body position, movement and acceleration. The kinaesthesia explain the movement senses, this kinaesthesia been used in two ways either inconstantly that refer either alone proprioception or refer to the brain that integrate the proprioceptive and visible inputs. The Proprioception is recognized as something that contribute the overall body ownership.

In the Case of humanoid robots two major kind of sensors are used,

- accelerometers
- Tilt. Sensor

i. **Accelerometers:**

The most common inertial sensors is the accelerometer, it’s a dynamic sensor capable of a vast range of sensing. It is available in which accelerometer can detect the orthogonal axes. Theoretically, an accelerator operate as a damped mass on a spring. When an accelerometer know-how an acceleration, the mass is moved to the point that the spring is able to accelerate the mass same rate as the spring is able to accelerate the mass at the same rate.

What are the purpose of using the Accelerometer? .The answer would be the simple and easy that by calculating the number of static acceleration that occur because of the gravity, we can find out the devices that is tilted at with regarding to the earth. By feeling the number of dynamic acceleration, we can evaluate the device that is moving.

- **Tilt Sensors:**

It’s to measure the affection, the force sensor is placed in robotics hand and feet to measure the contact force with environment, position sensors that mark the actual position of the robot or even the speed of the sensors.

ii. **Exteroceptive sensors:**

The Exteroceptive sensors determine the measurement of the object relative to the robot’s frame. Thus these sensors are being place in a category of proximity sensors. These sensors kept the robot from collapsing with other object objects. They can also worked to determine the distance from the robot to the object.

These Exteroceptive sensor further divide into three category.

- **Contact Sensors:** has typically a simple mechanism in which it sends a signal when physical contact is made. Contact sensor detect the interaction or contact when two parts interact or mate with each other and to measure the interaction forces and torque when the robot perform the interaction or mating functionality.in contact sensor there is another part or type of it, in which it measure the multitude of parameters of the touched object surface which is known as the **Tactile Sensor.**

- **Range Sensor:** functionality simple that measure the distance of the object in operational performed area. IT can be also called as distance detection Device that provide a simple binary signal when a particular threshold detect. This sensor has the ability or it may use to for navigation and obstacle navigation. This Sensor depend on the Major principle: Time of flight and triangulation.

- **Vision Sensor:** complex sensing Process, which involves the extracting, characterizing and interpreter information about the image in order to identify images or describe objects environment.

vii. Actuator

An **actuator** is a type of motor that is responsible for moving or controlling a mechanism or system. It is operated and handled by a source of energy, typically electric current, hydraulic fluid pressure and converts that energy into motion. An actuator is the mechanism by which a control system acts upon an environment. Humanoid robots are constructed in such a way that they mimic the human body, so they use actuators that perform like muscles and joints, though with a different structure. To achieve the same effect as human motion, humanoid robots use mainly rotary actuators. They can be either electric, pneumatic, hydraulic, piezoelectric or ultrasonic.

Hydraulic

A hydraulic actuator consists of a cylinder or fluid motor that uses hydraulic power to facilitate mechanical operation. The hydraulic cylinder consists of a hollow cylindrical tube along which a piston can slide. The term single acting is used when the fluid pressure is applied to just one side of the piston. The piston can move in only one direction, a spring being frequently used to give the piston a return stroke. The term double acting is used when pressure is applied on each side of the piston; any difference in pressure between the two sides of the piston moves the piston to one side or the other.

Pneumatic

A pneumatic actuator converts energy formed by vacuum or compressed air at high pressure into either linear or rotary motion. Pneumatic energy is desirable for main engine controls because it can quickly respond in starting and stopping as the power source does not need to be stored in reserve for operation.

Electric

An electric actuator is powered by a motor that converts electrical energy into mechanical torque. The electrical energy is used to actuate equipment such as multi-turn valves. It is one of the cleanest and most readily available forms of actuator because it does not involve oil.

Mechanical

A mechanical actuator functions by converting rotary motion into linear motion to execute movement. It involves gears, rails, pulleys, chains and other devices to operate. An example is a rack and pinion.

viii. Robotics Simulation

A robotics simulator is used to create embedded applications for a robot without depending physically on the actual machine, thus saving cost and time. In some case, these applications can be transferred on the real robot (or rebuilt) without modifications. The term robotics simulator can refer to several different robotics simulation applications. For example, in mobile robotics applications, behaviour-based robotics simulators allow users to create simple worlds of rigid objects and light sources and to program robots to interact with these worlds. Behaviour-based simulation allows for actions that are more biological in nature when compared to simulators that are more binary, or computational. In addition, behaviour-based simulators may "learn" from mistakes and are capable of demonstrating the anthropomorphic quality of tenacity.

3. DESIGN

In this chapter we have discussed the overall design and architecture of our project. Design is very important for any software as it helps in understanding any software or application. Design is something which is the initial step in making any software or product. Implementation cannot be started before making a design.

The objective of a designer is to make a model or a representation using which a product can be built later. The process through which a design is made requires experience of building the similar product, a set of rules which provides guidance in evolving a model in a way we want, a set of criteria to judge the quality and an iterative process which finally leads us to the final design representation.

We will start from the development plan that contains different phases and perform each function which exist in this plan

i. DEVELOPMENT PLAN

Phase 1 Activity: PLANNING

Tasks:

1. Checking economic and social feasibility of development.
2. Selecting the process model that is to be followed.
3. Setting the duration of every phase and activity.

Phase 2 Activity: MODELING PHASE

Tasks:

1. Prototyping
2. Development of scenarios based elements
1. Developing Use-Cases

Phase 3 Activity: DESIGN

Task

1. Developing Architectural Design Elements
2. Developing component Design Elements
3. Developing Graphical User Interface

Phase 4 Activity: CONSTRUCTION

Task

1. Code Generation

Phase 5 Activity: TESTING

Task

1. Unit Testing
2. Integration Testing
3. System Testing

ii. Requirement Analysis

Requirement analysis is a process of gathering requirements from various stakeholders or users and then incorporating those requirements into the system. Requirements analysis is critical to the success of a system or software project. The requirements should be documented and related to the identified business needs or opportunities, and defined to a level of detail sufficient for system design. The challenge for the software engineers is to clearly analyse and define the system requirements for a new software that is to be designed so that final product is according to the needs defined by the stakeholders or users.

TYPES OF REQUIREMENTS

1) Functional Requirements
2) Non- Functional Requirements

FUNCTIONAL REQUIREMENTS

Functional requirements define the behaviour of what the system must do that is a set of inputs, outputs, calculations etc. it defines the components of a software system and the tasks it must accomplish.

Following are the functional requirements:

1. Proper Movement of iCub's Limbs.
2. ICub Understanding and observing its surrounding world
3. Object Detection.
4. Object Tracking.

5. Achievement Expressions:

    1. Moving arms in happiness.
    2. Changing its facial expressions.
    3. Picking up the detected object showing it to the audience.

NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are the qualities of the system that specify what the system shall be. These requirements specify the general criteria that judge the overall working of the system rather that specific behaviour's.

Following are the non-functional requirements of our system:

1. Efficient.
2. High Performance.
3. Extensibility.
4. Productivity.

iii. TESTING

The Project is based on the Automation of ICub, in which ICub will detect the object and show some emotion on to his face so this all work is been performed on ICub simulator.

Test Cases:

Thus Linux is best platform for executing and using the ICub Simulator as it would provide better and helpful platform for processing. Linux provide it a better platform to execute through its Terminal. The ICub can only been Access through the Terminal, Terminal is a black window that can execute command, it work the same way as the command line interface or MS_DOS works in the windows. IT execute commands that would perform Functionality direct.

The test Cases that are tested would be like writing a command on the terminal, and functionality would be performed against those command.

The test Case that are tested and Performed are given below:

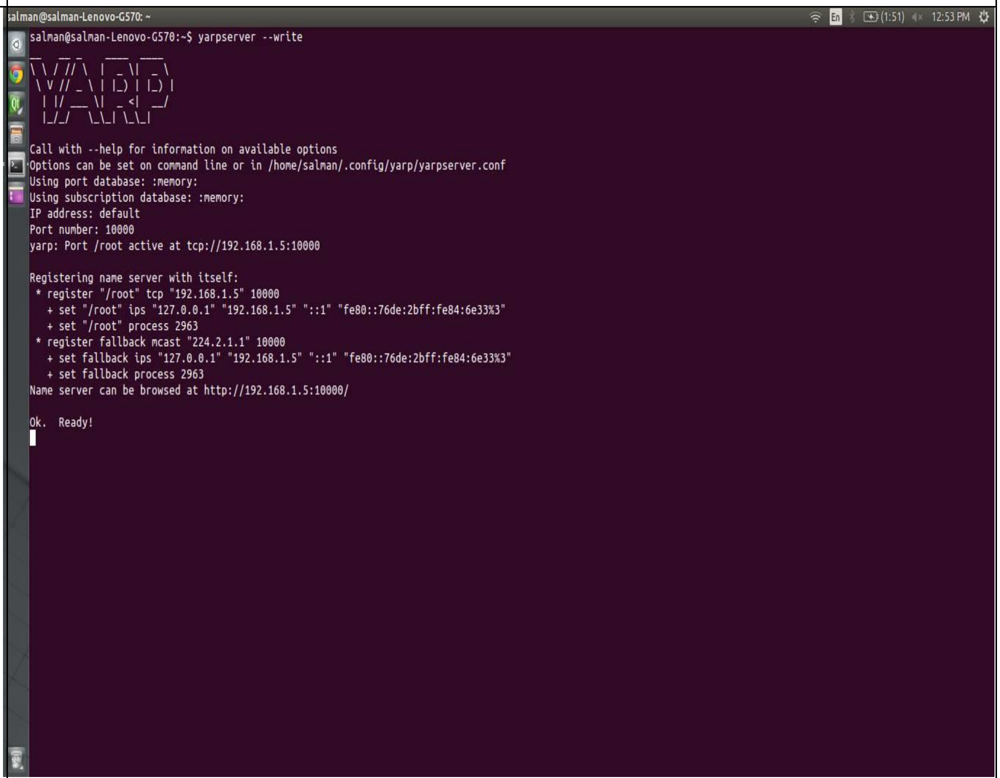
1. Yarp Server connections
2. Robot Simulator connections
3. Robot Motor Controls Interface
4. Head Motors Connections
5. Hand Motors Connections
6. Arms Motors Connections
7. Legs Motors Connections

Test Case Description:

Each and Every component of iCub is performed on to the terminal. Unit testing is performed on to the ICub that show how the ICub would performed on to the terminal when various Command having functionality is being performed, Does it Work the Same way as expected result. If the Result is obtained as it is expected then the Test Case Would be Passed Otherwise it would be Failed. The Reason of Performing test case would be that user would not to face any error or problem when they uses this application.

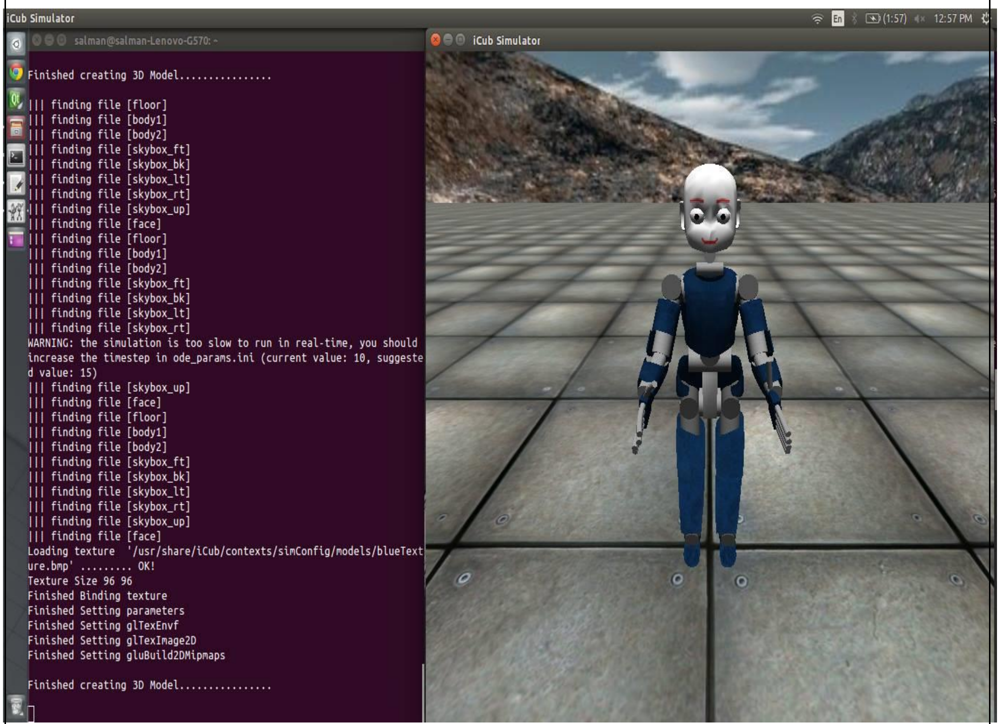
The Test Cases that have been successfully performed or carried out are as Follows:

i. Yarp Server

<table>
<tr><td>Test Case ID</td><td>TC-001</td><td>System Tester</td><td>Salman Zafar</td></tr>
<tr><td>Version</td><td>Ubuntu 14.10</td><td>Date</td><td>26-7-2015</td></tr>
<tr><td>Purpose</td><td colspan="3">To activate The Yarp Server</td></tr>
<tr><td>Pre Requisite</td><td colspan="3">Terminal</td></tr>
<tr><td>Execution Description</td><td colspan="3">2. Make The System Ready For Execution
3. GO to Search Option On The Top OF desktop
4. Write Terminal , Terminal Would Opened</td></tr>
<tr><td>Input</td><td colspan="3">Write “yarpserver --write” onto the Terminal</td></tr>
<tr><td>Result</td><td colspan="3">The result Would Be “Okay. Ready”</td></tr>
<tr><td>Result Image</td><td colspan="3">


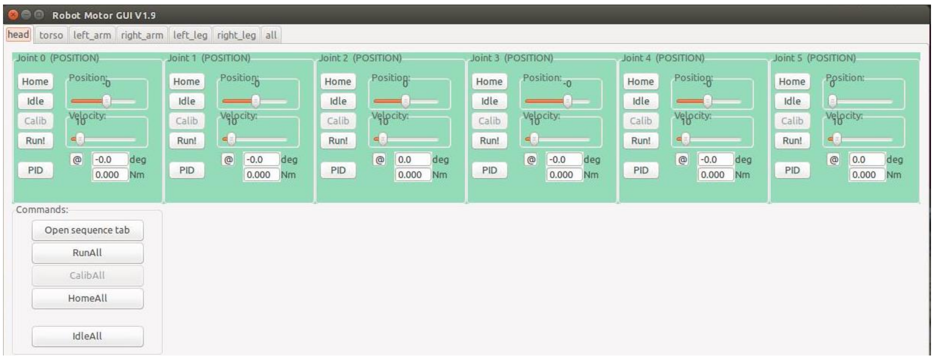
</td></tr>
</table>

ii. <u>ICub</u>

Test Case ID	TC-003	**System Tester**	Shahmeer Arshad
Version	Ubuntu 14.10	**Date**	26-7-2015
Purpose	To activate Robot Motor Controlling Interface		
Pre Requisite	Terminal		
Execution Description	1. Make The System Ready For Execution 2. GO to Search Option On The Top OF desktop 3. Write Terminal , Terminal Would Opened		
Input	Enter “**iCub_SIM”** in the Terminal		
Result	The result Would Be **“ICUB Simulator ”**		
Result Image			

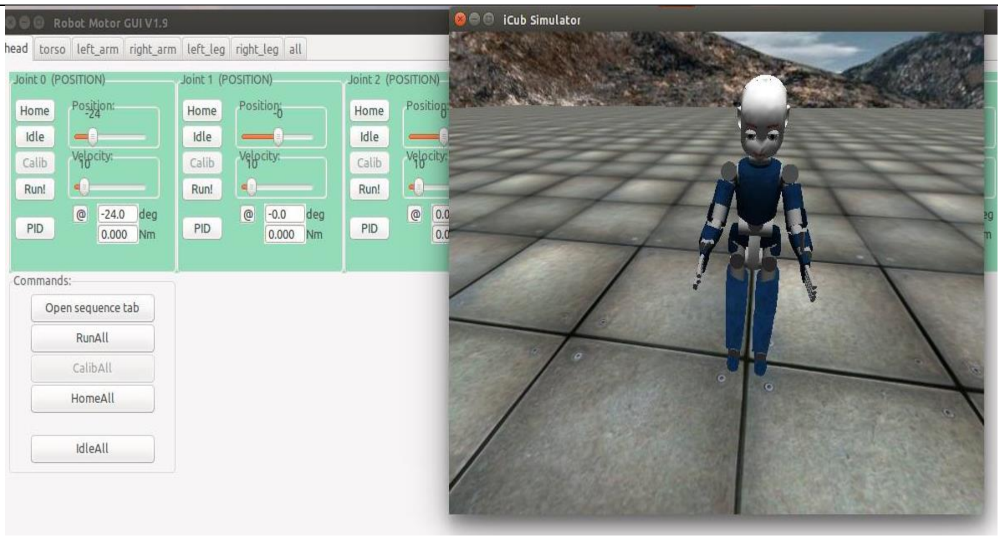
iii. Robot Motor Interface

Test Case ID	TC-003	**System Tester**	Daniyal Waris
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To activate The Robot Motor Interface		
Pre Requisite	Terminal		
Execution Description	1. After The Completion of First Two Step. 2. After The ICub is activate 3. Open Another Terminal		
Input	Write “**robotMotorGui”** onto the Terminal		
Result	The result Would Be “**it Will Open Robot Motor Control System”,** Write IcubSim in the blank Space**, To** Open Robot Motor Interface.		

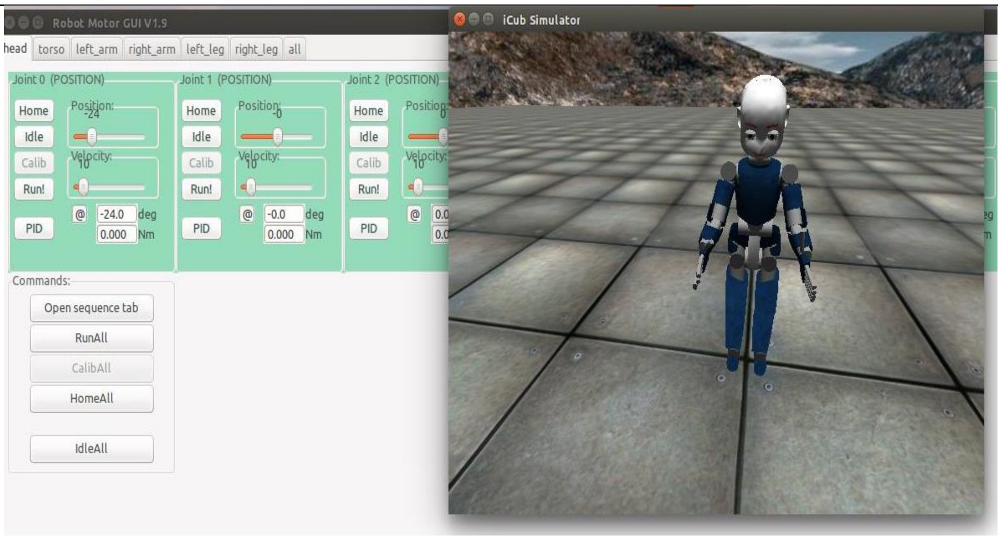


iv. Head Motor Test Case

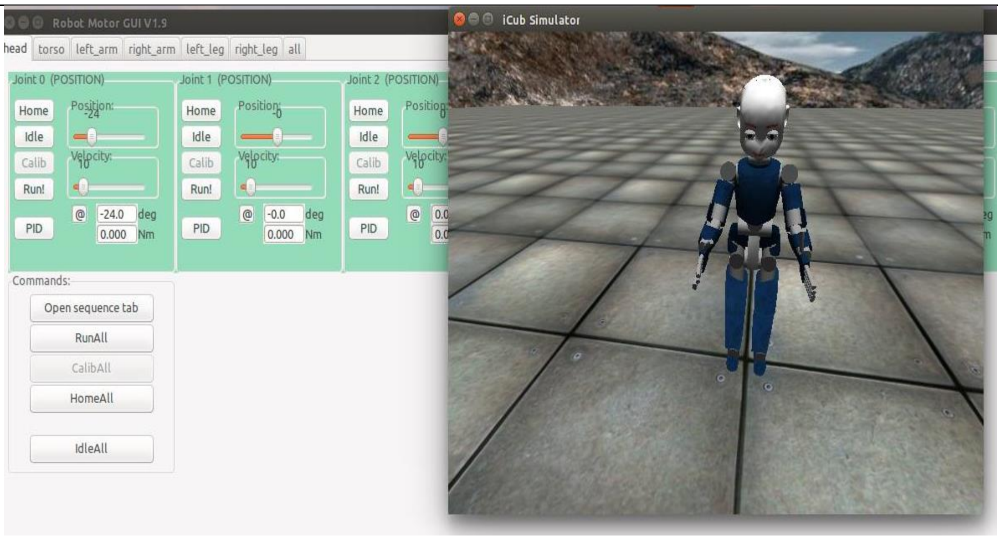
1. Joint O (Position) Downward

Test Case ID	TC-004	**System Tester**	Salman Zafar
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The Head Motor Working at Joint 0		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	1. To check The Head Motor Work At Joint 0 (Position) 2. On Robot Motor GUI Head Tab will be at the top. 3. Use Position to Move the Position of head .Initial Position is at 0.		
Input	Control The Position Using the scroller to change the position of head		
Result	The result Would Be **"it will move the Head in Downward direction"**		
Result Image			

2. Joint 1 (Position) LEFT Side

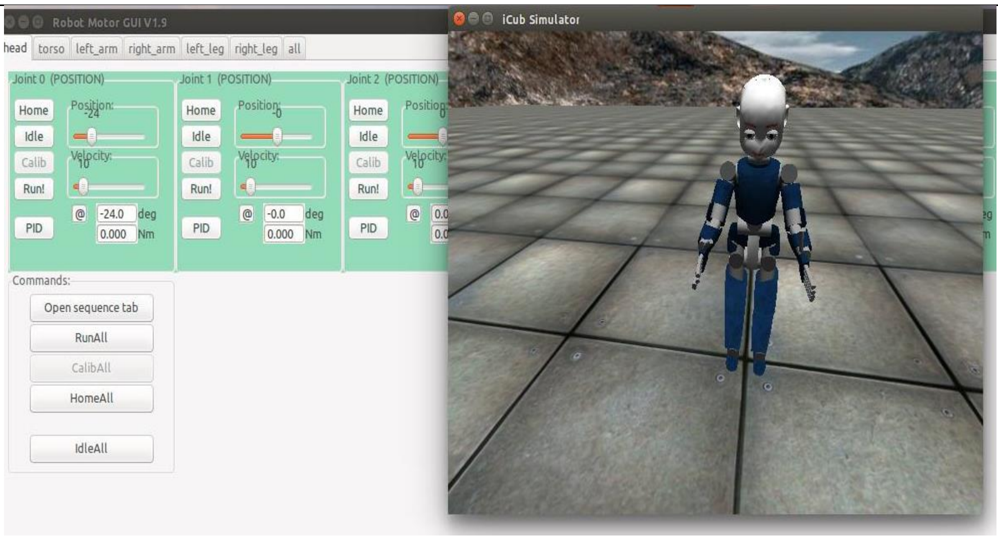
Test Case ID	TC-005	**System Tester**	Salman Zafar
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The Head Motor Working at Joint 1		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	1. To check The Head Motor Work At Joint 1 (Position) 2. On Robot Motor GUI Head Tab will be at the top. 3. Use Position to Move the Position of head .Initial Position is at 0.		
Input	Control of the Position Using the scroller to change the position of head		
Result	The result Would Be **"It would move the Head to the left side of User View and Right side With respect to Robot Head Position"**.		
Result Image			

3. Joint 2 (Position) Right Side

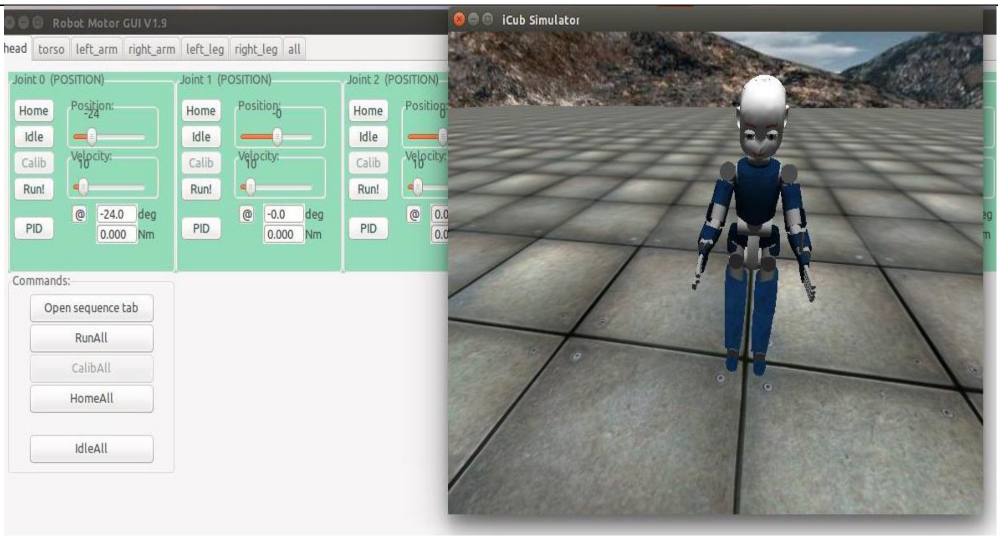
Test Case ID	TC-006	**System Tester**	Shahmeer Arshad
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The Head Motor Working at Joint 2		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	1. To check The Head Motor Work At Joint 2 (Position) 2. On Robot Motor GUI Head Tab will be at the top. 3. Use Position to Move the Position of head .Initial Position is at 0.		
Input	Control of the Position Using the scroller to change the position of head		
Result	The result Would Be **“It would move the Head to the Right side of User View and Left side With respect to Robot Head Position”**.		
Result Image			

v. <u>Torso Motor Test Case</u>

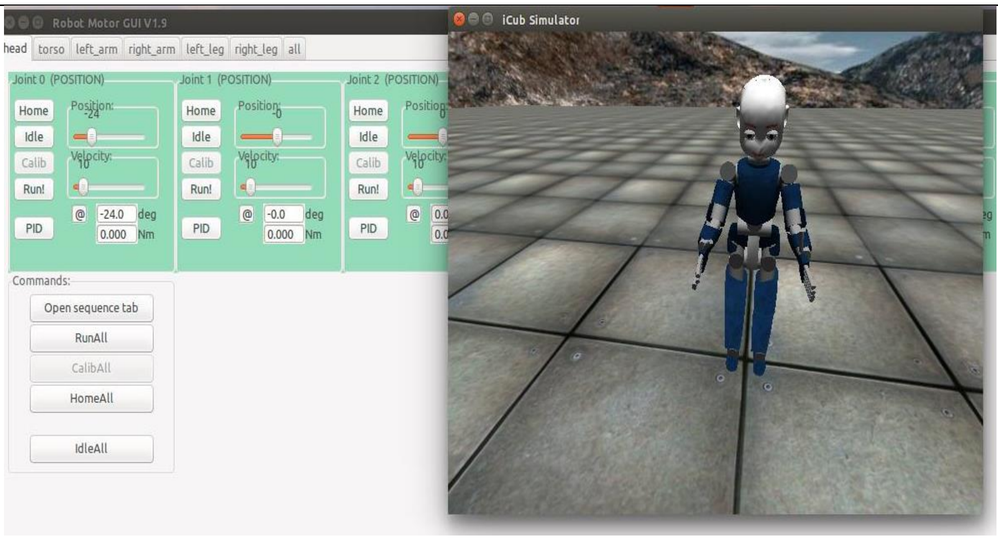
1 .Joint O (Position) Movement of body will move toward right and Left side

Test Case ID	TC-007	**System Tester**	Shahmeer Arshad
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The Body Movement,		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	The body upper Part will Move by scrolling the bar toward Positive side to left side and on negative side it will move to right side		
Input	Control The Position Using the scroller to change the position of upper Part of body		
Result	The result Would Be **"The Upper part of the Body will Move in the Respective direction"**		
Result Image			

2 .Joint 1 (Position) Movement of Upper body will bend in Right And Left Side

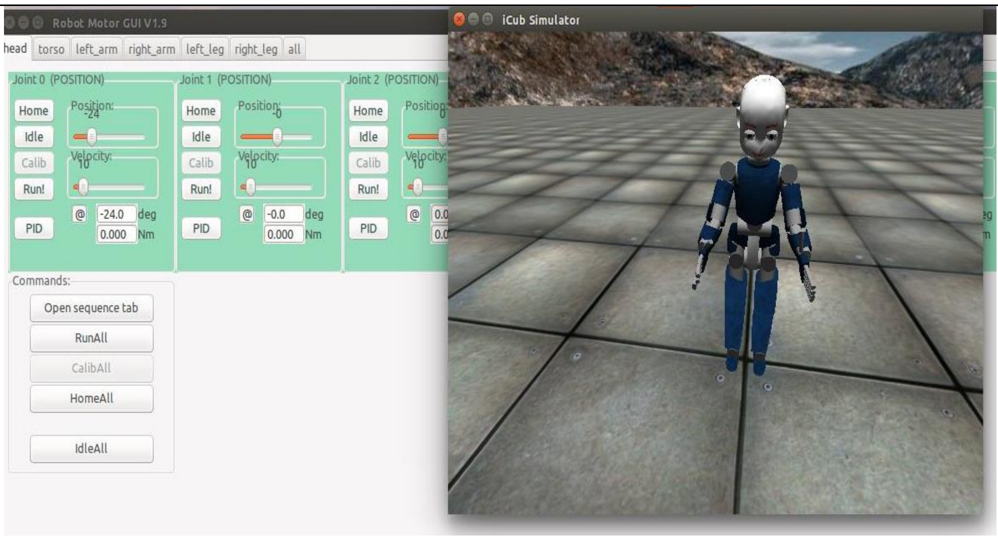
Test Case ID	TC-008	**System Tester**	Shahmeer Arshad
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The Body Movement,		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case the upper body will bend Towards right or left side.		
Input	Using the scroller to change the position of upper Part of body		
Result	The result Would Be **"The Upper part of the Body will bend Either in the Right or left side "**		
Result Image			

3 .Joint 2 (Position) Movement OF Upper Body Forward and Backward

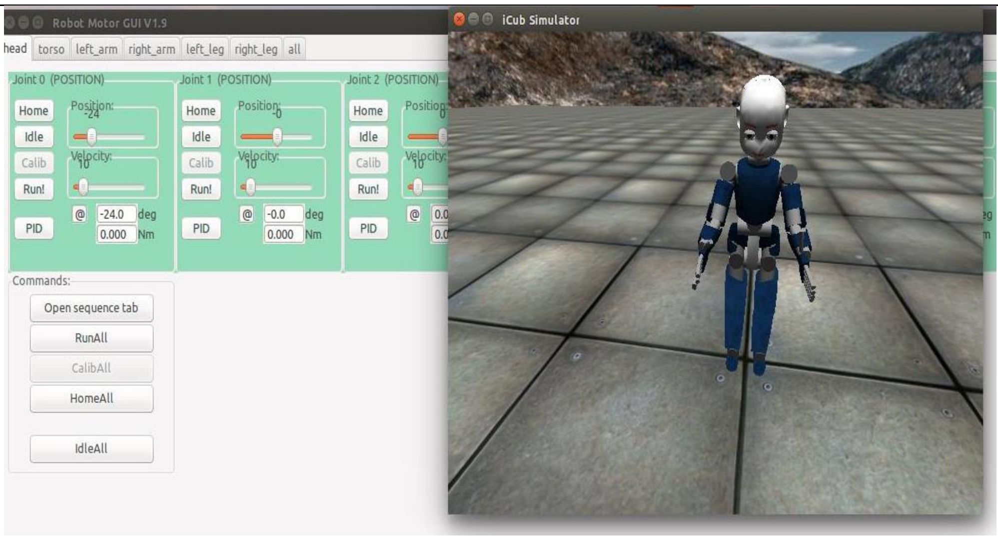
Test Case ID	TC-009	**System Tester**	Salman Zafar
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The Body Movement,		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case the upper body will bend Towards front and backward.		
Input	Using the scroller to change the position of upper Part of body		
Result	The result Would Be **“The Upper part of the Body will bend Either in the Front or backward ”**		
Result Image			

vi. Right Arm Test Case

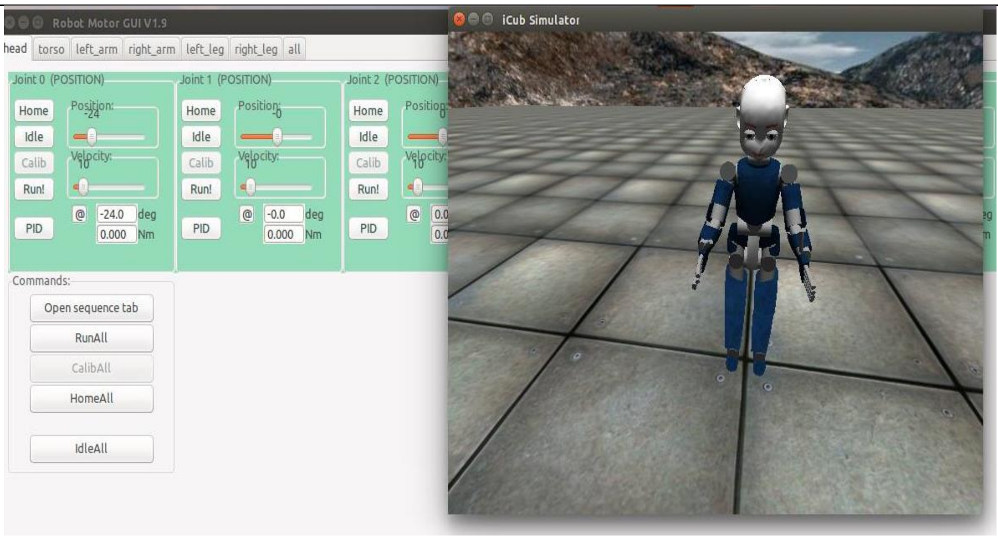
i. Joint 0 (Position) Shoulder Movement

Test Case ID	TC-010	**System Tester**	Salman Zafar
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The movement of Right Shoulder.		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case Right Shoulder would move Up and Down from its rest position.		
Input	Using the scroller to change the position of Right Shoulder.		
Result	The result Would Be **"The Right shoulder would Move Upward when its value is Negative and when its value is Positive then it will move to Downward Direction."**		
Result Image			

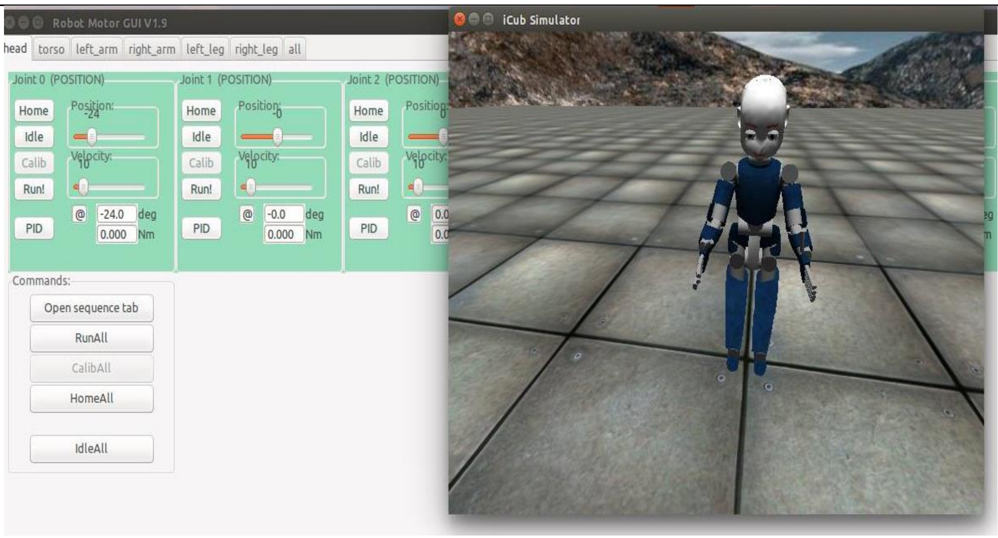
ii. Joint 1 (Position) Shoulder Movement Right

Test Case ID	TC-011	**System Tester**	Salman Zafar
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The movement of Right Shoulder.		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case Right Shoulder would move Right and left from its rest position.		
Input	Using the scroller to change the position of Right Shoulder.		
Result	The result Would Be **"The Right shoulder would Move Right side when its value is increased."**		
Result Image			

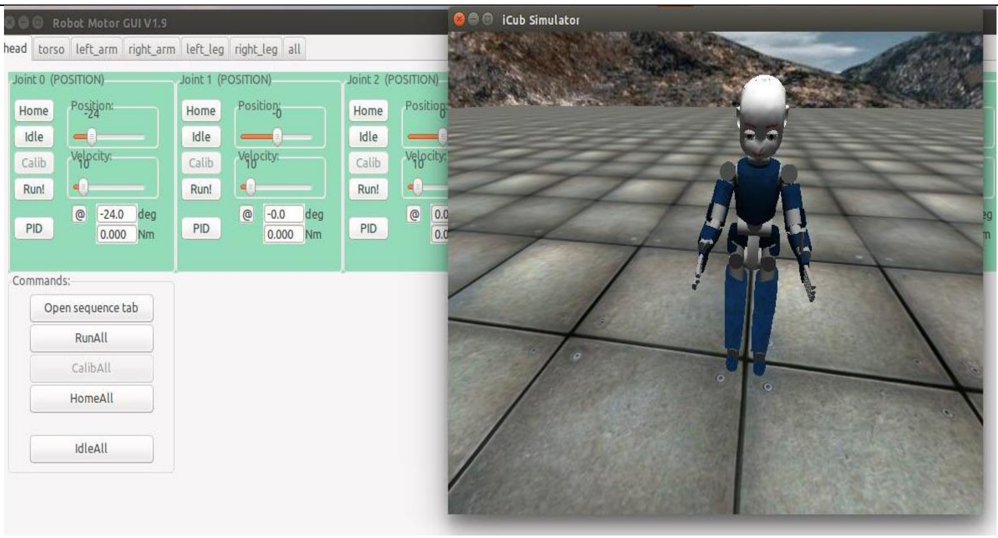
iii. Joint 2 (Position) elbow Movement

Test Case ID	TC-012	**System Tester**	Salman Zafar
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The movement of Right Elbow.		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case Right Arm Elbow would either move to the left or right side.		
Input	Using the scroller to change the position of Right Elbow.		
Result	The result Would Be **“The Right Arm elbow would Move either Right side or Left side as its value changes.”**		
Result Image			

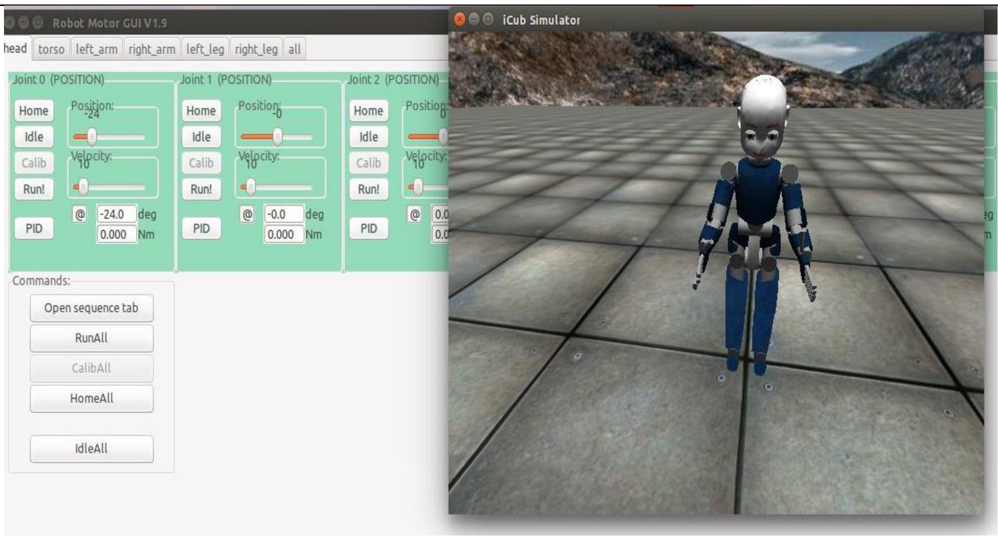
iv. Joint 3 (Position) elbow UP or down Movement

Test Case ID	TC-013	**System Tester**	Daniyal Waris
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The movement of Right Arm Elbow.		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case Right Arm Elbow would either move Up or Down side from its rest position.		
Input	Using the scroller to change the position of Right Elbow.		
Result	The result Would Be **"the right Arm Elbow would move up or downward as its value changes From, Robot Motor Interface."**		
Result Image			

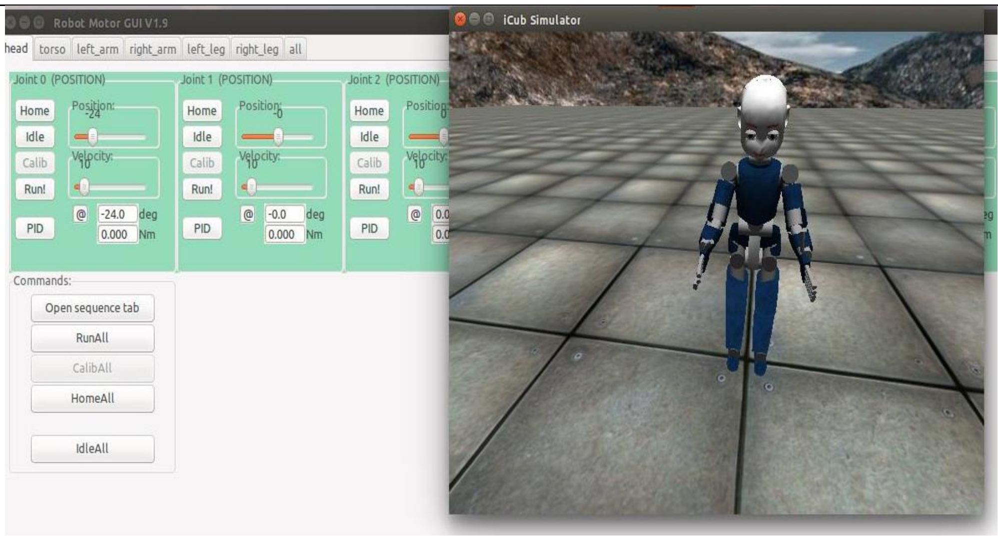
v. Joint 4 (Position) hand Rotate

Test Case ID	TC-014	**System Tester**	Salman Zafar
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The movement of Right Arm hand.		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this Case The Right Arm Hand would rotate upside and downside.		
Input	Using the scroller to change the position of Right Arm hand.		
Result	The result Would Be " **the arm will rotate either in upward or downward direction** "		
Result Image			

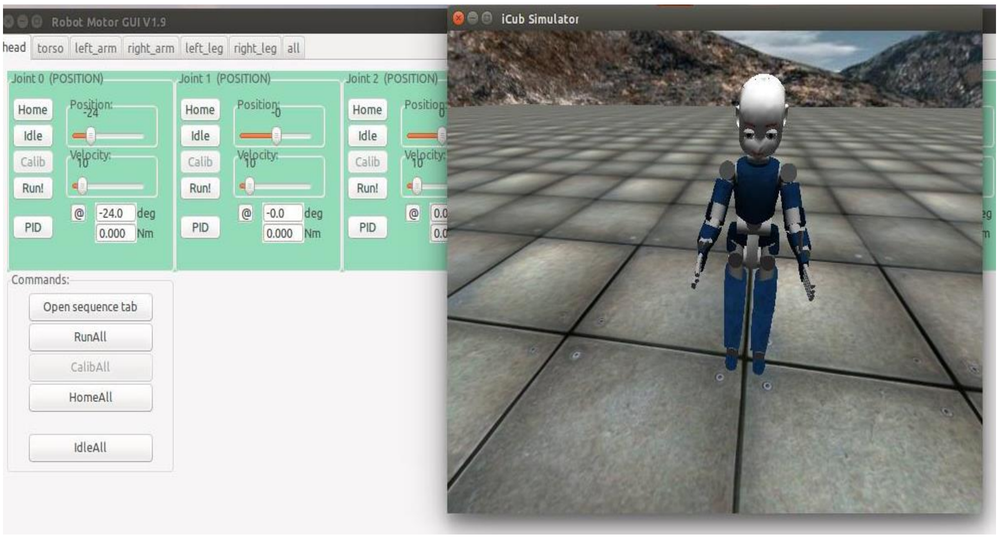
vi. Joint 5 (Position) hand down movement

Test Case ID	TC-015	**System Tester**	Daniyal Waris
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The movement of Right Arm hand Down movement.		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case Right Arm hand would move Down side from its rest position.		
Input	Using the scroller to change the position of Right Elbow.		
Result	The result Would Be **"The Right Arm hand will move in downward Direction"**		
Result Image			

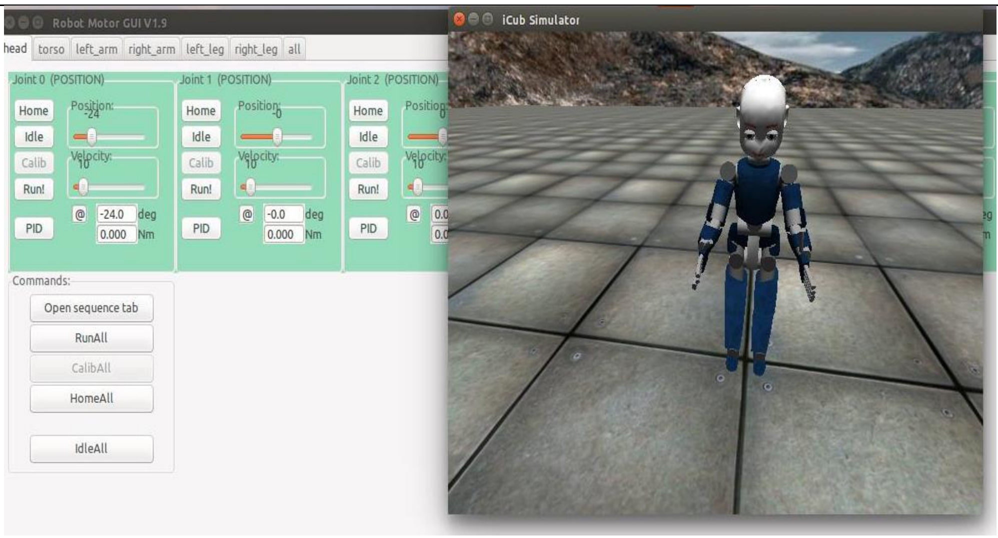
vii. Joint 6 (Position) hand down movement

Test Case ID	TC-016	**System Tester**	Daniyal Waris
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The movement of Right Arm hand Down movement.		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case Right Arm hand would move Down side from its rest position.		
Input	Using the scroller to change the position of Right Elbow.		
Result	The result Would Be **"The Right Arm hand will move in downward Direction"**		
Result Image			

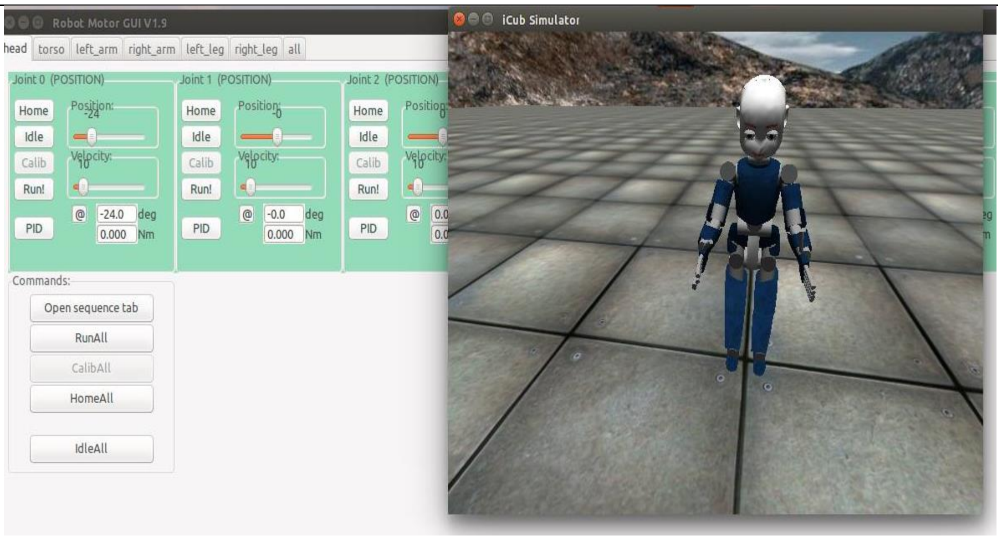
viii. Joint 8 (Position) Thumb Movement (Upward /Downward)

Test Case ID	TC-018	**System Tester**	Daniyal Waris
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The movement of Thumb OF Right Hand Does it move Upward Or Downward		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case Right Hand Thumb would Move in the upward or Downward Direction.		
Input	Using the scroller to change the position of Right Hand Thumb.		
Result	The result Would Be **"The Right hand Thumb will move in downward Or Upward Direction according to the co-ordinates that changes "**		
Result Image			

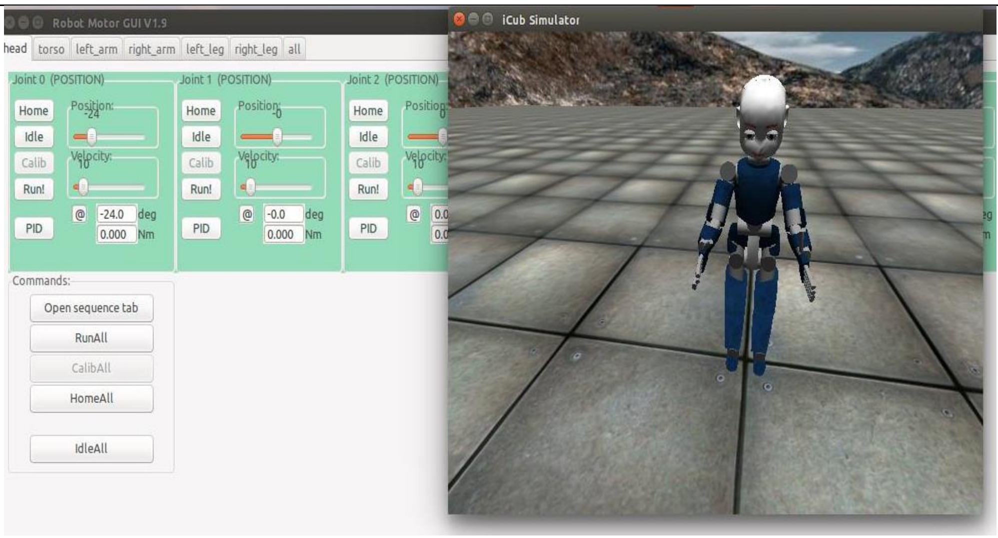
ix. Joint 7 (Position) Finger Expand

Test Case ID	TC-017	**System Tester**	Daniyal Waris
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check Fingers Expand		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case Finger of The Right Hand Is examine whether it Expand.		
Input	Using the scroller to change the position of Finger of The Right hand.		
Result	The result Would Be **“The Finger of Right Hand Would be Expand”**		
Result Image			

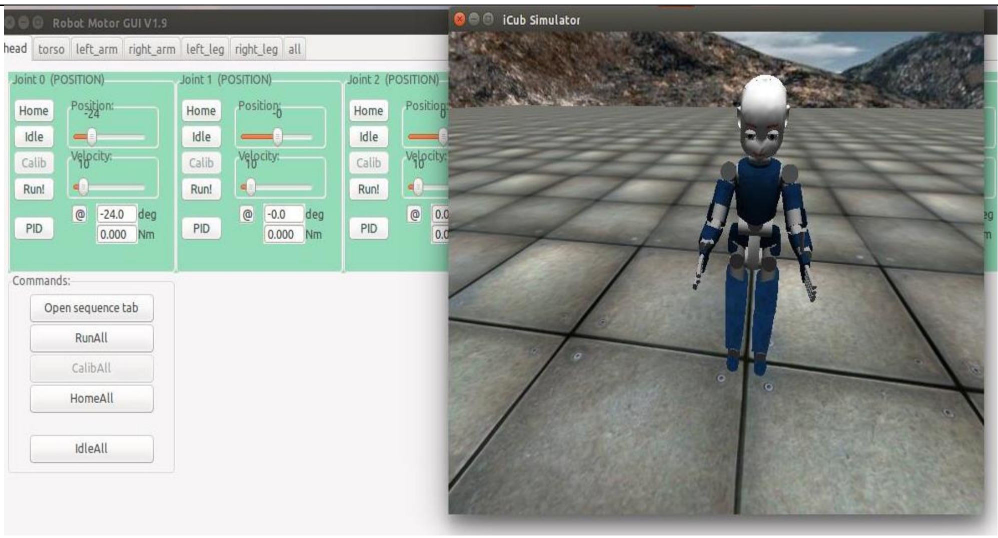
x. Joint 9 (Position) Thumb movement right /left

Test Case ID	TC-019	**System Tester**	Daniyal Waris
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The movement of Right Hand Thumb, Does it Move left or Right		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case Right hand Thumb would move to left or right side from its rest position.		
Input	Using the scroller to change the position of Right Hand Thumb		
Result	The result Would Be **"The Thumb will move Either Left side or Right side as According to the Coordinates"**		
Result Image			

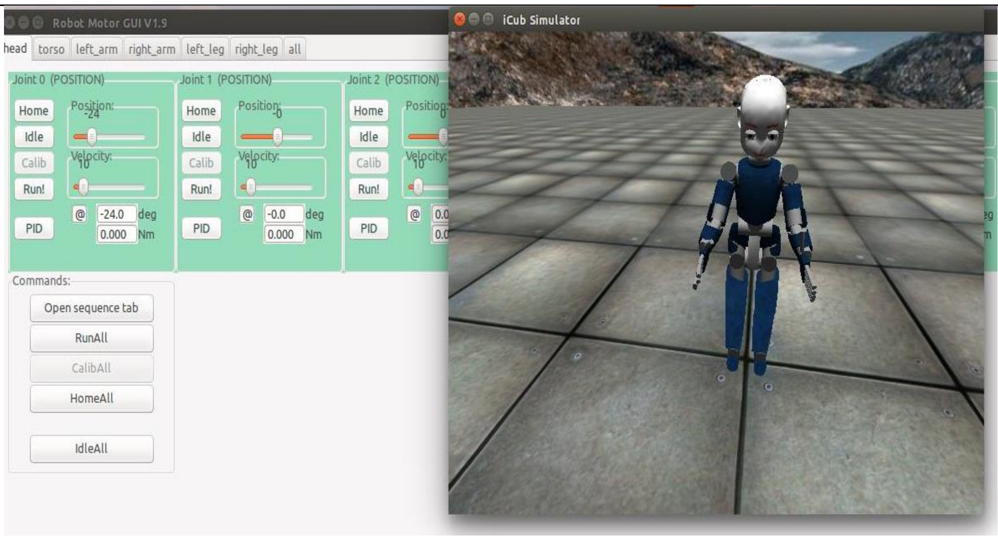
xi. Joint 10 (Position) Thumb movement to grab

Test Case ID	TC-020	**System Tester**	Shahmeer Arshad
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The movement of Right Hand Thumb either it grab Things		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case Right hand Thumb would perform the functionality of Grabbing things		
Input	Using the scroller to change the position of Right Hand Thumb		
Result	The result Would Be **“The Thumb will Grab things”**		
Result Image			

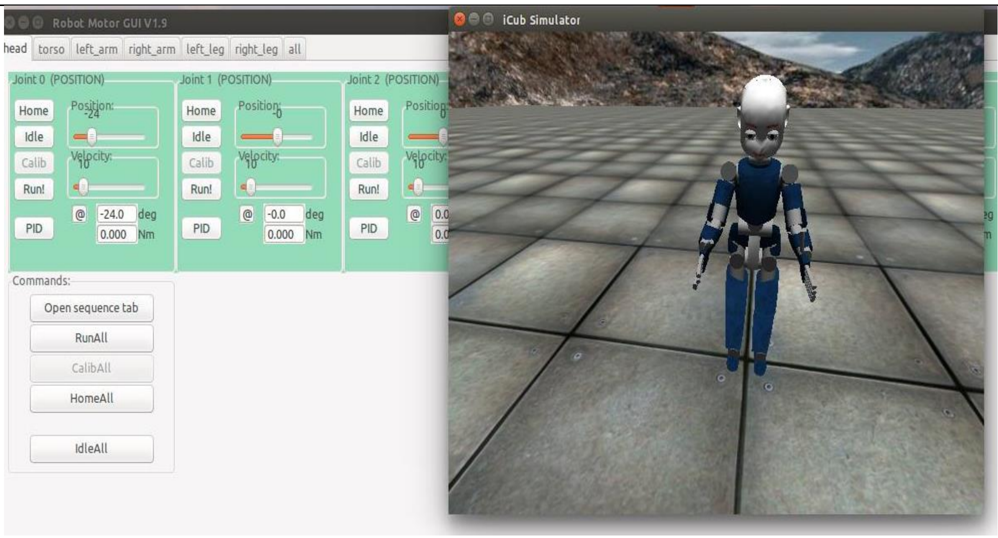
xii. Joint 11 (Position) Index Finger Movement

Test Case ID	TC-021	**System Tester**	Shahmeer Arshad
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The movement of Right Hand Index Finger		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case movement of Right Arm Index Finger been Checked		
Input	Using the scroller to change the position of Right Arm Index Finger.		
Result	The result Would Be **“The movement of Right Arm Index Finger Would be occurred”**		
Result Image			

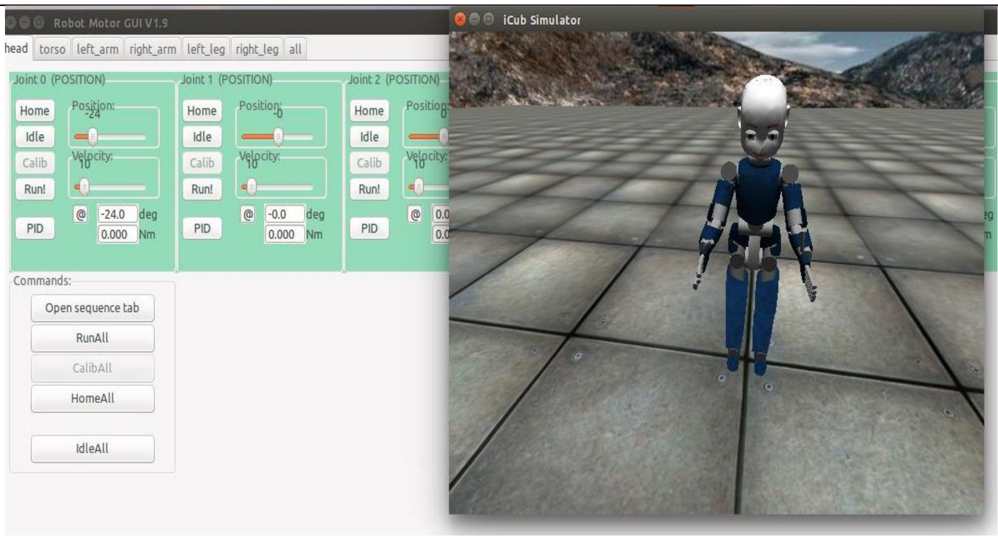
xiii. Joint 12 (Position) Index Finger Grab

Test Case ID	TC-022	**System Tester**	Shahmeer Arshad
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The Right Arm Index Finger Could Grab Things.		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case Right Arm Index Finger would be checked that could it grab things.		
Input	Using the scroller to change the position of Right Arm Index Finger.		
Result	The result Would Be **"The Right Arm Index finger would Grab things"**		
Result Image			

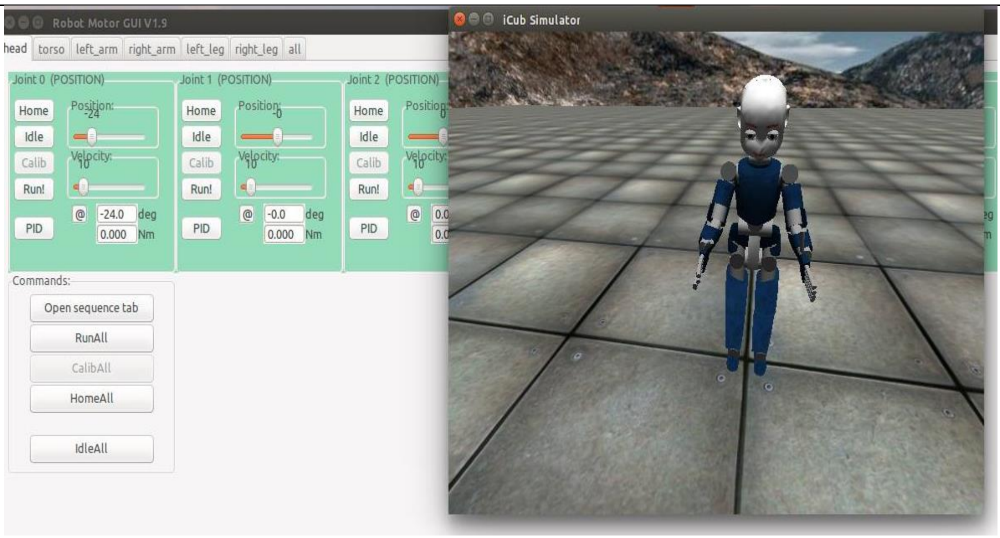
xiv. Joint 13 (Position) Middle Finger movement

Test Case ID	TC-023	**System Tester**	Shahmeer Arshad
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The movement of Right hand Middle Finger		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case Right hand Middle Finger Movement is tested.		
Input	Using the scroller to change the position of Right hand Middle Finger.		
Result	The result Would Be **“The middle finger will move in various Direction as its co-ordinate value changes”**		
Result Image			

xv. Joint 14 (Position) Middle Finger Grab

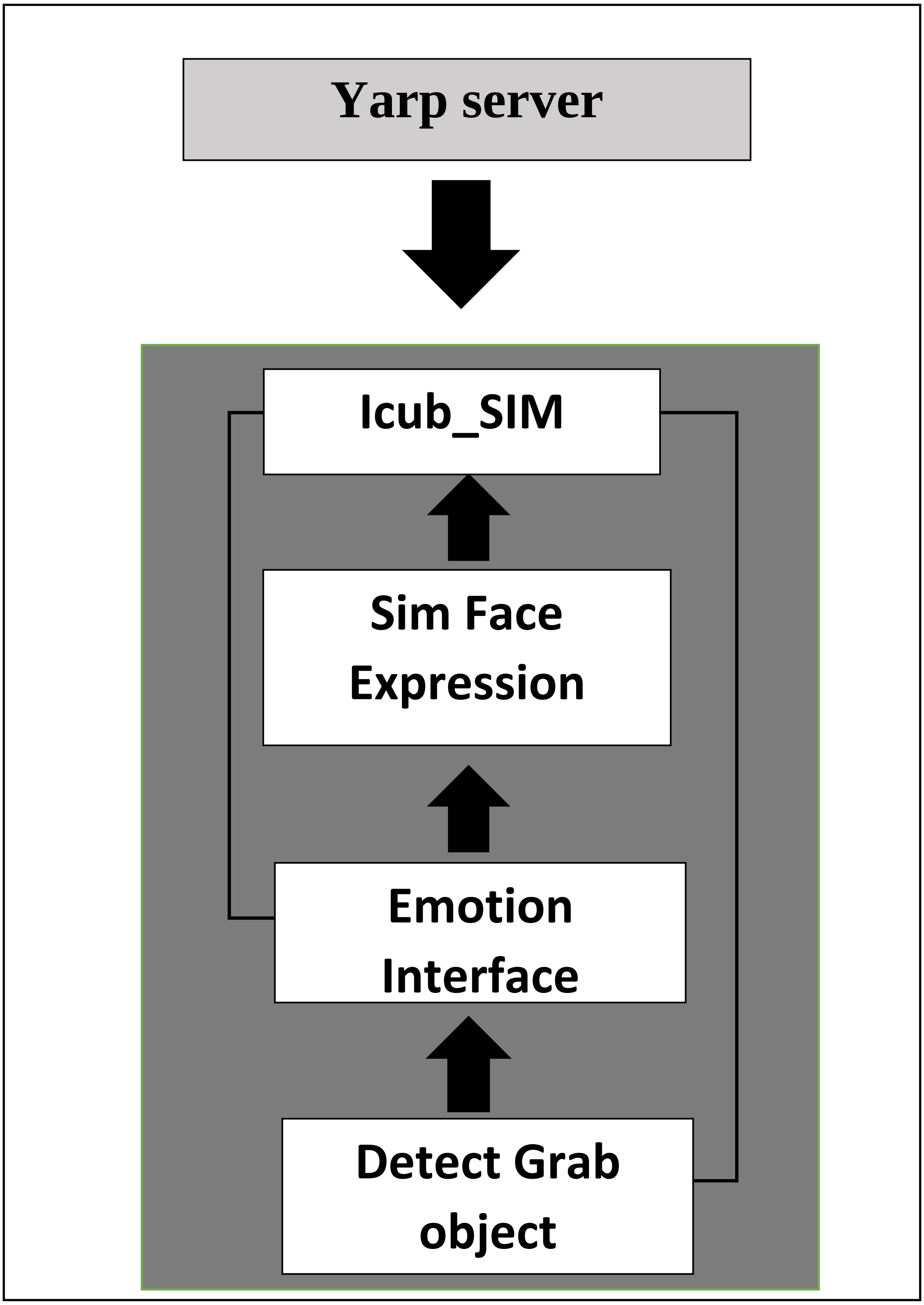
Test Case ID	TC-024	**System Tester**	Shahmeer Arshad
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The Grab Functionality of Right hand Middle Finger		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case Right hand Middle Finger Grab Functionality is been Checked.		
Input	Using the scroller to change the position of Right hand Middle Finger to grab things.		
Result	The result Would Be **“The middle finger will Grab object ”**		
Result Image			

xvi. Joint 15 (Position) Ring and last Finger Grab

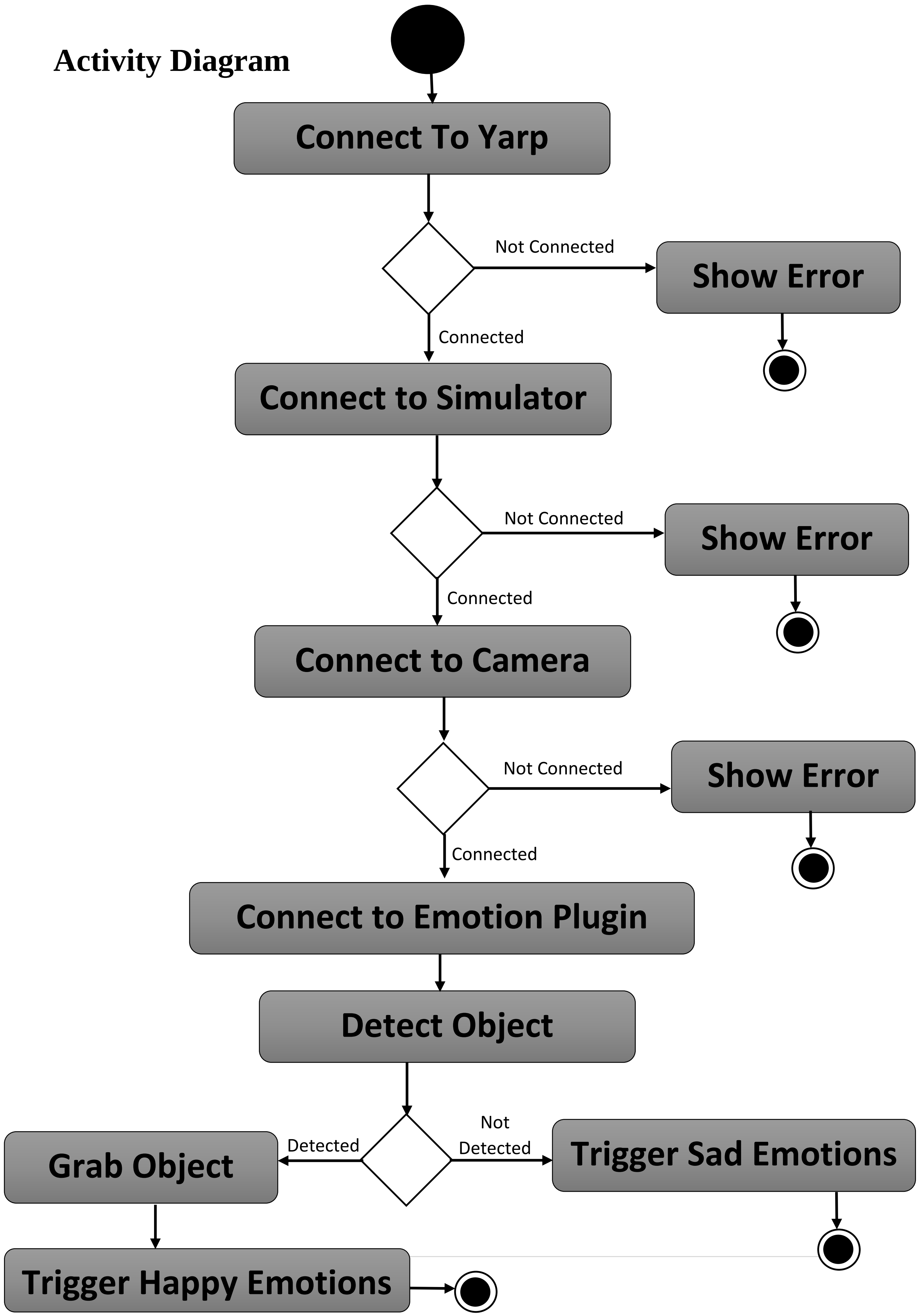
Test Case ID	TC-025	**System Tester**	Shahmeer Arshad
Version	Robot Motor GUI V1.9	**Date**	26-7-2015
Purpose	To Check The Grab Functionality of Right hand Ring and Last Finger.		
Pre Requisite	Robot Motor GUI Interface		
Execution Description	In this test case Right hand Ring and Last Finger, would perform the functionality of Grab an Object.		
Input	Using the scroller to change the position of Right hand Ring and Last Finger to Grab Things		
Result	The result Would Be **"The Ring and Last Finger will Grab Things"**		
Result Image			

iv. ARCHITECTURE:

Architecture tell us about the Internal Communication of data. It is an arrangement that portrays critical Frame works choose from a mixture of points of view.

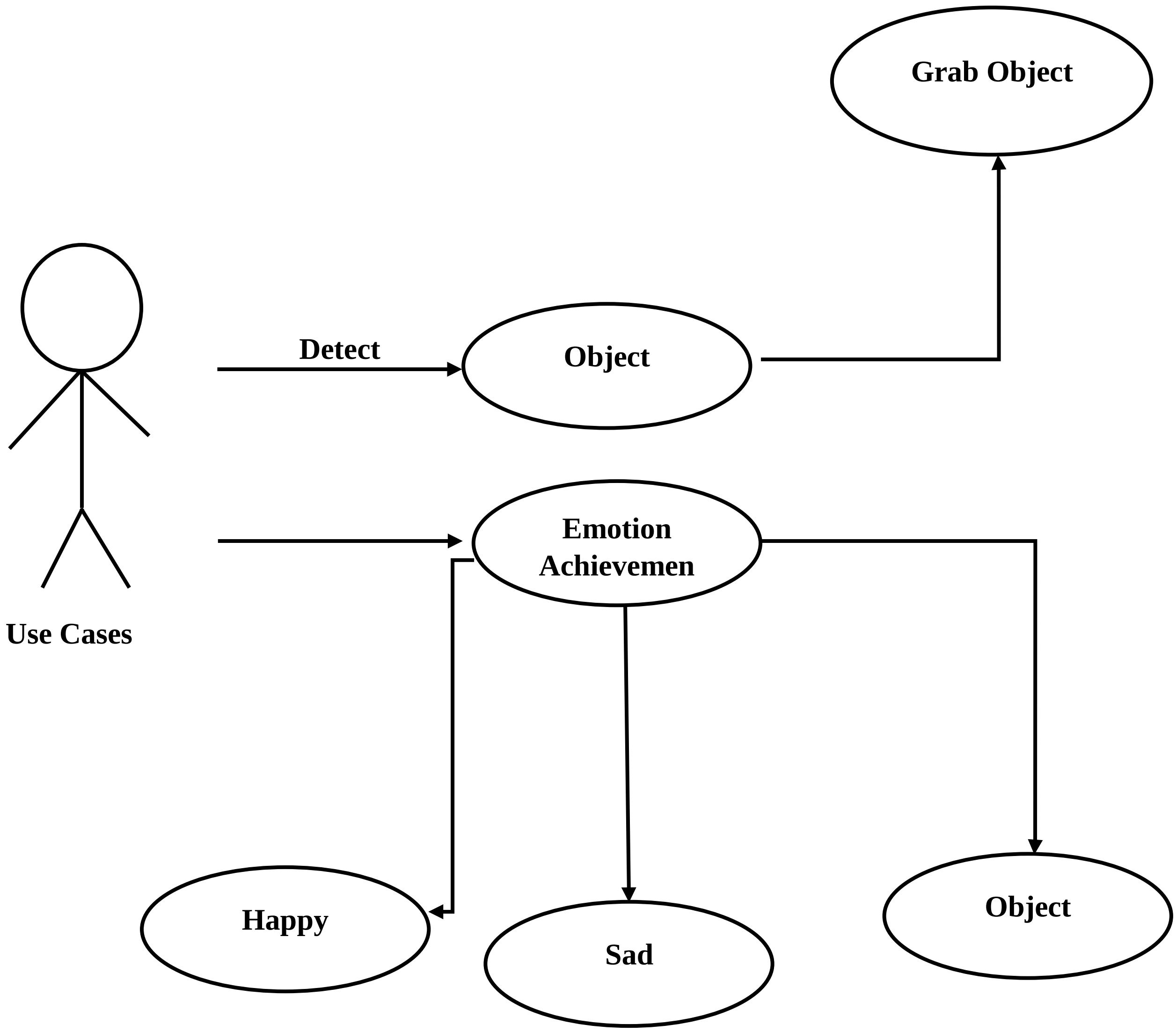


Activity Diagram



Use Cases

Use Case 1



Sequence Diagram

4. IMPLEMENTATION

Thought and ideas are giver physical existence in implementation phase. It is an important phase of software life cycle. In this phase the software development progresses in full swing.

The developers translate the requirements bunched up during requirement engineering phase into executable source code. This includes implementing attributes and properties of each object and integrating all the objects to form a single functioning system. The implementation activity extends over the gap between object and design model and a complete set of source code files that can be compiled together.

<u>Tool and Technologies</u>

- QT Creator
- Yarp Server
- iCub Simulator

5. APPLICATION SETUP

Our application is focused on improving the process of machine learning, we have tried to play our little part in making our simulated robot iCub learn to detect, track and pick up objects in real world like environments.

Compatibility:

Our application is compatible for the following

Software:

- Linux 14.04
- Windows 7

Hardware (optional):

- iCub Robot

Although our work is done in a simulated environment but our application can be burned and run on an actual iCub Robot and it will run the same as it runs on the simulator

v. WORKING

To run the execution properly without any problems we need to confirm that all the required hardware, software and system specifications are available