Intuitive and Adaptive Robotic Control Using Leap Motion

P. K. R. Vineeth, R. Baskaran, and Saswati Mukherjee

***Abstract*—There has been a global increase in arms and ammunition production and procurement and the use of these arms and ammunition recently has been coupled with various new technologies. The bomb squads of various countries have started using robots for defusing and detonating bombs but the use of the robots has been very inefficient due to use of joysticks. We propose a solution to this problem by making use of Leap motion controller instead of the traditional joystick. Along with this we have implemented inverse kinematics and tremor cancellation to ensure that the robot can exactly mirror the users hand movement. Therefore, with prior training and knowledge about bomb diffusion and detonation the use of robot with this will be much simpler and easier to control. The results shows us that the effective movement can be achieved through our system and it also provides us with a keen insight on errors that could occur sometimes.**

***Index Terms*—Arduino, inverse kinematics, leap motion, machine learning.**

I. Introduction

The growth in inventions and innovations over the recent years has paved the way for using different technologies for varied uses. Among the modern day technique used by many of the bomb squads in different countries, the most prominent way to detonate an enemy bomb is to place a detonator charge on top of the bomb to detonate it. This process can either be done manually or with robots controlled through joysticks. Manually placing the detonator charge is difficult and dangerous as it puts the life of the individual performing the task at stake. Therefore, the task is generally performed with robots controlled by joystick or keyboard. The person controlling the joystick has a lot of complications attached to it in terms of maneuvering the robot properly. We propose a long ranged bomb detonation, which is easier to control by making use of the Leap motion controller. The proposed Leap motion controller works with two infrared (IR) cameras and three Infrared LEDs as a depth sensor in a limited field of view (FOV) of 8 cubic feet.

Using the stereoscopic views from both cameras, the device can minimize errors from tools, fingers and hand features and is built on a unique mathematical model to maximize speed and precision.

We have proposed the use of Leap motion controller instead of the other existing technologies like Nintendo Wii and Microsoft Kinect for detecting the user's hand movement. While the Nintendo Wii and Microsoft Kinect have their own advantages of detecting the entire user body and multiple users, the precision levels in these systems are very low [1]-[4]. However, tasks like bomb diffusion and detonation must be executed in an environment where the precision level is very high. In the proposed Leap motion controller the hand tracking is very precise although it has a limited space to work with. The Leap Motion SDK comes with three different presets of tracking: Robust mode, Low resource mode and Auto orientation device mode.

Users generally are hesitant to use joysticks in controlling the robot because of the difficulties associated with it. To address the concerns, we propose a system, which senses the hand movement with the help of leap motion. Once the data is retrieved with the help of Leap SDK it is then sent to the Arduno Duemilanove via Bluetooth after manipulating the data based on the gestures and actions performed by the user [5], [6]. This is then sent to the robot as electric signals. Most of the existing techniques directly map the Leap SDK coordinate values to positioning of the robot [7], [8]. Our technique uses a novel approach that puts the user in the training phase where his tremor cancellation value is determined using machine learning algorithm and weighted average. Inverse kinematics equation helps improve accuracy in the proposed method.

Several machine learning methods have distinctive characteristics that make them good candidates for this empirical comparison. K-nearest neighbor (kNN) is a widely used instance-based learning method that scores high on simplicity. Recursion Tree (RT) is a popular inductive inference learning method that has a built-in feature selection capability. Bayesian Network tree (BNT) has the distinct advantage of uncovering causal relationships among attributes, hence providing added knowledge regarding the problem domain. Support Vector Machine (SVM) is supported by statistical learning theory and usually shows good generalization performance [9].

We have implemented tremor cancellation by making use of kNN (k-nearest neighbor) and Bresenham's line drawing algorithm. The movement of the user's hand during the training phase is classified into any of its closest possible values that are obtained using Bresenham's line drawing algorithm based on the initial and final coordinate value. The weighted average of the difference is taken as the tremor and

Manuscript received November 18, 2015; revised January 15, 2016. This work was supported in part by the Department of Computer Science and Department of Information Science and Technology of college of Engineering Guindy, Anna University, Chennai 600025 India.

P. K. R. Vineeth was with the College of Engineering Guindy, Anna University, Chennai 600025, India (e-mail: vineethpkr@gmail.com).

R.Baskaran is with the Computer Science Department, College of Engineering Guindy, Anna University, Chennai 600025 India (e-mail: baaski@cs.annauniv.edu).

Saswati Mukherjee is with the Department of the Information Science and Technology, College of Engineering Guindy, Anna University, Chennai 600025, India (e-mail: msaswati@auist.net).

DOI: 10.7763/IJCTE.2017.V9.1119

this value in turn is used for tremor cancellation when the user is performing the task.

Even though inverse kinematics is already implemented in different types of robots, we propose a novel approach for robotic control in the field of bomb detonation by combining inverse kinematics with tremor cancellation. In this field where position and accuracy are primarily important, Leap motion controller recognizes the users hand movements and gestures. In order to get the robot to mirror the exact movements of the user the shoulder joint movement is also needed. Therefore, we make use of inverse kinematics equation in order to determine the angle at which the shoulder joint should be moved to perform the action.

The remainder of our paper is organized as follows: Section II reviews the related work in the field of Robotic control, Leap motion, Degrees of freedom and the various other motion sensing. Section III describes in detail the kNN and inverse kinematics implemented in the system, while Section IV explains the construction of the robot and other implementation details. We present the Experimental Results in Section V. Section VI summarizes the findings of our experiments.

II. Related Work

In this section, we surveyed the various Machine learning algorithms and its application, Leap motion and its uses, Degrees of Freedom, Arduino Duemilanove and Inverse kinematics.

A. Machine Learning Algorithms and Its Applications

There are several machine learning algorithms. The most prominent ones are kNN, SVM, RT and BNT. They are used in wide areas of research topics such as cryptography, networks, human computer interaction (HCI) and robotics [10]-[12]. Each of the algorithms has its own advantage and disadvantage associated with it.

Typically, machine learning algorithms are mostly used for classification and regression. The simplest approach is kNN. The kNN is based on calculating the Euclidean distance between the closest n neighbors and classifying it to the closest group. It has its own disadvantage when working with skewed data set. In order to overcome this there are many kNN modifications like one based on weighted average as proposed by Tan and Sogbo [13].

B. Leap Motion Sensing and Its Uses and Degrees of Freedom

Leap motion is a motion sensing technology used in wide variety of fields. The basic functioning of leap motion is explained in detail by Frank Weichert and Daniel Bachmann [14]. Leap motion is made use of in everyday activities like in ambient assisted living where it contributes to the quality of life of the elderly and handicapped people and help them to maintain an independent lifestyle [15], converting sign language to text [2], tele-operation where it can be used by doctors to perform operations which require the use of robotic arm due to requirement of high degree of precision and also in the field of welding due to the same reason [8].

Degree of Freedom (DoF) is the number of independent ways in which the robot can move. The degrees of freedom are determined based on the use cases of the robot. There are many robots with varied degrees of freedom but the basic robot has four degrees of freedom like explained by Dr. A. G. Keskar Divesh Sawhney, Priyanka Khera in [16].

C. Arduino Duemilanove

There are a lot of micro controllers. The most prominent ones are Arduino and Raspberry Pi. Each has its own disadvantage and advantage associated with it. Arduino is mainly preferred when the micro controller is used mainly for sensor readings and manipulating the motors based on it. Since it is low powered it is easy to install and maintain as mentioned by Barrett, Steven F in [6]. Raspberry Pi on the other hand makes a slew of operations easier to manage, when one intends to connect to the Internet to read and write data, view media of any kind, or connect to an external display.

D. Inverse Kinematics

Inverse kinematics refers to the use of the kinematics equations of a robot to determine the joint parameters that provide a desired position of the end effectors. Goldenberg,

A. A. stated in his paper a generalized solution which is based on a combination of two nonlinear programming techniques and the forward recursion formulas, with the joint limitations of the robot being handled implicitly as simple boundary constraints [17]. So this can be applied to a robot with any number of degrees of freedom.

III. Proposed Work

We propose a long ranged bomb detonation, which is easier to control. The robot can be controlled, just by moving the hand of the user over the Leap motion controller. The robot mirrors the exact action of the user. We further propose to introduce inverse kinematics and tremor cancellation for better robotic arm movement, thereby providing a holistic solution.

A. k-Nearest Neighbor Classification

In this paper we have investigated and implemented the idea of k-NN classification. This is where you try to classify the unknown value based on *k* closest neighbors. k-Nearest Neighbor computes the similarity between the test instance and the training instance and considering the k top ranking nearest instances, finds out the category that is most similar. The similarity between the test and training instance is computed based on Euclidean distance formula.

$$D_{a,b} = \sqrt{}\ \Sigma_k(a_k - b_k)^2$$

where if *a* and *b* are the two points and *k* is the number of dimensions it has. In our case we consider three dimensions (*X*, *Y*, *Z*). The Euclidean distance treats each feature as equally important but this might not be the case always.

If *K* = 5, then in this case query instance (Fig. 1) x_q will be classified as negative since three of its nearest neighbors are classified as negative.

The k-NN is primarily used for classification and regression and it provides equal weight-age to all the neighboring points. However, this might not give accurate results for skewed data. That is, examples of a more frequent class tend to dominate the prediction of the new example, because they tend to be common among the *k* nearest neighbors due to their large number. In order to overcome this we have implemented weighted kNN, where we execute kNN algorithm and weighted average algorithm. The appropriate weight that is taken for each point is the inverse of the distance it is away from the training set. Thus, the more farther it is lesser would be the weight of it. The weighted average is calculated using the equation below and the weighted average algorithm implemented is shown below.

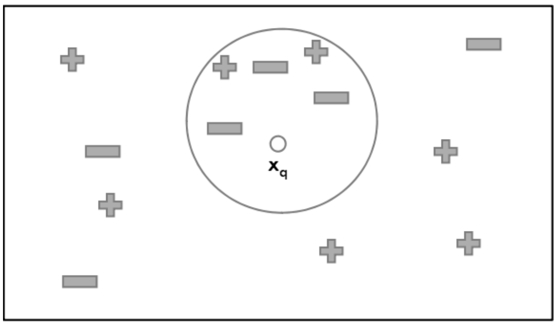


Fig. 1. k-Nearest neighbor.

$$\text{Weighted Average} = \sum_{n}^{i=1} (D_i * W_i)/(\sum_{n}^{i=1} (W_i))$$

Algorithm 1: **Weighted Average Algorithm**

```
procedure WEIGHTEDAVG (char axis, int TremorId, int TremorMax, int ResultSet [])
    weight [TremorMax]
    Count ← 0
    Denominator ← 0
    Avg ← 0
    WtAvg ← 0
    for Count < TremorMax do
        weight[Count++] ← 0
    end for
    for i < TremorId do
        diff ← ResultSet[axis]
        if diff! = 0 then
            Denominator ← Denominator+1
            weight[diff] ← weight[diff ]+1
        end if
        i ← i+1
    end for
    Counter ← 0
    for Counter < TremorMax do
        weight[Count++] ← 0
        if weight[Counter]! = 0 then
            Avg ← Avg + weight[Counter]*Counter
        end if
    end for
    WtAvg ← Avg/Denominator
    return WtAvg
end procedure
```

B. Inverse Kinematics

Specification of the movement of a robot so that its gripper achieves a desired task is known as motion planning. Inverse kinematics transforms the motion plan into joint actuator trajectories for the robot. Since the Leap motion controller only detects the movement of the hand, the angle at which the shoulder and the elbow joint should be moved is not known. Therefore in order to determine the exact angle by which the shoulder and the elbow are to move is determined using inverse kinematics.

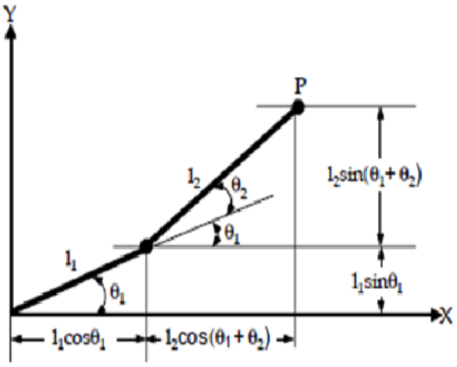


Fig. 2. Inverse kinematics.

The Fig. 2 shows us the mathematical formula diagrammatically. Where,

$P_{x,y}$ = End Effector's Coordinates

Θ_1 = Elbow Angle − Shoulder Angle

Θ_2 = Shoulder Angle

L_1 = Upper Arm Length

L_2 = Lower Arm Length

The inverse kinematics equation is shown below in Fig. 3.

$$\theta_1 = A\tan 2\left(\pm\sqrt{1-\left(\frac{p_x(l_1+l_2c\theta_2)+p_yl_2s\theta_2}{p_x^2+p_y^2}\right)^2}, \frac{p_x(l_1+l_2c\theta_2)+p_yl_2s\theta_2}{p_x^2+p_y^2}\right)$$

$$\theta_2 = A\tan 2\left(\pm\sqrt{1-\left(\frac{p_x^2+p_y^2-l_1^2-l_2^2}{2l_1l_2}\right)^2}, \frac{p_x^2+p_y^2-l_1^2-l_2^2}{2l_1l_2}\right)$$

Fig. 3. Inverse kinematics equation.

C. Robot Construction

The robot consists of the chassis and arm where the arm is mounted on top of the chassis. The movement of the chassis is powered by two DC motors for the left and the right wheel respectively. The arm consists of the shoulder joint and the elbow joint. Each of these joints movements is achieved with the help of servomotor. Mini servomotor is used for the shoulder joint and the micro servomotor is used for the elbow joint.

The grab and release function of the gripper is performed with the help of DC motor which is mounted at the edge of the elbow joint. These motors receive the input values through the Arduino board (ATMEGA 328), which in turn receives from the Bluetooth shield. The data given out by Leap SDK is re-factored and transmitted through Bluetooth.

IV. Implementation Details

In our proposed system we have made use of Arduino duemilanove, It is a micro controller that works with Atmega 328 and has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button [5], [6]. The micro controller is made use of in order to convert the data obtained into electric signals, which in turn makes the robot move. In this paper we aim to present a holistic solution for robotic control. Fig. 4 and Fig. 5 describe the design of the proposed system.

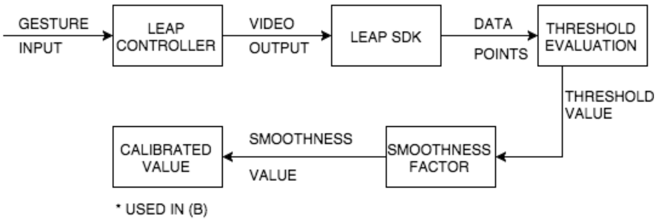


Fig. 4. Proposed system (A) – calibration.

The system has a training phase and an execution phase. In the training phase the user's hand is moved along the straight line and the training set is derived by drawing a straight line connecting the starting and the ending point of the user's hand movement using line drawing algorithm.

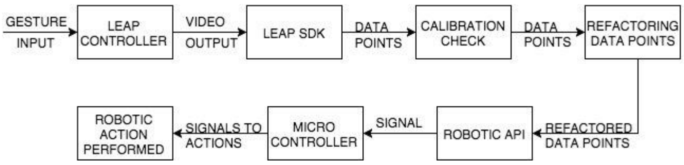
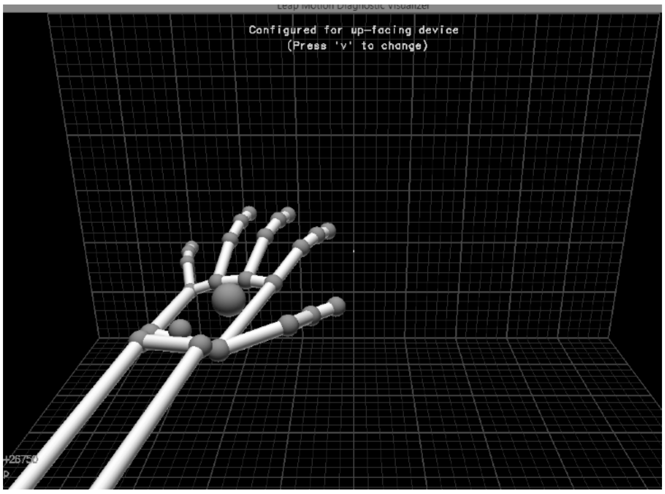


Fig. 5. Proposed system (B).

The Query set is the actual path that the user's hand moves in, while the user is asked to move his hand along a straight line. This process is done both horizontally and vertically. The difference between the Query set and the Training set is the Result set. Threshold value is the weighted average of the Training set values. Weighted average of Result set is taken so that sudden abrupt deviation in the user's hand movement does not lead to higher threshold value. This tremor rate is later used as the calibration check value.

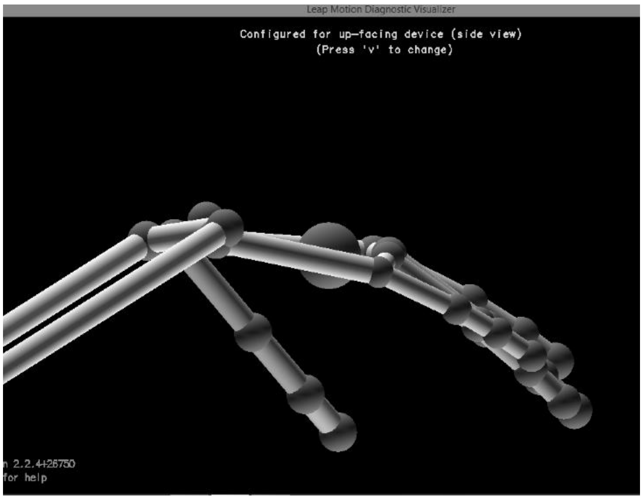
The user's hand movement over the Leap motion is gotten out of Leap SDK is gotten as coordinate values. The change in coordinate value is determined by comparing different frames of the hand movement captured by the Leap motion. Each frame has a list of tracking data such as hands, fingers, tools, recognized gestures and factors describing the overall motion of the scene. This allows Leap Motion to recognize hand features such as hand palm orientation, fingers length, width and orientation, hand opening and other non-hand features including tools and computer screen location. Furthermore, it incorporates in its Software Development Kit (SDK) the recognition of a few gestures including circle gesture, swipe and tap.

If the movement is observed to be greater than the calibration check value then the robot mirrors it. The data derived from the movement of the user's hand is passed on via Bluetooth to the Arduino. Bluetooth is a standard wire-replacement communications protocol primarily designed for low-power consumption, with a short range. We have adopted for Bluetooth over other wireless transmissions because it satisfies our use case sufficiently, that is to efficiently transfer data between two sources and to perform the same with low energy consumption. Though Bluetooth faces the connectivity and reliability issue, it is well equipped to handle the transfer of information between the master and the slave. The connectivity issue can be avoided by making use of better quality transceivers and the reliability issue can be overcome by encrypting the data that is sent over Bluetooth. The Arduino converts the data into high and low value so that the respective motors can be run. The movement of the robotic wheels is controlled through user's gestures and the robotic arm is controlled through user's hand movement as shown in Fig. 6 and Fig. 7. Inverse kinematics is implemented from the Arduino side where it provides movement to the shoulder and elbow joint based on the movement of the gripper.


Leap Motion Diagnostic Visualizer

Configured for up-facing device
(Press 'v' to change)

Fig. 6. Gesture for acceleration.


Leap Motion Diagnostic Visualizer

Configured for up-facing device (side view)
(Press 'v' to change)

n 2.2.4+26750
for help

Fig. 7. Gesture for drop.

V. Experimental Results

In this section, we demonstrate a thorough experimental evaluation of the proposed system. The whole experimental system is implemented using Java and python programming language, the micro controller Arduino is controlled using Arduino code based on objective C. It is implemented on a system running Linux OS with Intel i5 2.5GHz processor. The functioning of the system requires the use of Leap motion. To arrive at our results, we have made use of various inbuilt functions of Leap. These functions can be made use of by including the Leap class libraries.

A. Training Set and Query Set

In the training phase the user's hand is moved along the straight line. The training set is derived by, drawing a straight line connecting the starting and the ending point of the user's hand movement using line drawing algorithm. The query set is the actual path that the user's hand moves in. Result set is the difference between the training set and query set. It can be inferred from both the Fig. 8 and Fig. 9 that there is slight deviation from the path that the user tries to move in and that this deviation is unavoidable. Also since Leap motion detects the movement of the hand to great accuracy it is helpful in determining even the slightest movement. Based on the data derived through this we determine the tremor check value.

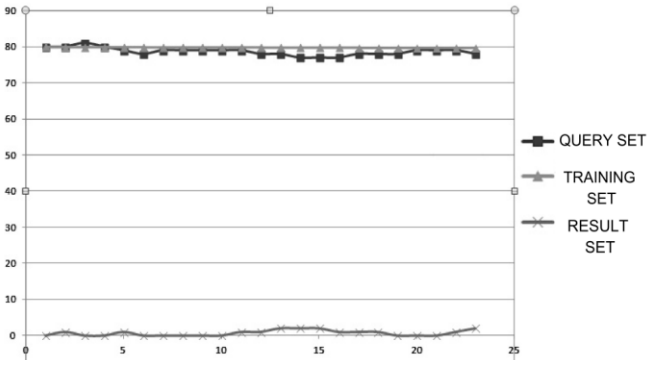


Fig. 8. Along the horizontal axis.

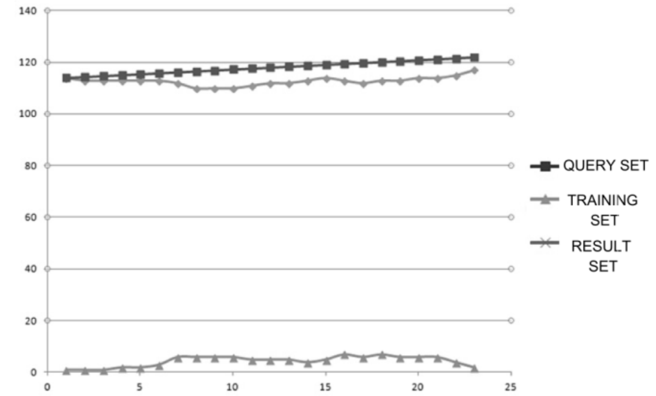


Fig. 9. Along the vertical axis.

Note that the graph values in Fig. 8 and Fig. 9 are derived after a testing the user's hand movement over a straight line for over 50 times. The difference is shown significantly due the high accuracy with which Leap motion operates.

B. Detection Rate

The Leap motion controller is relatively new in its domain. The skeletal tracking algorithm of the controller was developed very recently since the frames per second (fps) varies based on the light intensity on Leap motion controller. The data extracted from the controller is not 100 percent accurate. The data given out by the controller through Leap SDK is coordinate values. Leap has built-in software, which is useful in gesture detection based on the coordinate values. It can be inferred from the figure 10 that error rate in detection of gesture and movement is minimal but it still occurs due to inaccuracy in skeletal structure detection by Leap motion controller.

Note that the detection rate of leap motion of varied gestures is calculated after testing with the user's hand movement for over 100 trials.

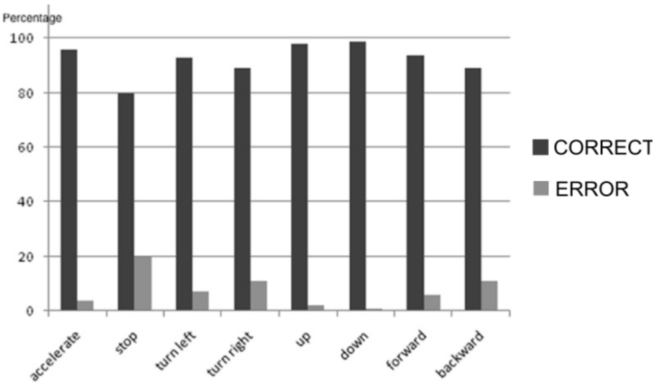


Fig. 10. Error rate.

C. False Positive and False Negative

The error rate in Leap motion can either be due to false positive or false negative cases. False positive is where the robot makes the movement but the user did not perform the gesture. False negative is where the user performs the gesture and the robot does not make the movement. It can be inferred from the graph (Fig. 11) that greater percentages of errors occur due to false negative than that of false positive. False positive error occurs when the data gets buffered due to high income of data, that is the micro controller does not immediately process it. False negative error occurs due to error in skeletal structure detection.

Since the Leap motion is being made use of in the field of bomb detonation here, even the slightest error is highly critical. Therefore, the errors in the detection as shown in Fig. 10 and Fig. 11 are primarily due to false negative, which could be avoided by improving on the Leap skeletal recognition. The false positive errors would most certainly be due to the choice of micro controller and the choice of wireless transmission that is made use of.

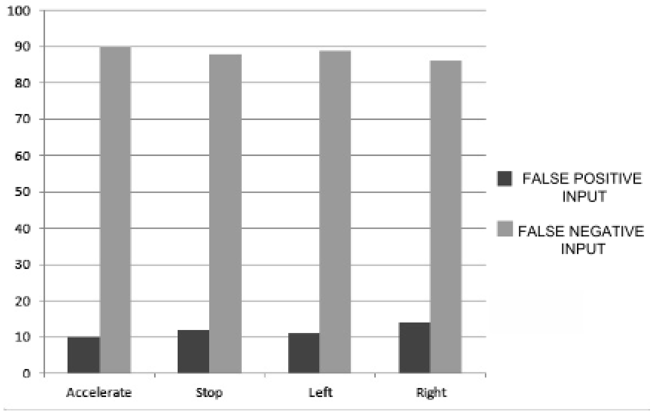


Fig. 11. False positive and false negative.

Therefore, by making use of an advanced micro controller and a more reliable wireless transmission this could be avoided.

Note that Fig. 11 shows the proportion by which the error rate is distributed among false positive and false negative cases.

VI. Conclusions

Manually detonating and triggering bombs is dangerous and very tedious. Therefore people prefer doing it from a safe distance but even with this approach they are constrained due to the use of joysticks and keyboards. These do not offer the same convenience as performing by hand. We have been able to overcome this difficulty by making use of Leap motion controller and a robot. Whereby, We are able to advantages of both the world, the user can safely perform the task from a distance and at the same time would be able to perform the task conveniently by making use of his hand. Also, there are other devices like Microsoft Kinect and Nintendo Wii, which could be instead of Leap motion controller, each having its own advantages and disadvantages. Since accuracy and latency is of primary importance we have adopted the Leap motion. The robot exactly mirrors the actions of the user, who is at a distance away from the bomb. This makes the task easier and safer to perform. In order to exactly mirror the user's hand movement we have implemented a novel approach by combining tremor cancellation and inverse kinematics. The tremor cancellation is uniquely implemented with the help of k-NN classifier and weighted average. Inverse kinematics is implemented using inverse kinematics equation. This is implemented in order to get the exact angle at which the gripper is to be moved to perform the task.

Leap motion controller is a recent technology and there are a lot of improvements and bug fixes that are yet come. It has a huge scope for implementation in various other fields. Since, we have made use of Leap motion controller in bomb detonation, field that relies heavily on accuracy. The errors that occurs currently, though minimum, it needs to be avoided. This could be done by making use of a more advanced micro controllers and a more reliable data transmission. Also, the errors could be avoided by improving the Leap skeletal recognition.

References

[1] S. Eduardo, O. J. Anderson *et al*., "A preliminary evaluation of the leap motion sensor as controller of new digital musical instruments," in *Proc. 14th IEEE SBCM - Brazilian Symposium on Computer Music*, 2013, pp. 202-205.

[2] P. L. Ellen, A. Jake, and C. Lewis, "The leap motion controller: A view on sign language," in *Proc. the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, 2013, pp. 175-178.

[3] M. Giulio, D. Fabio, and Z. Pietro, "Hand gesture recognition with leap motion and kinect devices," in *Proc. 21st IEEE International Conference on Image Processing*, 2014, pp. 1565-1569.

[4] G. Jozˇe, J. Grega, P. Matevzˇ, T. Sasˇo, and S. Jaka, "An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking," *Journal on Sensors*, vol. 14, no. 2, pp. 3702-3720, 2014.

[5] G. M. Alicia, "New media art, design, and the Arduino microcontroller: A malleable tool," Master's Thesis, Pratt Institute, New York, 2010.

[6] B. F. Steven, "Arduino microcontroller processing for everyone," Morgan & Claypool Publishers, vol. 8, no. 4, pp. 1-513, 2013.

[7] K. Futoshi and O. Kouki, and K. Fumio, "Robot-human handover based on position and posture of human hand," in *Proc. 15th International Symposium on Joint 7th International Conference on and Advanced Intelligent Systems*, 2014, pp. 918–921.

[8] Y. K. Liu, Y. M. Zhang *et al*., "Predictive control for robot arm teleoperation," in *Proc. 39th Annual Conference on Industrial Electronics Society*, 2013, pp. 3693–3698.

[9] R. Pramila, V. Eric *et al*., "An empirical study of machine learning techniques for affect recognition in human-robot interaction," *Pattern Analysis and Applications*, Springer, vol. 9, no. 1, pp. 58-69, 2006.

[10] R. L. Rivest, "Cryptography and machine learning," *Advances in Cryptology Asiacrypt'91*, pp. 427-439, 1993.

[11] Z. Sebastian, N. Thuy, and A. Grenville, "Automated traffic classification and application identification using machine learning," in *Proc. the 30th Conference on Local Computer Networks*, 2005, pp. 250-257.

[12] F. B. Mo ńica, R. L. Paulo, L. Nuno, and C. Gladys, "Machine learning algorithms applied to the classification of robotic soccer formations and opponent teams," in *Proc. Conference on Cybernetics and Intelligent Systems*, 2010, pp. 344-349.

[13] S. B. Tan, "Neighbor-weighted k-nearest neighbor for unbalanced text corpus," *Expert Systems with Applications*, vol. 28, no. 4, pp. 667-671, 2005.

[14] W. Frank, B. Daniel *et al*., "Analysis of the accuracy and robustness of the leap motion controller," Sensors, Multidisciplinary Digital Publishing Institute, vol. 13, no. 5, pp. 6380-6393, 2013.

[15] D. Bassily, C. Georgoulas, J. Guettler, T. Linner, and T. Bock, "Intuitive and adaptive robotic arm manipulation using the leap motion controller," in *Proc. 41st International Symposium on Robotics, ISR/Robotik*, 2014, pp. 1-7.

[16] D. Sawhney, P. Khera, and A. G. Keskar, "Robotic arm with four degrees of freedom," in *Proc. International Conference on Emerging Trends in Engineering Technology*, 2008, pp. 802-806.

[17] G. Andrew, B. Beno, and R. G. Fenton, "A complete generalized solution to the inverse kinematics of robots," *Journal of Robotics and Automation*, vol. 1, no. 1, pp. 14-20, 1985.

P. K. R. Vineeth was born on October 16, 1993 in Chennai, Tamil Nadu, India. He graduated in the field of computer science from College of Engineering Guindy, Anna University. He is currently working in a software firm in Chennai.

R. Baskaran did his undergraduate in the field of electrical and electronic engineering from Pondicherry University, India and later did his masters in computer science engineering from Vellore Engineering College, India.