Gesture Control Robotic Arm

D Sreeharsha[1]

[1] B.Tech. Student Dept. of Mechanical Engineering, Sri Venkateswara College of Engineering, Tirupati, A.P., India

Correspondence: D Sreeharsha, B.Tech. Student Dept. of Mechanical Engineering, Sri Venkateswara College of Engineering, Tirupati, A.P., India. E-mail: hsree1998@gmail.com

Received: January 2, 2020 Accepted: January 21, 2020 Online Published: January 31, 2020

doi:10.5539/mer.v9n2p51 URL: https://doi.org/10.5539/mer.v9n2p51

Abstract

Robot plays a vital part in making our lives more facile. The scope of this project is to provide a relation between human and machine by the interaction of human hand and robotic arm. The arm consists of five Degree of Freedom (DOF) and an end effectors, which allows the interaction with the real world. Now the obligations for the controller arise and along the way settled with the exploration of leap motion sensor.

As earlier, robotic arm was controlled by the keypad or joystick which required a lot of practices and calculations to manipulate the robotic arm to reach desired position. The exploitation of the leap motion results in explicitly acquiring for hand gesture and provides set of points.

Keywords: Leap Motion, Components, Softwares

1. Introduction

Robotic arm is a programmable manipulator, it comprises of linear and rotary joints to allow for controlled movements (Robots.com, 2017). It widely used in various field such as industry, medical and military for many years due to its high repeatability, accuracy and efficiency. The flexibility or dexterity of an articulated robotic arm is proportional to the number of axes of it. For the industrial articulated robotic arm, it ranging from different sizes depending on different application, for instance, the big heavy duty articulated arm perform automotive assembly while application such as electronics assembly is performed by smaller articulated arm. In military, articulated robotic arm is used in bomb disposal robot. According to (Ray Allison, 2016), bomb disposal robot is used to disable explosive ordnance over years and saved countless lives. The robot is remotely controlled by bomb expert at a safe distance to examine explosive devices closely without putting themselves in danger. Thus the robot usually equipped with cameras as the "eyes" of the robot to provide the vision of the surrounding situation so that the operator can control the robotic arm through cameras to examine and dispose explosive devices. The robot also equipped with pairs of caterpillar tracks or wheels to allow it to traverse rough terrain such as climbing stairs and tools can be attached on the robotic arm such as wire cutter in order to bypass fences. The body itself also armed with explosive detectors and X- 2 ray devices for the detection of explosives, not only bomb but unexploded munitions and landmines as well. Bomb disposal robot technically is not a "robot", more accurately as a drone, as human control still needed since bomb experts' experience and decision is crucial in explosive disposal operation.

1.1 Aims and Objectives

The objectives of the project are shown as following: -

- To build a 6-axis robotic arm.
- To implement gesture control on the robotic arm.
- To familiar with Leap motion sensor technology.

Gesture control method will be adopted in this project. With this gesture controlled robotic arm, the bomb disposal operation will be higher efficiency as the robot can be operate in faster and more intuitive way and no training is needed.

2. Gesture Control

Gesture control "Gesture control is the ability to recognize and interpret movements of the human body in order to interact with and control a computer system without direct physical contact." The interface of these system is known as natural user interface (NUI), generally lack of intermediate devices between the system and the user. (Gartner IT

Glossary, 2017) Gesture control is much more effortless and intuitive compared to pressing switches, tweaking knobs, manipulating mouse and touching screens. It especially contributes to ease the interaction between user and devices, replacing or reducing the need for keyboard, mouse or buttons. Human language will be more understandable by the computers and that will create a better user experience through gestures when face recognition and voice commands such advanced user interface technologies combined together. (Dipert et al., 2013) Gesture control is being used in medical applications, alternative computer interfaces, entertainment applications and automation systems. In medical applications, life threatening conditions can be recognized with advanced gesture recognition robotic systems. For the entertainment 5 applications, gesture control can provide more intuitive control environment that can immerse the player in the game like never before while in the automation systems in vehicles, homes and offices, gesture control can be incorporated to reduce the necessity of primary and secondary input systems such as buttons in the car entertainment systems as that will distract the drivers when the driver wants to control it while driving.

3. Block Diagram (Input/Output)

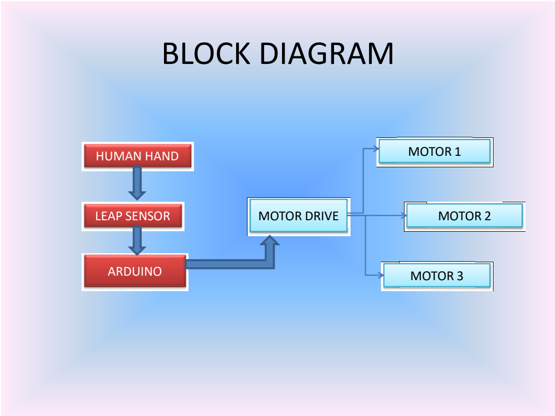

Figure 1. Block Diagram

4. Leap Motion Controller

The leap motion controller is a little USB fringe contrivance, which brings controlling of virtual reality into being, Utilizing two monochromatic IR cameras and three infrared LEDs.

The data that retrieved from the leap motion contains various information's from the positioning frame id to the detection of bones and fingers. Leap motion can detect a human hand within the distance of one millimeter and consuming time to be 300 second. As per for these information leap motion is a good choice to be integrated with the robotic arm.

Figure 2. Leap Motion

4.1 Leap Motion Co-Ordinates

As shown in the Figure leap motion sensor works in 3D coordinates (X, Y, Z). After the obligation of the position of a human hand the data is then send to the Pc or controller due to the reason that the machine does not know how to get to the specific position 1. The controller converts the data obtained from the leap and convert it into coordinates onto which robotic arm can move.

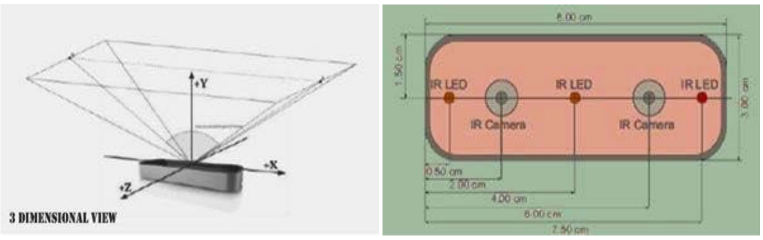

Figure 3. Leap Motion Coordinates

Motion tracking data generated by the sensor can be displayed in the Leap motion visualizer in the form of modelled 3D hand as shown in the Figure below. Various fps (frame per second) such as render fps, data fps and device fps were shown at the top right corner of the window. By referring to the specification of the Leap motion sensor, which claimed the tracking can have a fps of up to 120, which is very accurate. The basic three axes coordinate of the hand also been tracked and displayed here as well as the speed. Thus, the Leap motion visualizer was used to ensure the hand or gesture is being well tracked by the sensor and also visualization can be done to get a better understanding of how the gesture looks like.

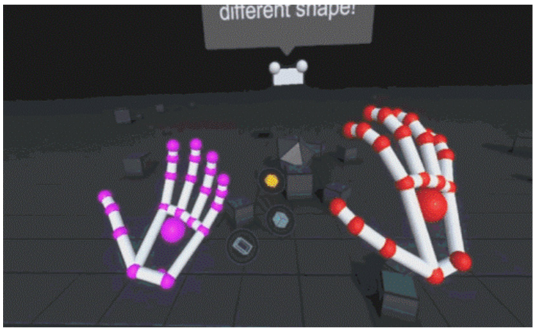

Figure 4. Leap Motion visualizer

5. Hardware Components

- As the motors consist of 3 servomotors connected to the robotic arm with the respected required torque. The hand gesture is acquired by the leap motion is then transmitted to the mechanical arm and mimics the gesture position. The equipment comprises of a straightforward open equipment plan for the Arduino board9. The motors used in the movement of a robotic arm, are as follows: Generally, the mechanical arms utilize Servo motors for activation because of its consolidated advantage of exactness, accuracy and torque.
- Servo Motor
- Elbow
- Gripper
- U-Bracket

5.1 Servo Motor

Figure 5.

The forward kinematic equations of the manipulator are as follows:

Symbol Terminologies:

θ: Rotation about the z-axis.

d: Distance on the z-axis.

a: Joint offset.

α: Joint twist.

Only ‘θ ‘and ‘d’ are joint variables.

5.2 Gripper

These are the simplest robot grippers, **suitable for many industrial products** and easy to manufacture. Within this group, different alternatives can be found: with opening control, pressure control, with distance control in the opening and closing, picking up pieces by inserting the two fingers inside a hole. They can also have pneumatic or electric actuation.

Our favorite models are the one from On-Robot with its Plug & Play feature in the Universal Robots that makes them easy to use in a factory.

Figure 6. Gripper (ARM)

5.3 U- Shaped Brackets

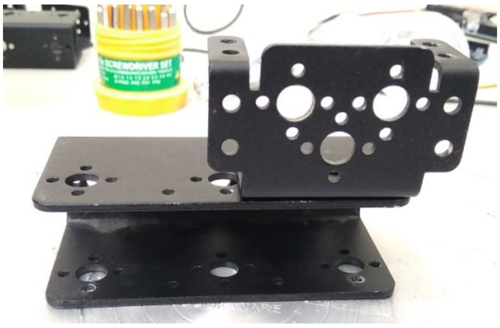

Figure 7. U-Shaper Bracket

6. Coding in Processing Software

Importing libraries is a very crucial step in code writing, since there are some important data such as configuration data, classes and parameter types that were predefined inside, those data need to be imported when the programmer needs it for program development. There are few libraries were imported in the code by writing "import" in front of the library name as shown in Figure 3.5.1.1 below, those libraries are: "de.voidplus.leapmotion.*" for extracting the motion tracking data for the gesture captured by the Leap motion sensor, "controlP5.*" and "g4p_controls.*" for creating the GUI (graphic user interface) and the last one is "processing.serial.*" to enable the serial communication function of Processing software in order to send data between Processing software and Linkit smart 7688 duo board.

```
import de.voidplus.leapmotion.*;   // lib : leap motion for processing
import controlP5.*;                // lib for GUI
LeapMotion leap;                   // var. leap, LeapMotion type.
import g4p_controls.*;             // lib for GUI
Serial myPort;                     // declare the port as serial type
import processing.serial.*;        // serial lib
```

6.1 Data Extraction and Conversion

This section is the main part of the entire system that shows how the tracked gesture data being extracted and processed by the codes. The main 6 data of the hand to control the robotic arm were extracted by accessing the "hand1" class that created earlier. The statements were extracting the position of the hand, stabilized position of the hand which includes the x, y and z coordinates that need to be access later, roll of the palm in degree that used to control the wrist of the robotic arm, the time for the hand visible on the sensor that used to confirm whether the hand is appeared on the sensor, the pitch of the palm in degree that used to control the wrist of the robotic arm too and lastly, the sphere radius of the palm or palm size that used to control the gripper of the robotic arm.

```
PVector hand1_position    = hand1.getPosition();
PVector hand1_stabilized  = hand1.getStabilizedPosition();
float   hand1_roll        = hand1.getRoll();
float   hand1_time        = hand1.getTimeVisible();
float   hand1_pitch       = hand1.getPitch();
float   hand1_grip        = hand1.getSphereRadius();
```

6.2 Setting up Servo Motors

The task of the Linkit board is interfacing servo motors and gesture data from Processing software. Certain pins must be reserved for the servo motors in order for sending PWM (pulse width modulation) signal from the Linkit board to servo motors. Figure 3.6.1.1 shows that how the pins were assigned for the servo motors. For instance, the pin 3 was assigned to servo 0 which is for the x coordinate, meaning that this servo 0 is responsible to turn left and right for the robotic arm. For the Figure 6.1, which 23 is the pre-setting the angle for the servo motors, this pre-setting is for the robotic arm starting pose.

```
servo[0].attach(3);   //x
//delay(100);
servo[1].attach(4);   //for z
//delay(100);
servo[2].attach(5);   //for y
//delay(100);
servo[3].attach(6);   //for roll
//delay(100);
servo[4].attach(7);   //for pitch
//delay(100);
servo[5].attach(8);   //for gripper
//delay(100);
```

Figure 8. Setting Up Pins for Servo Motors

```
servo[0].write(110);  //for x
//delay(100);
servo[1].write(50);    //for z
//delay(100);
servo[2].write(45);   //for y
//delay(100);
servo[3].write(90);   //for roll
//delay(100);
servo[4].write(0);   //for pitch
//delay(100);
servo[5].write(90);   //for gripper
//delay(100);
```

Figure 9. Pre-Setting The Angle for The Servo Motors

7. Robotic Arm Outlook and Result

The robotic arm was built with six servo motors, servo motor bracket holders and a gripper. It powered up by power bank with 5 Volts and 2.1 Ampere output. There are 3 pins in the cable of servo motor which positive, negative and signal pin. The positive and negative pin of the servo motor connected to the positive and negative wire that separated out from the USB to micro USB cable of power bank. The servo motor signal pin connected to the assigned pin on the Linkit board for receiving signal.

Figure 10.

8. Conclusion

In conclusion, the 6-axis robotic arm was built and it is able follow the gesture of the user. Six (6) for the axis number of a robotic arm is consider high, 6-axis was chosen to build is because when the number of axis higher meaning that higher the dexterity of the robotic arm is. Next, with the technology of Leap motion, gesture control is successfully implemented on the robotic arm, meaning that the robotic arm is now able to follow the user hand gesture movement, controlling the robotic arm intuitively. This project showed how convenient is the robotic arm controlled by this pendrive-size Leap motion sensor with a computer. This addressed the problem of using typical controller of the bomb disposal robot. Leap motion sensor can capture the gesture with high accuracy, according to the its manufacturer, the

Leap motion sensor can capture the gesture up to 200 frame per second by two infrared cameras with accuracy up to 0.2 mm. (J.S., A, 2016) This technology can be applied to many applications especially the one that required human control such as medical surgery, controlling the robotic arm remotely in operation with high accuracy to avoid infection. One of the limitation of this project is the robotic arm is bonded to the computer for serial communication. Wi-Fi technology can be implemented into this project in future improvement work for better portability of the robotic arm and user can remotely control the robotic arm anywhere for better convenience.

References

39 Robots.com. (2017). In A. Sarkar, K. Arvindbhai Patel, R. K. G. Ram, & G. Krishna Capoor (Eds.), *What is a robot arm?* Retrieved from https://www.robots.com/faq/show/what-is-a-robot-arm

Army Technology. (2014). Detect and diffuse – The top 5 military robots for explosive ordnance disposal. Retrieved from http://www.armytechnology.com/features/featuredetect-and-diffuse-the-top-5-military-robots-forexplosive-ordnance-disposal-4372678/

Cost-effective hand gesture computer control interface - IEEE Xplore Document. Retrieved from http://ieeexplore.ieee.org/document/7804555/

De Luca, A. (2018). *Inverse kinematics*. Retrieved from http://www.diag.uniroma1.it/~deluca/rob1_en/10_InverseKinematics.pdf

Future with Gesture Recognition Technology Control and Applications. [online] Edgefx Kits Official Blog. Retrieved from http://www.edgefxkits.com/blog/what-is-gesture-recognition-technology-glimpseof-the-future/

Gartner IT Glossary. (2017). *The gesture interface: A compelling competitive advantage In the technology race | EE Times*. [online] EETimes. Retrieved from http://www.eetimes.com/document.asp?doc_id=1280756

Gondane, R. (2017). *Gesture Control - Gartner IT Glossary*. Retrieved from http://www.gartner.com/it-glossary/gesture-control/

J. S., A., & J. L., M. (2016). *Development of a Robotic Arm and implementation of a control strategy for gesture recognition through Leap Motion device*. Retrieved from http://ieeexplore.ieee.org.libezp.utar.edu.my/document/7528373/

Ray Allison, P. (2016). *What does a bomb disposal robot actually do?* Retrieved from http://www.bbc.com/future/story/20160714-what-does-abomb-disposal-robot-actually-do

Sziládi, G., Ujbányi, T., & Katona, J. (2016). *Gesture Control of Drone Using a Motion Controller*. [online] IEEE. Retrieved from http://ieeexplore.ieee.org.libezp.utar.edu.my/stamp/stamp.jsp?arnumber=7462401

Y., P., P., C., N., T., K., K., & Ch., P. (2016). *Robot-arm control system using LEAP motion controller*. [online] IEEE Xplore. Retrieved from http://ieeexplore.ieee.org.libezp.utar.edu.my/document/7782091/

Copyrights