

# Kinect Based Gesture Controlled Robotic Arm: A research work at HuT Labs

Rajesh Kannan Megalingam, Nihil Saboo, Nitin Ajithkumar, Sreeram Unny, Deepansh Menon  
Electronics and Communication Engineering, Amrita Vishwa Vidyapeetham University, Kollam, Kerala 690525  
rajeshkannan@ieee.org, nihilsaboo@yahoo.com, nitinakumar@gmail.com, unny.sreeram@gmail.com,  
deepu92scorpio@gmail.com

**Abstract**— Undergraduate (UG) Research is the first step in a students' life to taste the experience of a research work. It is also a unique experience for the teachers to guide the students in research work during their UG study. In this paper we want to share our research experience through the project that involves the building of a Robotic arm which mimics the motion of the human arm of the user at Humanitarian Technology (HuT) Labs of Amrita. The system monitors the motion of the user's arm using a Kinect. The skeletal image of the arm obtained using the "Kinect Skeletal Image" project of Kinect SDK, consists of 3 joints and links connecting them. Coordinate Geometry is used to calculate the angles between the links connecting the joints. This gives us the angles for a 3D representation of the human arm. The angles thus obtained are sent using a serial communication port to the Arduino microcontroller, which in turn generates signals which are sent to the servo motors. The servo motors rotate according to the angles given as input. The combined motion of the servos results in a complete Robotic arm movement which is a mimic of the human arm movement.

**Keywords**—robotic arm; kinect; coordinate geometry; human arm; arduino microcontroller; servo motors

## I. INTRODUCTION

The growth of any university is measured in terms of the number of research projects carried out; number of publications in the form of conference, journal etc. Even though there are various other factors to measure, research related to the particular discipline of engineering or science stream – computer science, electrical, mechanical engineering etc., forms undisputable factors in getting reputation and rank worldwide for a university. At our university we introduce research to the undergraduate students even from the second year into their degree. In this paper, through the kinect based robot arm control project at HuT Labs, we want to share the undergraduate research experiences.

This project proposes a robotic arm which mimics the action of human arm. The movement of the human arm in 3d space is captured, processed and replicated by the robotic arm. The Human arm's movement is captured in real time by a camera which is connected to a system which analyses the motion and sends the essential data to the microcontroller which generates the necessary signals to robotic arm.

## II. MOTIVATION

This century has witnessed the growth of Robotics like never before. The prospect of most physical and computational work being done by machines has changed the way the world

functions. Being a high end application of our technological know-how, Robotics has also influenced other areas of human functioning. Industries have witnessed the automation of its processes, increasing the efficiency and decreasing time and human labor. Electronics has also progressed with the coming of this age, as new technologies were discovered for chip fabrication and circuit realization. The challenge to improve these robots has also inspired a whole generation of engineers to push the barriers of technology.

One such challenge is making the robots as human-like as possible. This has many advantages. Under extreme conditions, it is likely that humans will fail to respond in an effective way. This is due to the physical and mental limitations humans have. Here we can use robots to accomplish the same but in a much more effective way. This can even lead to life-saving operations. Real world examples include search and rescue operation for the military and robotic surgery in the field of medicine. Through our UG research work we wanted to develop knowledge of technological know-how of robots, influence of robots in various fields, innovation in Electronics industry and challenges in using robots as substitute for humans etc.

## III. RELATED WORKS

The concept of Gesture control to manipulate robots has been used in many earlier projects. The paper [5] uses a real-time gesture control algorithm to control a robot. The pointing gesture are translated into particular instructions which the robot which direction to move. However the functionality of this robot is restricted to directed movement. The full potential of gesture control is not being used. Considering the issues such as navigation and manipulation, only few robotic systems are equipped with flexible user interfaces that permit controlling the robot by "natural" means. The paper [6] describes a gesture interface for the control of a mobile robot equipped with a manipulator. The interface uses a camera to track a person and recognize gestures involving arm motion.

Most of industrial robots are still programmed using the typical teaching process, through the use of the robot teach pendant. The paper [7] proposes an accelerometer-based system to control an industrial robot using two low-cost and small 3-axis wireless accelerometers. These accelerometers are attached to the human arms, capturing its behavior (gestures and postures). The aim is that the robot starts the movement almost at the same time as the user starts to perform a gesture or posture (low response time). Another

attempt to make Robots, human like is cited in the paper [8] which uses a few desired points in task-space and a control method with Jacobian transpose and joint velocity damping. This is different from other projects of its kind, which normally involves kinematics.

This project involves a simple and straightforward method of using a special algorithm to calculate the required information directly from the skeletal image. This makes it different from other projects which use more complicated methods to achieve the same. Moreover the overall cost and complexity is reduced by the usage of the methods that we have used. The Kinect sensor allows us to directly derive information from the environment, thus making the initial process easier and efficient.

#### IV. BLOCK DIAGRAM

The whole system consists of 2 main parts: Angle determination and data transfer from Kinect and manipulation of robotic arm using Arduino microcontroller. Figure 1 shows the block diagram of the entire system.



Fig 1- Kinect Controlled Robotic arm Block Diagram

#### Kinect

The Microsoft Kinect sensor is used as the input device. It captures the motion of the human arm in real time and transfer the data to the processing unit.

#### Computer

It processes the information from the Kinect and converts it into a skeletal image. It then calculates the angles between the vectors.

#### Serial or wireless communication device

It sends the data (angles) serially from the Kinect to the Arduino. The programming part of both the Kinect and Arduino has to be made to send and receive the data using the computer as an interface.

#### Arduino

Depending on the data (angles) received, the Arduino generates PWM signals designed to move a Servo-Motor to a specific angle.

#### Robotic Arm

The combined movement of the 3 servos at the 2 joints gives the required movement.

The current prototype tracks and mimics the human arm only up to the wrist. The fingers have not yet been included, so as to develop a specific type of application. This movement is achieved using the ‘‘Skeletal Image’’ in Microsoft SDK and a set of computation tools we developed to determine the angles directly from the skeletal image. The program to implement this has been written in Visual Studio in C#. An Arduino program has also been written which generates PWM signals depending on the magnitude of the angles received. The PWM

signals controls servo motors in the Robotic arm. The servo motors turn their axles to the angles which is received at the Arduino from the Kinect, and the combined effect of the three motors give the required movement.

#### V. DESIGN METHODOLOGY

##### Kinect and Angle Computation

###### 1) Necessity of Angle Computation

The proper functioning of the Kinect based gesture control robotic arm necessitates the proper computation of the angles formed by the various links (vectors) and the joints. From a mechanical point of view, there exist a number of joints and links spread throughout the robotic arm but considering the various tasks to be performed by the arm, only three of them are to be monitored in real time. The angles aforementioned include both two-dimensional and three-dimensional ones, which can only be determined by Forward Kinematics. The method of Forward Kinematics, seen on a very frequent basis in almost all the mechanical systems that have been designed up to date, involves calculating the position and the orientation of the End Effector of the robotic arm from the joint parameters. Serial communication is used to transfer the obtained angles to an Arduino microcontroller, which then moves the robotic arm in the requisite fashion.

###### 2) Links and Joints

The robotic arm is a mechanical system that constitutes multiple rigid structures called Links, connected by Joints. Drawing an analogy with respect to the human skeleton framework, Links becomes equivalent to bones while Joints translates to skeletal joints in our body. The degrees of freedom of various Links are governed by the design strategy of the corresponding Joints. The Links and the Joints constitute a co-ordinate frame. The skeleton image program provides the coordinate frame and the various joint parameters on the basis of an arbitrary origin. The Skeleton Image program in Kinect SDK is designed so that an arbitrary origin is set, a virtual co-ordinate system is constructed and the various co-ordinates of the joints are computed. The updated version of the program possesses the critical functionality of angle computation on the basis of the coordinates offered by the basic Skeleton Image program.

###### 3) Kinect Programming Algorithm

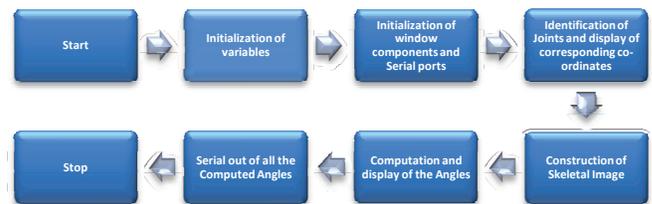


Fig 2- Kinect Programming Flow

The Kinect programming flow as shown in Fig 2 consists of the following sections:

###### a) Initialization of Variables

The data required for preparing the frame for the skeleton image and the colours used for the drawing of the skeleton is initialized. Other variables include the width of the brushes used in drawing the lines joining the joints.

*b) Initialization of Window Components and Serial Ports*

Here, the GUI interface is initialized by preparing frames, buttons and other elements that are accessible by the user for the easy handling of the software. Serial Port with a specific baud rate, Stop bits, Data bits, parity and port name matching to the serial port to which the Arduino is connected, is setup.

*c) Identification of Joints and display of the corresponding co-ordinates*

With the help of Kinect SDK, the retrieved data from the Kinect is easily utilized to identify exact coordinate position of each joints of our body. This is then displayed through the WPF window that is used as the GUI for easy analysis.

*d) Construction of Skeletal Image*

Different functions are created to construct the joints and the links between them, thus obtaining the skeletal image. Colours are assigned to them and the final image is obtained. This is shown in figure 3.

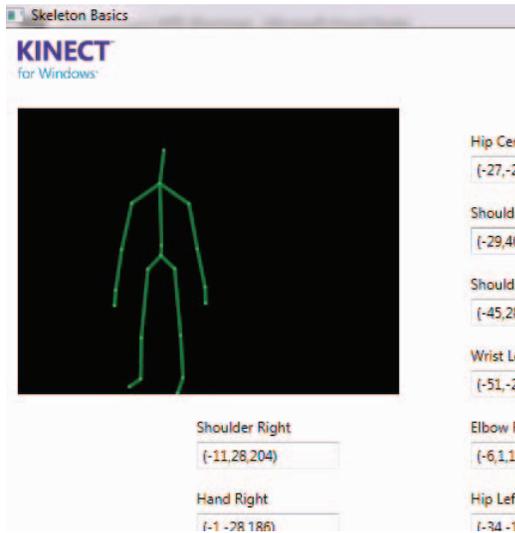


Fig 3: Kinect Image with joints

*e) Computation and Display of Angles*

Using the skeletal image, different vectors are assigned to the links and using parameterization method we obtain the angles. Three angles are obtained, which are to be used as input for the robotic arm. The angles are as follows:

- $\emptyset$ - The angle between the fore-arm and the upper arm.
- $\in$ - The angle between the projection of the upper arm and the spine.
- $\alpha$ - The angle between the projection of the upper arm and the upper arm.

The computed angles are shown in Figure 4.



Fig 4: Image of computed angles

*f) Serial out of all the computed angles*

The computed angles are then sent to the Arduino using a serial communication port.

*Arduino Control*

The angles that are computed by the techniques mentioned in the previous chapter are sent serially to the Arduino. The magnitudes of these angles have to be processed and the necessary control signals have to be generated. Therefore there are two parts to the Arduino program:-

*4) Serial Communication*

The angles that are computed are sent to the Arduino as single characters. This poses a problem as the range of angles that form the actual input can be between 0 to 360 degrees. Therefore the information that is received is stored into a string of three characters at a time. This gives the string a range from 0 to 360 degrees. The string is then converted into an integer type variable which makes it possible to give it as an input.

*5) Servo Control*

The angles, thus obtained, is sent as input to servo motors. The servo motors turn their wheels to the exact degree that is input. The combined movement of the motors moves the arm according to the movement of the user's arm.

*6) Arduino Programming Algorithm*



Fig 5: Arduino Programming Flow

*a) Initializing Serial Communication*

The serial communication should be first established with a specific bit rate (9600) so that information can be sent to the Arduino serially.

*b) Assigning the corresponding pins to servo motors*

The servo motors are connected to specific pins in the Arduino board and initialized as output.

*c) Reading and Type Conversion of received values*

The values received through the serial port are read by the Arduino, and converted into the necessary input format.

*d) Checking angle validity*

There received angles are passed through a conditional loop which determines if the values of the angles are valid.

*e) Sending the required values to motors*

The corresponding angles are given as input to the corresponding servo motors which turn their wheels to the degree that is input.

## VI. MODULE TESTING

The software modules compute the necessary data required for the motion of the robotic arm and transfers them to the hardware module (the robotic arm) which executes the requisite task. The proper functioning of the robotic arm is assured only if there exists a proper compatibility between the software and the hardware. The testing procedure for standalone Kinect involved the software: Visual C#. The skeletal image conjured with the “Kinect Skeletal Image” project in Kinect SDK reflected precisely the motion of the human framework with the state-of-the-art sensors included in the Kinect hardware. The angle computing algorithm, in the Kinect SDK computes the necessary angles with a high degree of accuracy (tolerance  $\leq 10^\circ$ ). The three angles that the algorithm calculated included:

- $\emptyset$ - The angle between the fore-arm and the upper arm.
- $\epsilon$ - The angle between the projection of the upper arm and the spine.
- $\alpha$ - The angle between the projection of the upper arm and the upper arm.

These angles are measured relative to an origin and an arbitrary co-ordinate system fixed by the Kinect SDK. We had set an initial reference angle by the following configuration:  $\emptyset = 180^\circ, \alpha = 0^\circ, \epsilon = 0^\circ$ . This corresponds to the normal posture of a human hand.

The Kinect SDK displayed the various angles in the skeletal image. The testing was done primarily by repeatedly setting a particular configuration of  $\emptyset, \alpha$  and  $\epsilon$  and comparing it with the same set of angles displayed in the Kinect SDK interface. The angle computations were tested by facing towards the Kinect sensor and also facing the sensor at some angle. This ensured that the algorithm works irrespective of the direction the user faces the sensor (the angles however, must be in the visual field of the sensor).

The standalone Arduino testing was done by inputting the numeric data in the serial COM port of the Arduino and by revisiting and refining the program code many times before that the angles could be obtained the way they had to be. The output included the regeneration of the three angles alongside their corresponding identity labels. This allowed no room for any sort of ambiguity and confusion from the point of view of the user handling the module. The efficient transfer of data from the Kinect module and the efficient synthesis of the data in the Arduino module according to the predetermined syntax allowed a precise and efficient functioning of the robotic arm.

The integrated testing involved both Kinect and Arduino connected to a computer via serial port. The angles that are obtained from the Kinect Skeletal program have to be sent serially to the Arduino. For this purpose, a sample program was made in Visual studio so as to transfer data to the Arduino program. In this testing method, the Arduino was connected to a led and when the received data was above a certain value, the led would be lit up. The data was input from the program in Visual studio and send serially to the Arduino program. The received data was analyzed and the required action was taken. This was done many times to check the validity of the code and

the method. When servos are connected to the Arduino and the angles are sent, they are analyzed in a similar way and the servo motors rotate accordingly. This proves that such a communication between the Kinect module and the Arduino module is indeed possible.

## VII. UNDERGRADUATE RESEARCH EXPERIENCES – STUDENTS’ PERSPECTIVE

The experiences we had during our project were truly multifaceted. We faced a lot of trouble managing our academics, which is in itself demanding, along with our research work. So in order to manage time and avoid confusion, we devoted a specific time for our project work after our usual college hours, when we would stay back at the college and prepare for this project. Motivation was really essential for bringing out the best in each of us while doing the project. So we began with discussions about the possibilities of our project. Assessing each of our strengths and distributing our work was critical in the success of our project. We did a lot of research to understand the problem which motivated this project. We read blogs and watched numerous videos online, related to our project, which spurred our motivation level. We had to learn to use different new tools and languages in order to begin our work. This helped us attain new knowledge which we wouldn’t have got if we concentrated only on our college syllabus. However it took about four weeks of learning to actually start working on the project. It was then that real technical challenges started to get clearer. As our project involved multiple hardware (Kinect sensor, Arduino microcontroller and the robotic arm) and software modules (Arduino assembler, Kinect SDK and Visual Studio), challenges arose frequently and this led to the rethinking and redesigning of everything that we had already thought of previously. The project enabled us to properly apply the concepts most of which we had already learnt in our high school and undergraduate courses. The project necessitated frequent thinking and designing so that the ideas could be properly implemented. We learnt that there truly exist significant differences between the mere understanding of a concept and the application of that particular concept. There were a lot of problems that constantly came up and finding the solutions to these problems offered us great challenges, both physically and mentally. The unexpected barriers that sprang up so frequently in our software modules and how we were able to get past them are mentioned henceforth.

We learnt the basics of the language C# through various video lectures and tutorials. The new software packages; Visual Studio, which served as the programming platform for C #and C #in itself offered some difficulties, however with time, and practise our expertise grew and we were able to use these software packages with considerable ease.

### *Kinect SDK*

The Kinect hardware module was learnt from the very basics using various video lectures. Setting up the Kinect Hardware and the corresponding “Kinect skeleton” image in Kinect SDK were the first major problems that we faced. The co-ordinate magnitudes were too small and this forced us to use an additional scaling factor. The scaling factor, when multiplied with the co-ordinate values ensured the requisite

enhancement in magnitude. The problems in displaying these values in the Graphical user interface of the Kinect SDK were solved using the same logic involved in the skeletal image.

#### *Selecting of angles*

The angles that would allow the synchronous motion of the robotic arm with the motion of the Human user had to be selected. The existence of multiple angles forced us to revisit the basic principles of co-ordinate geometry to tackle the problem. The angles were chosen with some insight into the working of the arm, and involved multiple trial and error strategies too. Constant references were frequently made on multiple articles and video sessions on co-ordinate geometry

#### *Angle Implementation*

The algorithm, based upon certain mathematical equations, involved complexities in design since both 2d and 3d angles had to be computed. The algorithm had to set up a virtual plane and create a projection, which served as the framework for computing the 3d angles. This involved orthogonal projections. The algorithm was then merged with the Kinect SDK codes.

#### *Serial Communication*

The problems, though simple in nature lacked proper testing procedure. The testing phase involved LEDs rather than motors as they were not available. This was done by using a sample program to light up LEDs if the angles were received at the Arduino side. The Arduino module problem concerning the bit by bit reading was solved using an appropriate code.

#### *Future prospects*

This project and its design methods can be used in many different applications. Our algorithm, which essentially computes the angles directly from the skeletal image, can be extended to the whole body thus making it possible to make a full size humanoid robot. High torque motors can be used instead of the servo motors so that the arm can be used to lift heavier loads. This will also increase the strength of the arm. Also, sensors can be incorporated to increase its functionality. Another useful modification is making serial transmission wireless thus making the whole system more mobile.

### VIII. BENEFITS OF RESEARCH WORK

The sole purpose of doing a research is to gain knowledge in various aspects which includes theoretical, technical, management, implementation, debugging and humanitarian aspect.

#### a) Theoretical aspect:

To start off with the project, the students had to collect theoretical knowledge which would help them through the course of the project. The tools they used in our project are some necessary tools which could be extended to serve other purposes. For example, by learning C#, we became capable of creating simple windows application to perform different tasks.

#### b) Technical aspect:

The students had to have a clear idea about the latest technological advancements. This project involved the use of Kinect, which was released in year 2009. Learning about this

technology which is at its premature state is a great asset for the students as this technology revolutionized the Gesture Controlled Systems. The technical aspect behind the Kinect involved the understanding of the image processing by taking the input through the RGB camera and the depth detection by the IR transmitter and receiver. The architecture of the entire system consists of Kinect, Arduino, Serial Port Communication and a Processing Unit (i.e. the C# program used in the computer). So technical aspect behind each component had to be learned and applied into the project for the completion of the project.

#### c) Management aspect:

During the course of this project, the problems students faced and actions they took have helped them improved their analytical and problem solving skills which are essential in managing problems when we are working in a firm. They have learned to manage time even through the hectic schedule put forth by their undergraduate studies. When it comes to a team, each member of the team had to understand and realize the cause and work together keeping the team spirit. Leadership quality is improved when one of the member come forward to motivate the rest of the team and help them work with their highest potential.

#### d) Implementation aspect:

Students, through this project are introduced first time in their UG studies on how to carry out a project. This involves various phases: specification, architecture definition, software simulation, hardware implementation, hardware testing and verification and documentation. The students are involved all phases of the project and learnt about them. They also learnt about how to document the project and did the documentation which helped them most in completing this research paper.

#### e) Debugging:

During the debugging process, the students dig deep into the theoretical knowledge they have in order to solve problems. This gives the students clear idea about the topic.

#### f) Humanitarian:

Kinect is designed for the purpose of gaming, but here we extended its application to a technical level. This can further be implemented in different applications which could be used to serve humanitarian purposes. For example, self-learning bots to help the handicapped could be an extension of the application of the project. By this way students experienced how to use technology for humanitarian cause.

### IX. CONCLUSIONS

This project was aimed at introducing UG students to research by creating a Robotic arm which is controlled by the movements of the user's (human) arm. The students' first experience with a research work is always an unique one. Further more, they could not believe that they are able to write a paper and submit to an international conference. They developed their knowledge in various aspects, tested their understanding in the theory class with realtime circuits and simulations, experienced the benefits of UG research work.

This project has been a great learning experience for them. A lot of challenges were faced in the development of this project right starting from the familiarization of C# programming language, to creating an effective algorithm to calculate angles with ease. The literature available online regarding the use of Kinect and the programming associated with it was very limited due to Microsoft copyrights. Hence the “Kinect Skeletal Image” project in Kinect SDK, was used for implementing our algorithm. They had to find a way to redesign the whole program so as to calculate the required information directly.

#### X. ACKNOWLEDGEMENTS

We gratefully acknowledge Amrita Vishwa Vidyapeetham and M A Math, Kerala, which gave us strength and motivation to successfully complete this venture. We would also like to express our gratitude towards Humanitarian Technology Labs of the department of ECE for making available the labs and other research facility.

#### REFERENCES

- [1] Kinecttutorials,Microsoft.com,Retrievedfrom <http://www.microsoft.com/en-us/kinectforwindowsdev/Videos.aspx>
- [2] Angle and vectors, Khan Academy
- [3] Servo,Arduino.com.Retrievedfrom <http://arduino.cc/en/Reference/Servo>
- [4] Serial, Arduino.com. Retrieved from <http://arduino.cc/en/Reference/Serial>
- [5] Van den Bergh, M. ;Carton, D. ; de Nijs, R. ; Mitsou, N. ; Landsiedel, C. ; Kuehnlitz, K. ; Wollherr, D. ; Van Gool, L. and Buss, M. *.Real-time 3D hand gesture interaction with a robot for understanding directions from humans.* RO-MAN 2011,IEEE
- [6] Stephan Waldherr; Roseli Romero and Sebastian Thrun, *A Gesture Based Interface for Human-Robot Interaction.* , Autonomous Robots, September 2000, Volume 9, Issue 2, pp 151-173
- [7] Neto,Pedro;Pires.JN andMoreira,A.P.,*Accelerometer-based control of an industrial robotic arm.* Robot and Human Interactive Communication.2009, RO-MAN 2009
- [8] Woosung Yang ;Ji-Hun Bae ;Yonghwan Oh ; Nak Young Chong and Bum Jae You, *Biologically inspired control for robotic arm using neural oscillator network,*Intelligent Robots and Systems, 2009. IROS 2009