DESIGN OF A GESTURE CONTROLLED ROVER WITH 5 DOF MANIPULATOR FOR EXPLORATION AND RESCUE APPLICATIONS

Abhishek Nair[3], Madan Jagtap[2], Suyog Patil[3], S.N. Teli[4]

[1]Saraswati College Of Engineering, India, asnnovember@gmail.com

[2]Saraswati College Of Engineering, India, jagtap.aero@gmail.com

[3]Saraswati College Of Engineering, India, suyogpatil8688@gmail.com

[4]Saraswati College Of Engineering, India, shivanandteli@yahoo.com

Abstract: With the emergence of robots in many industries, various tasks that were once considered complex for humans are now effortless. Their functionality and accuracy has proven to be lucrative as well productive in various production sectors. This paper presents the design of a rover employing a rocker bogie mechanism and a 5 degree of freedom robotic arm for manipulation. The movements of the rover are controlled using the Myo armband. It is a independently working gesture recognition system that does not rely on any external sensors (motion capturing system) as it has its sensors incorporated in itself which recognizes the gesture commands and acts accordingly. To operate the robotic arm at higher speeds and with better accuracy, we need a new approach to bridge this gap. The Leap Motion Technology, a latest invention in Human-Computer interaction area is the solution to intricacies with operation. This remarkable piece of technology tracks a human hand in air accurate to millimeter. The position of hand is then used to calculate the joint angles that in turn help us to rotate the robotic arm joints by the computer with blazing speeds.

Keywords: Leap motion controller, Myo, Rocker-bogie, Arduino, Inverse kinematics.

INTRODUCTION

Robots are consistently integrated in various fields to relieve human workers of the arduous and tedious tasks. They are currently used for applications including office, military tasks, Hospital operations, industrial automation, planetary exploration, security systems, dangerous environment and agriculture [1]. The proposed design of the robot makes it capable of treading through rough terrains and regions that cannot be accessed by human beings. The rover could be implemented for exploration, surveillance and rescue applications. The Robot is designed to go into slightly destroyed areas to find help rescue people. The robot is even able to travel through fairly large amounts of rubbles and at high speeds on flat planes. On the robot there will be a camera which is used to take video. The robot will be built to discover areas which people cannot reach. For planetary exploration the rover will be capable to tread and survive in harsh conditions and even retrieve samples such as soil and rocks using the manipulator mounted on top it .The robot will be wirelessly communicated by establishing a Bluetooth communication. For this design we are using Arduino UNO as the microcontroller and HC05 Bluetooth model for communication with the user. No remote controller could possibly be compared to the dexterity of a human hand. Here, the rover as well as the manipulator are controlled by human hand gestures effortlessly and effectively without relying on the traditional remote controllers that are sophisticated and require skilled operators.

MECHANICAL DESIGN

Rocker Bogie Design

The Rocker-Bogie system was built to be used at slow speeds. It is capable of overcoming obstacles that are on the order of the size of a wheel [2].The Solidworks design in Fig. 1, displays an inexpensive Rocker bogie suspension system with minimum energy consumption. The rocker-bogie design has no springs and stub axles for each wheel, allowing the rover to climb over obstacles, such as rocks, that are up to twice the wheel's diameter in size while keeping all six wheels on the ground. The rover will possess a 4 wheel drive system i.e 4 DC motors will be used to power the front and rear wheels.The motors will be driven with the help of a motor controller such as a L298N motor driver module.

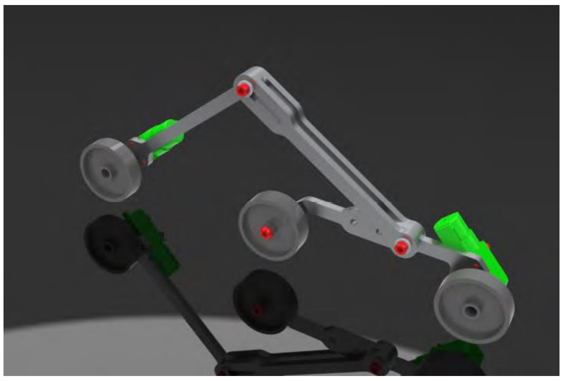


Fig 1: Solidworks design of the rocker-bogie.

Robotic Arm Design

This 5 DOF robotic arm delivers fast, accurate, and repeatable movement. The robotic arm is made up for numerous joints that replicate human arm joints i.e., base rotation, single plane shoulder, elbow, wrist motion, and a functional gripper as shown in Fig. 2.All these joints are driven by servo motors of different torque capacities. These servo motors do not require a motor driver module as they can be directly driven by the power supply provided by the microcontroller.

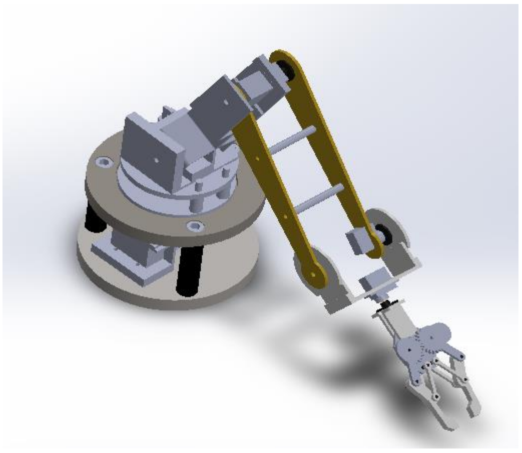


Fig 2: Solidworks design of 5 DOF robotic arm

EXISTING REMOTE CONTROLLERS

Traditional remote controllers are clustered with countless number of buttons. It is evident from Fig. 3 that existing controllers are quite sophisticated to manipulate robots. We need much simpler interactive system to handle this task.

Fig 3: A traditional Remote controller

LEAP MOTION SENSOR

The Leap Motion Sensor is a small USB peripheral device, designed to place on a physical desktop, facing upward. Using two monochromatic IR cameras and three infrared LEDs, the Device observes a roughly hemispherical area to a distance of about 1 meter. The Leap motion sensor procures information such as position, orientation, frame ids and finger bone information. The sensor is capable of tracking the human hand 300 times a second with millimeter accuracy. The heat signatures from the Hand is sensed with the two monochromatic IR sensors present inside and the other infrared LEDs determines the shape structure of the hand. The amalgam of these functions offers the desired data of positions of each joint in hand including bone of fingers. The two monochromatic IR sensors determines the movement of the hand whereas remaining three IR LEDs concentrate on the joints of hand. This makes the leap motion sensor an appropriate controller for the robotic arm. Fig. 4 shows the top view layout of Leap Motion Sensor [3].

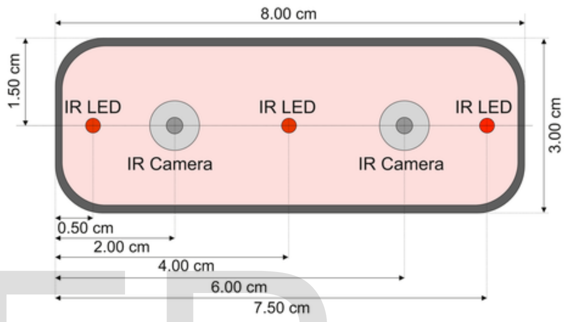


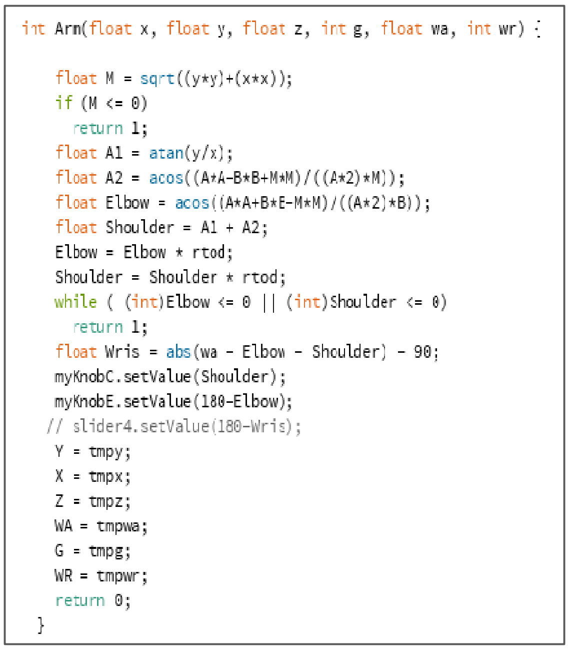
Fig 4: Top view of Leap Motion Sensor.

Inverse Kinematics

Inverse kinematics employs kinematic equations of a robot to determine the joint parameters that provide a desired position of the end-effecter. Inverse kinematics converts the motion plan into joint actuator trajectories [4] for the manipulator. It can be also defined as the method for determining the joint angles using end structure for direct manipulation [5]. It helps in solving mathematical equations which is used to determine the path of the robotic arm. Depending on the parameters of the robotic arm the equation of inverse kinematics vary. It depends on the length of the joints, degree of freedom and end points [3].

Leap Motion Interface Code

To interface the leap motion controller with the robotic arm we employ the processing IDE software. The JAVA program for calculating the inverse kinematics of the arm is entered in to the software. The code is written with the help of leap motion processing library. After retrieving the hand information from leap motion controller the JAVA code calculates the joint angles using inverse kinematics. This data is then sent to the Arduino IDE sketch to rotate the joints of the robotic arm. Fig. 5 illustrates the JAVA code for inverse kinematics of the robotic arm using Leap Motion controller.



```
int Arm(float x, float y, float z, int g, float wa, int wr) {

    float M = sqrt((y*y)+(x*x));
    if (M <= 0)
      return 1;
    float A1 = atan(y/x);
    float A2 = acos((A*A-B*B+M*M)/((A*2)*M));
    float Elbow = acos((A*A+B*E-M*M)/((A*2)*B));
    float Shoulder = A1 + A2;
    Elbow = Elbow * rtod;
    Shoulder = Shoulder * rtod;
    while ( (int)Elbow <= 0 || (int)Shoulder <= 0)
      return 1;
    float Wris = abs(wa - Elbow - Shoulder) - 90;
    myKnobC.setValue(Shoulder);
    myKnobE.setValue(180-Elbow);
   // slider4.setValue(180-Wris);
    Y = tmpy;
    X = tmpx;
    Z = tmpz;
    WA = tmpwa;
    G = tmpg;
    WR = tmpwr;
    return 0;
  }
```

Fig 5: Processing IDE code for inverse kinematics of robotic arm

MYO ARMBAND

Myo armband is a wearable device incorporating eight EMG muscle activity sensors built around its periphery along with a nine axis inertial measurement unit consisting of accelerometer, gyro and magnetometer. The EMG sensors pick up the electrical potential engendered by the muscle cells and with the Myo wrapped around the forearm, the sensors can read all of the muscles that control your fingers, letting them spy on finger positions as well as grip strength [6].The Myo encompasses a rechargeable lithium battery, an ARM processor, Bluetooth 4.0 LE and a micro USB port for charging. Fig 6 illustrates the elements of Myo armband.

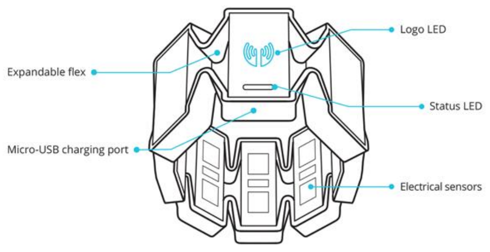


Fig 6: Myo armband structure

Myo Gestures to Control the Rover

The Myo armband gestures are implemented to control the translational motion of the rover. To direct the rover to a particular direction various gestures are utilized via which the robot will capable of moving forward, backward, and left and right. Fig. 7 represents how the gestures are assigned to control the rover. By implementing the same gesture for forward motion i.e. Spread Fingers, the rover's motion is stopped.

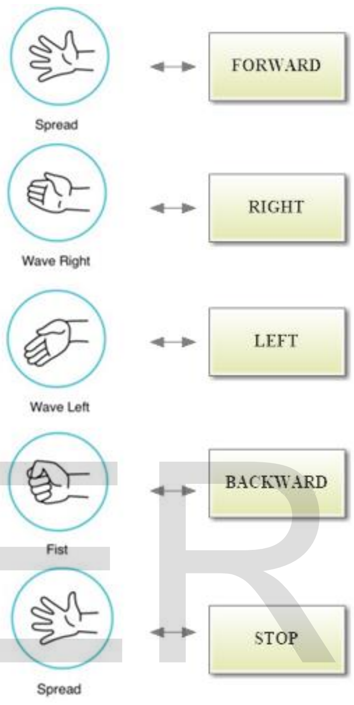
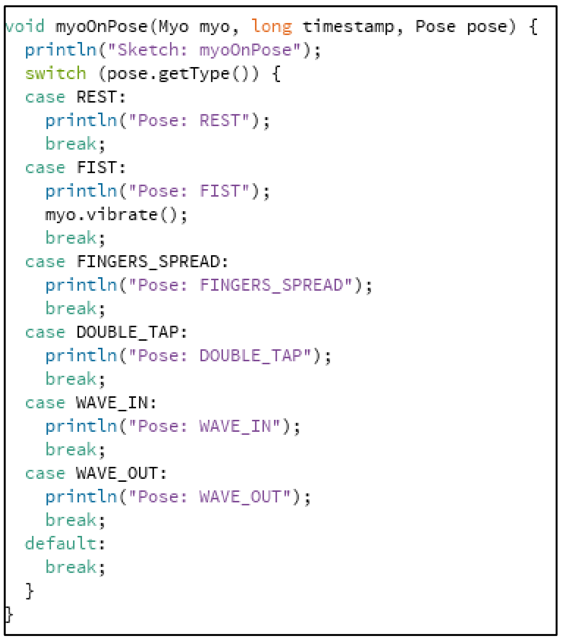


Fig 7: Various gestures used to control the rover

Myo Armband Interface Code

For Bluetooth connectivity and data transmission the software development kit (SDK) furnishes all the details linked to it. The application receives the data from SDK in form of three events i.e. gestural events, spatial events and auxiliary events [7]. The EMG data required for controlling the rover is obtained using the JAVA program implemented in Processing IDE. The Myo processing library is utilized to write the code. Values retrieved from the armband gestures are sent to the Arduino UNO microcontroller to drive the motors of the rover. Fig. 8 illustrates a JAVA program that displays the type of pose depending on the input from the Myo armband.



```
void myoOnPose(Myo myo, long timestamp, Pose pose) {
  println("Sketch: myoOnPose");
  switch (pose.getType()) {
  case REST:
    println("Pose: REST");
    break;
  case FIST:
    println("Pose: FIST");
    myo.vibrate();
    break;
  case FINGERS_SPREAD:
    println("Pose: FINGERS_SPREAD");
    break;
  case DOUBLE_TAP:
    println("Pose: DOUBLE_TAP");
    break;
  case WAVE_IN:
    println("Pose: WAVE_IN");
    break;
  case WAVE_OUT:
    println("Pose: WAVE_OUT");
    break;
  default:
    break;
  }
}
```

Fig 8: JAVA code which prints the type of pose after receiving input from the user's gestures

WIRELESS COMMUNICATION

Communication between Leap Motion Sensor and Robotic Arm

The communication between the user's arm gestures and the robotic arm is established via Bluetooth. The leap motion controller is connected to the laptop with the help of USB 2.0 cable. Required data for controlling the servo motors on the robotic arm is sent to the HC-05 Bluetooth module of the Arduino UNO from the laptop, as shown in the Fig. 9. The HC-05 module is operating in slave mode during this operation.

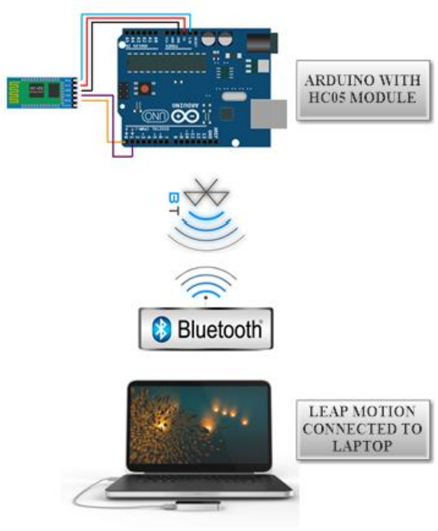


Fig 9: Interface between Leap motion controller and Arduino

Communication between Myo Armband and Rocker-Bogie Rover

The laptop helps in creating the communication network between the Myo armband and the Bluetooth module of Arduino, as shown in Fig. 10. Myo creates a link with the laptop via Bluetooth. The laptop sends the data from the Myo armband to the HC05 module of the microcontroller to drive the rover in the desired direction. The hand gesture signals from the user's armband are sent at high speeds to control the rover's movements.

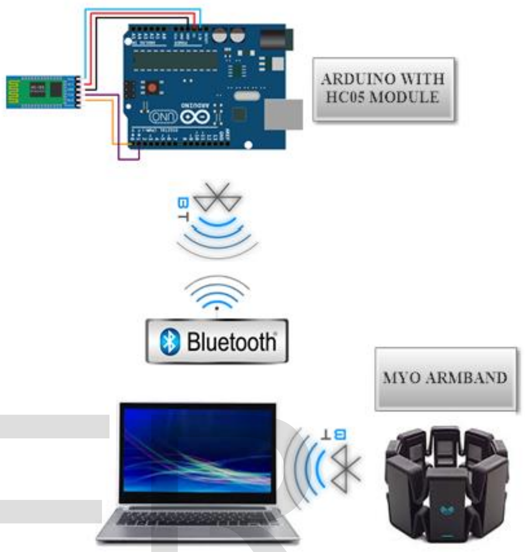


Fig 10: Interface between Myo armband and Arduino

CONCLUSION

Simultaneous control of the rover as well as the manipulator is undoubtedly very complex with traditional controllers. The gesture recognition system is indeed the most effective method to bridge this gap. With the Myo and Leap Motion Controller, a single user is capable of controlling the rover's motion and the robotic arm's movements at the same time. This incredible technology makes controlling a sophisticated robot elementary for any user, regardless of their skills on operation. This lucid system can help us avoid our reliance on bulky control systems. Gone are those days when we had to maintain and manage separate controllers for robots. Gesture control could be the solution to all those problems.

REFERENCES

[1] Basil Hamed, "Design and Implementation of Stair-Climbing Robot for Rescue Applications," International

Journal of Computer and Electrical Engineering vol. 3, no. 3, pp. 461-468, June 2011

[2] Miller, D. and Lee, T. “High-Speed Traversal of Rough Terrain Using a Rocker-Bogie Mobility System,” American Society of Civil Engineers, Space 2002 and Robotics 2002:, pp. 428-434, June 2002

[3] Venna, Trinadh Venkata Swamy Naidu; Patel, Sarosh “Real-Time Robot Control Using Leap Motion A Concept of Human-Robot Interaction,” presented at Northeast Section Conference of the American Society for Engineering Education,April 30,2015

[4] Werner, A., Lampariello, R., Ott, C., "Trajectory optimization for walking robots with series elastic actuators,” presented at Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on, vol., no., pp.2964, 2970, 15-17 Dec. 2014.

[5] Lukas Barinka, Ing. Roman Berka, “Inverse Kinematics - Basic Methods,” presented at Central European Seminar on Computer Graphics 2002, March 21, 2002

[6] Kristian Nymoen, Mari Romarheim Haugen, Alexander Refsum Jensenius,“MuMYO – Evaluating and Exploring the MYO Armband for Musical Interaction,” presented at International Conference On new Interfaces For Musical Expression, September 18,2015.

[7] Mithileysh Sathiyanarayanan, Tobias Mulling, Bushra Nazir. "Controlling a Robot Using a Wearable Device (MYO)," International Journal of Engineering Development and Research (IJEDR), Vol.3, Issue 3, pp. July 2015,