A 3D Spectroscopic Vision-based Control Technique for Bilateral Systems

P. D. S. H. GUNAWARDANE[1,2], Nimali T. MEDAGEDARA[1]

1. *Dept. Mechanical Engineering, Faculty of Engineering Technology The Open University of Sri Lanka, Nawala, Sri Lanka;*

2. *Dept. Mechanical Engineering, Faculty of Applied Sciences The University of British Columbia, Vancouver, Canada;*

pdgun@ou.ac.lk, tmmed@ou.ac.lk

Abstract: Visual servoing is an active and popular area of research among roboticists. Eventhough visual servoing techniques enhance the perfomance, the associated systems still use traditional methods for their input control. Many research activities and applications have been carried out to implement effective and precise controlling of bilateral systems. This paper presents a 3D spresctroscope-based control technique for bilateral systems. The effectiveness of the available master side designs are evaluated against gesture-based techniques. Joystick control, Electromyography (EMG), Voice control, Haptic control, Exoskeleton control, Gesture and Brain Control Interface (BCI) are identified in the litreature as available bilateral inputs. In the present technnique, Leap Motion Controller (LMC) has been introduced (LMC) to extract the human hand gestures and their parameters. Then these parameters are convereted into respective joint sapce angles using the presented mathematical model. The mathematical models for fingertip mapping, inverse kinematics, dynamics and trajectory generation are implemented and studied. Wolfman Mathematica 10 and MATLAB simulation framework are used to validate the mathematical models, simulations and developed control algorithms. The developed system has sucesfully imitated the fingertip motion. In particular, the system has been able to imitate the figretip motion with a deviation of 6.7 % in X axis, 5.5% in Y axis and 7.9% in Z axis with respect to the expected position.

Key words: Bilateral systems; fingertip imitation; robot manipulator; MATLAB robotics tool box; robot simulation; kinematic model; robot control; leap motion controller; hand gestures.

1 Introduction

Today's mega factories and their Industrial 4.0 developments require sophisticated techniques of human machine cooperation. These factors have motivated designers to think new perspectives of human machine co-existence. The traditional techniques of control have many disadvantages in real-time telepresence operations. Among different types of control, the use of human hand gesture control is more convenient for operators when they are connected to machines in real-time. Artificial cognitive communication capabilities also have been introduced to establish real-time communication between humans and computers [1]-[2].

Bilateral robotic systems are used to perform complex tasks and manipulate the interaction in sensible remote environments. Typically, it consists a human-operated master controller connected through a communication channel to a remotely operated slave electro-mechanical system. In industrial applications, bilateral systems have many advantages. These systems are being implemented for various applications. Main factors have to be considered in designing these systems, particularly for the effectiveness in Human Machine Interface (HMI) and real-time operation.

2 MasterSide Control Inputs

In the literature, mainly joystick control, haptic control, exoskeleton control, hand gesture control, brain control interface (BCI), electromyography (EMG) control, eye gaze control, and human body gesture control have been used as master side control

inputs. The present section discusses their construction and techniques used to detect the control inputs.

2.1 Joystick Control Device

Joystick is a popular input device and has been used widely to give control inputs to different systems. This technique can capture movements and forces and convert them into control signals.

For example, joysticks have been used as the control input tool in environmental control systems, communication aids, and most popularly in gaming PCs [3]. A limitation of this technique is that the motion control requires real-time mapping with the physical system. The degrees of freedom in the joystick control inputs are quite limited [4]. Inflexible joysticks use strain-gauge or piezo-electric sensor technologies to generate X and Y axes signals proportional to the applied forces. Movement- sensitive and force-sensitive joysticks are used to implement movement - force continuum [4]. There are many situations where joystick control is inapplicable due to the complexity of the slave mechanism [5].

2.2 Haptic Control Device

Haptics is a technique used for the master side control. This device captures force measurements of the operator who closes the control loop. This technique is widely used in surgical robotics due to its high accuracy, quality and tele-operability in operation [6]. The main shortcoming in this technique is the aggregated time delay in the closed loop system when haptic hand controllers are located at distance from the robot [6].

2.3 Exoskeleton Control Device

Exoskeleton techniques are used as a mechanism to capture the finger movements of an operator using direct motion capture. Actuated exoskeletons are usually utilized in robotics that guide neuro rehabilitation such as stroke patients, paraplegics, and children with cerebral palsy. The inverse version of this technique is used to register the joint movements. The same technique is used to measure the patient's actions and feedback, in rehabilitation therapy programs [7].

2.4 Brain-Computer Interface

Brain-Computer Interfaces (BCI) are used in controlling external devices by measuring the brain signals. The motor imagery (MI) is a BCI model, which enables the users to perform control actions [8]. Electroencephalogram (EEG) signals are captured using an electrode-based technique to implement the interface.

2.5 Electromyography Signal (EMG)

Electromyography signal (EMG) is also used as a control technique in master side designs. Electrical activities in the human muscle and tissues are recorded using electrodes to implement the control signals. A well measured and calibrated EMG signal has been used to perform precise control in prosthetic hands related systems [9].

2.6 Eye Gaze Tracking Control Device

Eye gaze tracking techniques are used in many applications. This technique is used in devices where clinching the point in space at which a person is focusing (to detect visual attention) has to be measured. A primary application for eye gaze tracking is Human-Computer Interaction (HCI), providing impaired users the ability to control the pointing and/or selection of components with eye movement in a User Interface, which are typically done by mouse and keyboard. To be used in a computer interface as an input device, a robust eye gaze tracking system should fulfill the following requirements: be highly precise and accurate; be non-intrusive; be immune to occlusions like blinks; be immune to changes in head displacement; be tolerant of changes in lighting; and, and be able to track continuously and in real-time [10]. This technique is highly challenging to implement in real-time control because of the low accuracy.

2.7 Human Gesture Control Device (Body and Hand)

Several gesture databases and human gesture control devices have been constructed to observe or analyze the characteristics of human motion and sub-

stantiate the developed algorithms and its applications [11]. Human gesture control techniques are discussed in Section 3.

The main aspects of these designs, for implementing bilateral systems, are safety and convenience (to improve the workability of the operator for a long time). Most of the techniques mentioned above have their own disadvantages in implementing these criteria. The use of hand gestures is convenient, flexible and robust in implementation [5]. The present paper presents the development of a 3D spectroscopic vision-based control approach. The proposed algorithm tracks and registers the fingertip movements of hand gestures using LMC.

3 Gesture Control Devices

Various techniques have been used to detect gestures. This section explores the most popular techniques that are available to track human gestures.

The first attempt of implementing Data Glove (DG) devices (Fig.1) is dated back to the 1980's. In general, DGs are used to measure the finger movements and positions using a flexible sensor. Typically, these measurements are taken in combination with a hand movement detection sensor that is capable of tracking six degrees of freedom [12].

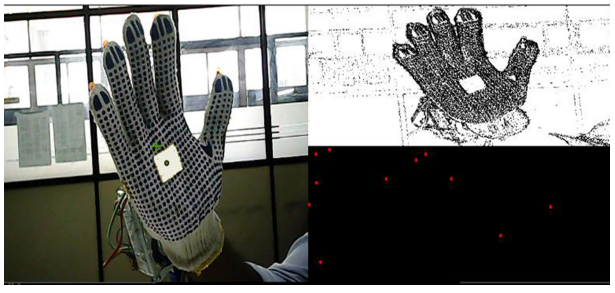


Fig. 1 Tracking of hand gestures using a data glove [13].

The positions of the hand and the signs can be identified witha camera system that uses a depth sensing technique (Fig. 2). The device has a Red-Green-Blue RGB camera, depth sensor, and multi-array microphone running proprietary software. This sensor can provide full-body 3D motion capture, facial recognition and voice recognition capabilities.

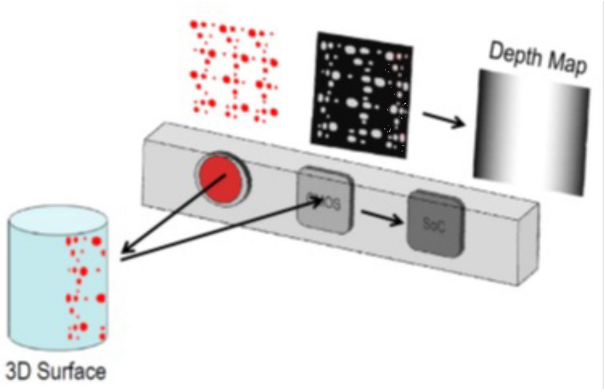


Fig. 2 3D surface detection method using light patterns [14].

A popular gesture sensing device is the Kinect sensor (Fig. 3). This is a full body scale 3D motion detection sensor system that uses RGB cameras, depth sensor and multi-array microphone running proprietary software.

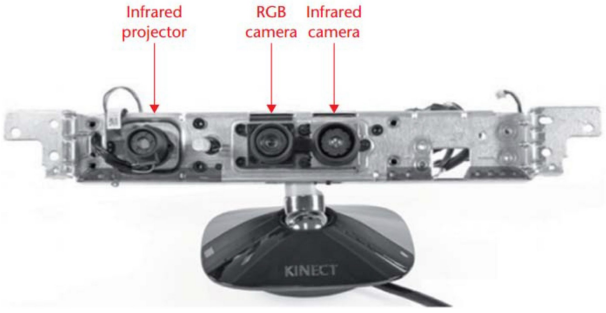


Fig. 3 Kinect device [15].

Different types of Wii Remotes are another type of special gesture detecting control device (Fig. 4). The sensor bar powers up the LED lights and transmits data to either the console or the Wii controllers. The LED lights are used as a reference point for the Wii remote, and the Wii remote picks up the light emitted from the sensor bar and uses it to determine the movement of the controller [12].

LMC is also used for hand and finger movement and position detection (Fig.5). The system measures the fingers movements separately, so the computer can be controlled with slow and tiny gestures as well[12]. Two high-speed, monochrome infrared cameras are built into the two sides of the controller with three infrared LEDs. The cameras and LEDs

cover a hemispherical area with a diameter of approximately one meter. The Leap Motion software processes the images and measures the exact position and location of the fingers in space.

Fig. 4 Wii remote sensor.

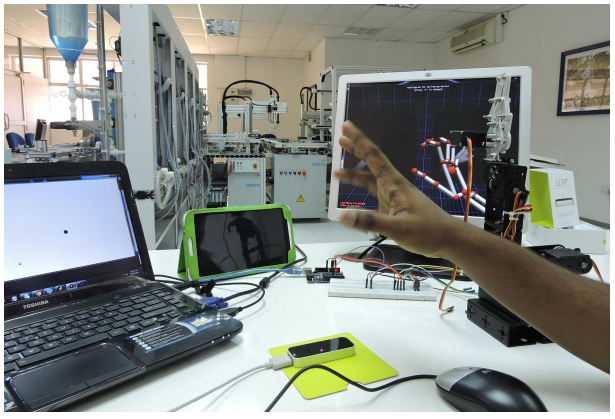


Fig. 5 The Leap Motion Controller system.

4 Analysis of LMC

In this section, the effectiveness of the master designs in bilateral systems is evaluated and compared with LMC. The effectiveness depends on many factors. Based on the literature, it has implemented a criterion to evaluate the effectiveness of each technique. Contactless operation, controllability, ease of making control decisions, response time, reliability and safety are considered as main factors that define effectiveness. These factors are associated with subfactors, which are used as binary decisions to make the evaluations. Fig.6 shows a graphical representation of the effectiveness of the master side design techniques. It is seen that the highest effectiveness is in hand gesture vision attentive systems and the lowest is in EMG. Other systems are averagely varied in their effectiveness values. These effectiveness values are application dependent.

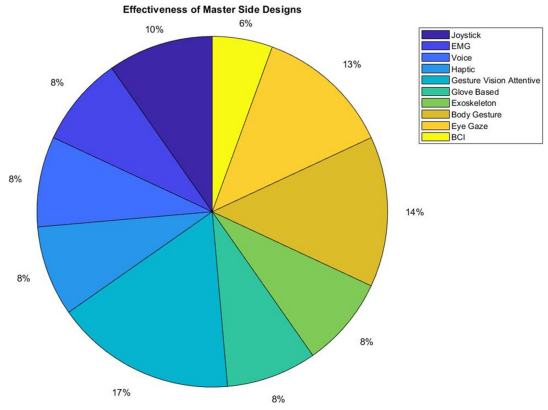


Fig. 6 Graphical representation of the effectiveness of master design methodologies.

For example, EMG is not suitable for use in epidemic environments but suitable in rehabilitation type applications. Therefore, hand gestures are quite appropriate for use in bilateral systems. Hence, a fingertip imitating robot is implemented using LMC as the gesture tracking device. The main reasons for choosing LMC as the 3D sensor for hand gesture detection are low cost (with respect to other available 3D sensor modules), range, special designed for hand gesture tracking, availability of free SDK (Software Development Kit) and libraries, MATLAB compatibility (matleap integrating package), application novelty for manipulator robot integration, high accuracy, lightweight, and being a portable and plug and play device.

Hand gesture controller designis based on stereo vision technology. Stereo vision is defined as the extraction of 3D information from digital images. It is artificial stereopsis, which is followed by two cameras to obtain the depth and 3D structural information. Literature survey [16] reveals that within the past few years, 3D data acquisition has been vastly increased in potential industrial applications. The accuracy and robustness of 3D sensors have increased while the price has dropped. Mainly 3D sensors are used for object tracking, motion analysis, 3D scene recon-

struction, and gesture-based user interfaces.

Optical 3D sensors can be categorized according to its design mechanism. Structured light analyzes the deformation of a known pattern in an unknown surface, to determine the 3D shape. The Microsoft Kinect sensor [17] shown in Fig. 7.e is a well-known example. It is made using a conventional RGB camera, a sensor with an infra-red (IR) camera and projector, an amplifier, and a built-in engine. This engine allows the base of the Kinect sensor to tilt [18]. The ToF (Time of Flight) 3D cameras use the well-known time of flight principle, which is based on phase shift. MESA Swissranger 4000 [19] shown in Fig. 7.b, PMD (Photonic Mixer Device) [20] Camboard Pico shown in Fig. 3.c, and FHOTONIC B70 shown in Fig. 7.d are well known examples. Stereo vision cameras are based on two optical 2D cameras with known extrinsic parameters. Determination of the depth in the scene is based on searching for correspondence points in both 2D images. Bumble bee 2 sensor [21] shown in Fig. 7.a and LMC [22] shown in Fig. 7.f are famous examples. In LMC, stereoscopic vision technology is integrated with structured light technology.

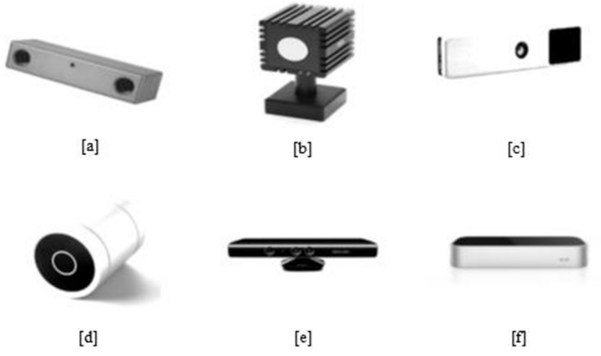


Fig. 7 Available types of 3D sensors
[a] Bumble bee 2 sensor [b] Mesa Swissranger 4000 [c] PMD Camboard PICO [d] FHOTONIC B70 [e] Microsoft Kinect sensor [f] Leap Motion Controller.

5 Modelling of the System

The Fingertip Imitating Robot Manipulator is a bilateral system and its main design constitutes a master side design, a slave side design, and communication in between. The master side is the LMC, which detects the fingertip, and the slave side is the Robot Manipulator Mechanism. The modelling and design are carried out individually for the two systems, and then the two systems are integrated.

LMC in conjunction with its API (Application Programming Interface) drive positions in the Cartesian space such predefined objects as the finger tips and pen tips. This is designed as an USB (Universal Serial Bus) peripheral device. Two monochromatic IR cameras and three infrared LEDs (Light Emitting Diodes) are assembled as shown in Fig. 8a. The LEDs generate a 3D pattern of dots of IR light, and the cameras are assembled in such a fashion to generate 300 frames per second of reflected data. This data is transmitted to the V2 Tracking software, and the 2D frames generated by the two cameras are analyzed further to generate 3D positions. As shown in the Fig. 10, LMC consists of a right-handed Cartesian coordinate system. The origin lies on the top of the LMC. X and Z axes lie on the horizontal plane, and Y axis is vertical. Table 2 gives the measurement parameters of the LMC with their units. The measurements can vary according to the external conditions.

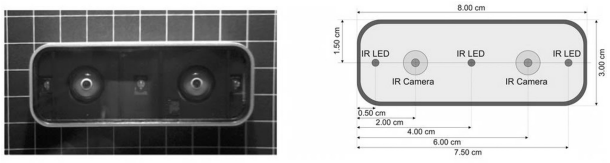


Fig. 8 (a). Contraction of the Leap Motion Controller.

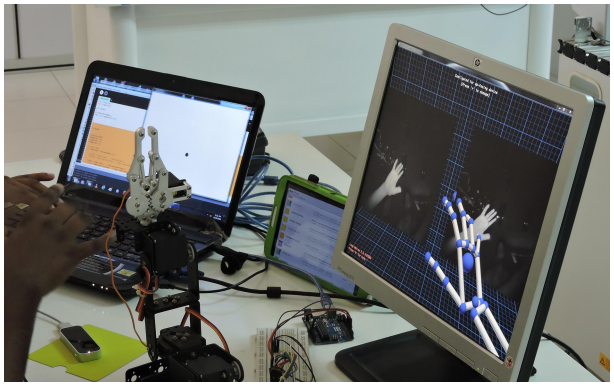


Fig. 8 (b). Detecting the hand skelton by LMC.

Table 2 Measurement information of the LMC.

Parameter	Units
Distance	Millimeters
Time	Microseconds
Speed	Millimeters/second
Angle	Radians

Leap Motion API consists of several classes developed to enable tracking. Hand Class API Model is used to detect the hand, as shown in the Fig. 9.a, Finger Class API Model is used to detect the fingers, as shown in Fig. 9.b, and Tool Class API Model is used to detect the hand, as shown in the Fig. 9.c. Matleap, MATLAB interface for LMC is used to intercommunicate with MATLAB software Robotics Tool Box 9.10. As shown in Fig. 10, the frame {0} is assigned to the LMC and the frame {1} is assigned to the fingertip. LMC detects the position, orientation, and the velocity of the fingertip. LMC provides the X, Y, Z, α, β, γ, V_x, V_y, and V_z values with respect to the coordinate frame {0}. To relate these two frames, the homogeneous transformation matrix between the frames {0} and {1} is used. The rotation is given by α, β, γ, which are represented as $\theta_1, \theta_2, \theta_3$, and the translations are X, Y and Z. Then the composite transformation matrix is derived as,

$$ {}^{0}_{1}T = \begin{bmatrix} C_2\,C_3 & S_1S_2\,S_3 - C_1S_3 & C_1S_2\,C_3 + S_1S_3 & X \\ C_2\,S_3 & S_1S_2\,S_3 + C_1C_3 & C_1S_2\,S_3 - S_1C_3 & Y \\ -S_2 & S_1C_2 & C_1\,C_2 & Z \\ 0 & 0 & 0 \quad 1 & \end{bmatrix} \tag{1} $$

where, $C_i = \cos\theta_i$ and $S_i = \sin\theta_i$.

Eqn. (1) gives the inverse kinematics of the robot mechanism, which determines the respective joint angles. Fig. 11 shows the mechanism of the robot. It consists of three R (Revolute) type joints, three links, and an end-effector. Each joint has an actuator, which is capable of performing the rotation action (by using DC servo motors).

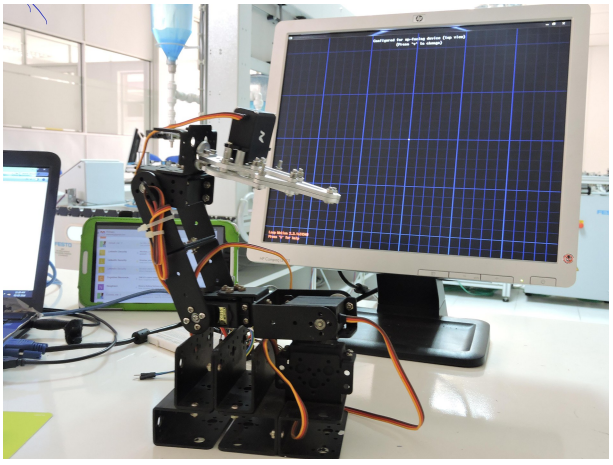


Fig. 9 Fingertip imitating robot manipulator.

The main task is to track the fingertip with the aid of the LMC, and imitate it through the controlled end-effector motion (Fig.9). A mathematical model between robot base and the end-effector has been developed using joint angles. The coordinate frames have been assigned to the system accordingly. The world coordinate frame is considered as the base of the robot, and the end-effector is mapped with respect to the base frame.

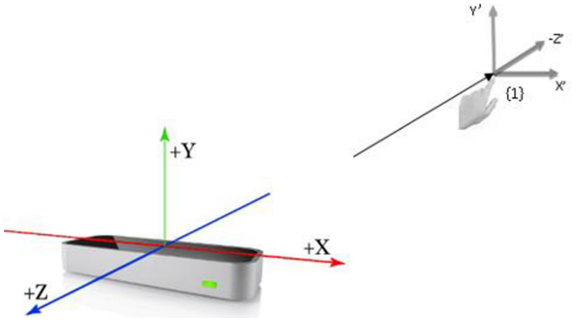


Fig. 10 Frame assignment of the hand gesture system.

Fig.11. a shows the coordinate frames that are assigned to the robot framework/mechanism. The frames are assigned using the Denavit- Hartenberg (DH) notation. Fig. 11.b shows the frame assignment of the robot and its home position. Using this information and the frame assignment in Fig. 12 (DH notation is used), the DH table (Table 3) has been developed.

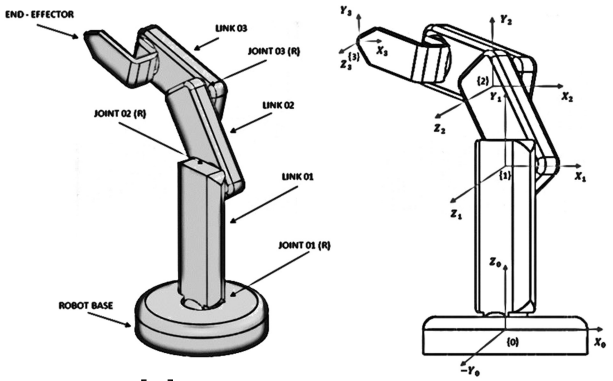


Fig. 11 [a] Configuration of the serial link robot mechanism. [b] Frame assignment in the ioint space.

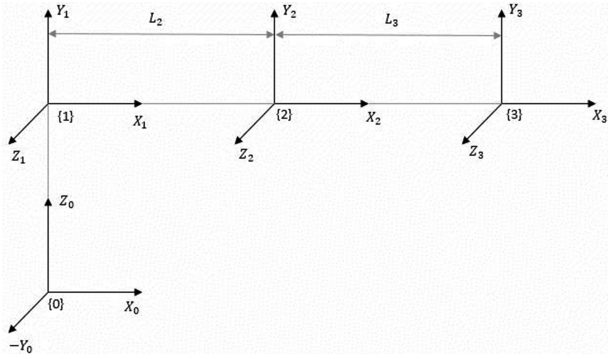


Fig. 12 Frame assignment considering the home position.

Table 3 DH table for the 3D robot.

Link_i	θ_i	d_i	a_i	α_i	$C\theta_i$	$S\theta_i$	$C\alpha_i$	$S\alpha_i$
1	θ_1	0	0	$\pi/2$	$C\theta_1$	$S\theta_1$	0	1
2	θ_2	0	l_2	0	$C\theta_2$	$S\theta_2$	1	0
3	θ_3	0	l_3	0	$C\theta_3$	$S\theta_3$	1	0

The final forward kinematics equation for the robot mechanism is obtained as,

$$ {}^{0}_{1}T = \begin{bmatrix} C_2 C_{23} & -C_1 S_{23} & S_1 & C_1[L_3 C_{23} + L_2 C_2] \\ C_2 S_3 & -S_1 S_{23} & -C_1 & S_2[L_3 C_{23} + L_2 C_2] \\ -S_2 & C_{23} & 0 & L_3 S_{23} + L_2 S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2) $$

where $C_i = \cos\theta_i$, $S_i = \sin\theta_i$, $C_{ij} = \cos(\theta_i + \theta_j)$, and $S_{ij} = \sin(\theta_i + \theta_j)$. Equation (2) gives the forward kinematics solution, which relates frame $\{0\}$ (base frame) to the end-effector frame $\{3\}$. For known joint angles, Eqn. (2) gives the end-effector configuration (position and orientation). To find the required joint angles for a known end-effector configuration, geometrical solutions are derived. Consider the end- effector configuration,

$$ T = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \quad (3) $$

Using the forward kinematic model developed above, the analytical solutions of the joint angles are obtained as,

$$ \theta_1 = Atan2(a_{24}, a_{14}) \quad (4) $$

$$ \theta_2 = Atan2\left[(-L_3 a_{31} + a_{34}),\ \pm\sqrt{(-L_3 a_{11} + a_{14})^2 + (-L_3 a_{21} + a_{24})^2}\right] \quad (5) $$

$$ \theta_3 = Atan2(a_{31}, a_{32}) - \theta_2 \quad (6) $$

Therefore, fora known end-effector configuration (as in Eqn. (3)), Eqn. (4), Eqn. (5) and Eqn. (6) can be used to determine the joint space values (θ_1, θ_2 and θ_3). Also, LMC provides the fingertip velocity in terms of V_x, V_y, and V_z. In this manner, a mathematical relation is derived using these parameters. The Jacobian matrix for the 3DOF robot is,

$$ J = \begin{bmatrix} -S_1[L_3 C_{23} + L_2 C_2] & -C_1[L_3 C_{23} + L_2 C_2] & -L_3 C_1 S_{23} \\ C_1[L_3 C_{23} + L_2 C_2] & -S_1[L_3 C_{23} + L_2 C_2] & -L_3 S_1 S_{23} \\ 0 & L_3 C_{23} + L_2 C_2 & L_3 C_{23} \\ 0 & -C_1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (7) $$

The inverse of the Jacobian is as follows:

$$[J^{-1}] = \frac{adj(J)}{det(J)} = \frac{F}{R} \quad (8)$$

Hence, the joint space velocities are as follows:

$$\dot{q} = J^{-1}(q)\, V_e \quad (9)$$

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \left[\frac{F}{R}\right] \cdot \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \quad (10)$$

Eqn. (9) and Eqn. (10) give the joint space velocity results for a known Cartesian space velocity. Therefore, using the detected fingertip position, orientation and velocities, it is possible to obtain the joint angles values and joint space velocities of the robot that would imitate the fingertip by the end-effector of the robot. The trajectory generation is implemented by considering the 5th prder polynomial equation as given by

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (11)$$

Velocity and acceleration may be expressed as

$$\dot{\theta}(t) = a_1 + a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \quad (12)$$

$$\ddot{\theta}(t) = 2\, a_2 t^2 + 6\, a_3 t + 12 a_4 t^2 + 20 a_5 t^3 \quad (13)$$

These are the first and the second derivatives of the trajectory polynomial. Table 4 presents the start and the end data for the trajectory solution.

Table 4 Start and end point parameters.

	Time	θ	$\dot{\theta}$	$\ddot{\theta}$
Start	t_s	θ_s	$\dot{\theta}_s$	$\ddot{\theta}_s$
End	t_f	θ_f	$\dot{\theta}_f$	$\ddot{\theta}_f$

Then the trajectory is obtainedby solving the matrix in the equation,

$$\begin{bmatrix} \theta_s \\ \theta_f \\ \dot{\theta}_s \\ \dot{\theta}_f \\ \ddot{\theta}_s \\ \ddot{\theta}_f \end{bmatrix} = \begin{bmatrix} 1 & t_s & t_s^2 & t_s^3 & t_s^4 & t_s^5 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_s & 3t_s^2 & 4t_s^3 & 5t_s^4 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^2 & 5t_f^2 \\ 0 & 0 & 2 & 6t_s & 12t_s^2 & 20t_s^3 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \quad (14)$$

Fig. 13 shows the behavior of the MATLAB generated trajectory for a single case, and Fig. 14 shows the Mathematica generated angle, velocity and acceleration plots. In this example, it has started and finished the points between 0 and 1, which has 50 steps. The velocity and acceleration are taken to be zero at the two extreme points.

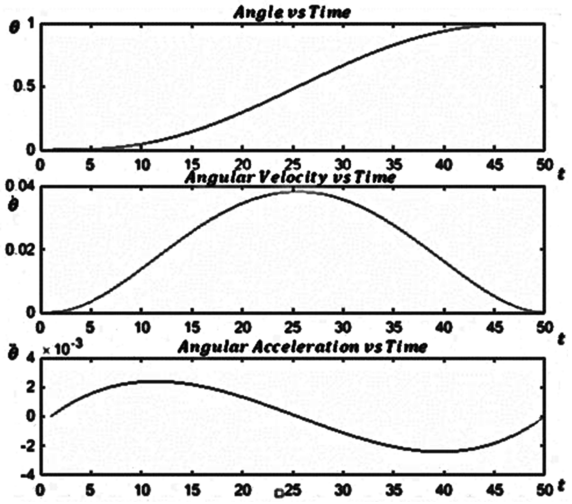


Fig. 13 MATLAB generated trajectory plot for a single joint.

The Dynamic model of the robot is shown in

$$Q = M(q)\, \ddot{q} + C(q, \dot{q}) + G(q) \quad (15)$$

where Q = *JointTorqueMatrix*, $M(q)$ = *Inertia Matrix*, $C(q,\dot{q})$ = *CorriolisMatrix*, and $G(q)$ = *GravityVector*. The joint's angular displacement, angular velocity and angular acceleration are as in Eqn. (11)-(13). Using the Lagrange-Euler formalization Equation (15) can be rewritten, and the joint torques are given by

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} C_{11} \\ C_{12} \\ C_{13} \end{bmatrix} + \begin{bmatrix} G_{11} \\ G_{12} \\ G_{13} \end{bmatrix} \quad (16)$$

The Inertia Matrix is given by

$$M_{11} = \frac{1}{2} m_1 l_1^2 + \frac{1}{3} m_2 l_2^2 \cos^2(\theta_2) + \frac{1}{3} m_3 l_3^2 \cos^2(\theta_2 + \theta_3) + m_2 l_2^2 \cos^2(\theta_2) + m_3 l_2 l_3 \cos(\theta_2 + \theta_3) \cos(\theta_2) \quad (17)$$

$$M_{11} = 0 \quad (18)$$

$$M_{21} = 0 \tag{19}$$

$$M_{13} = 0 \tag{20}$$

$$M_{12} = \frac{1}{3} m_3 l_3^2 \tag{21}$$

$$M_{22} = \frac{1}{3} m_2 l_2^2 + \frac{1}{3} m_3 l_3^2 + m_2 l_2^2 + m_3 l_2 l_3 \cos(\theta_3) \tag{22}$$

$$M_{23} = \frac{1}{3} m_3 l_3^2 + m_2 l_2^2 + \frac{1}{3} m_3 l_2 l_3 \cos(\theta_3) \tag{23}$$

$$M_{31} = 0 \tag{24}$$

$$M_{32} = \frac{1}{2} m_3 l_3^2 + m_3 l_2^2 + \frac{1}{3} m_3 l_2 l_3 \cos(\theta_3) \tag{25}$$

The Coriolis matrix is given by

$$C_{11} = \begin{bmatrix} -\frac{4}{3} m_2 l_2^2 \sin(2\theta_2) - \frac{1}{3} m_3 l_3^2 \sin 2(\theta_2 + \theta_3) - m_3 l_2 l_3 \sin(2\theta_1 + \theta_3)\, \theta_2\, \theta_1 \\ + \left[-\frac{1}{3} m_3 l_3^2 \sin 2(\theta_2 + \theta_3) - m_3 l_2 l_3 \cos(\theta_2) \sin(\theta_2 + \theta_3) \right] \theta_3\, \theta_1 \end{bmatrix} \tag{26}$$

$$C_{12} = \left[-m_3 l_2 l_3 \sin(\theta_3) \right] \dot{\theta}_2\, \dot{\theta}_3 + \left[-\frac{1}{2} m_3 l_2 l_3 \sin(\theta_3) \right] \dot{\theta}_3^2 +$$

$$\left[\frac{1}{6} m_2 l_2^2 \sin 2(\theta_2) + \frac{1}{6} m_3 l_3^2 \sin 2(\theta_2 + \theta_3) + \frac{1}{2} m_3 l_2^2 \sin(2\theta_2) + \frac{1}{2} m_3 l_2 l_3 \sin(2\theta_2 + \theta_3) \right] \dot{\theta}_1^2 \tag{27}$$

$$C_{13} = \left[\frac{1}{2} m_2 l_2 l_3 \sin(\theta_3) \right] \dot{\theta}_2^2 + \left[\frac{1}{6} m_3 l_3^2 \sin 2(\theta_1 + \theta_3) + \frac{1}{2} m_2 l_2 l_3 \cos(\theta_2) \sin(\theta_2 + \theta_3) \right] \dot{\theta}_1^2 \tag{28}$$

The Gravity vector is given by

$$G_{11} = 0 \tag{29}$$

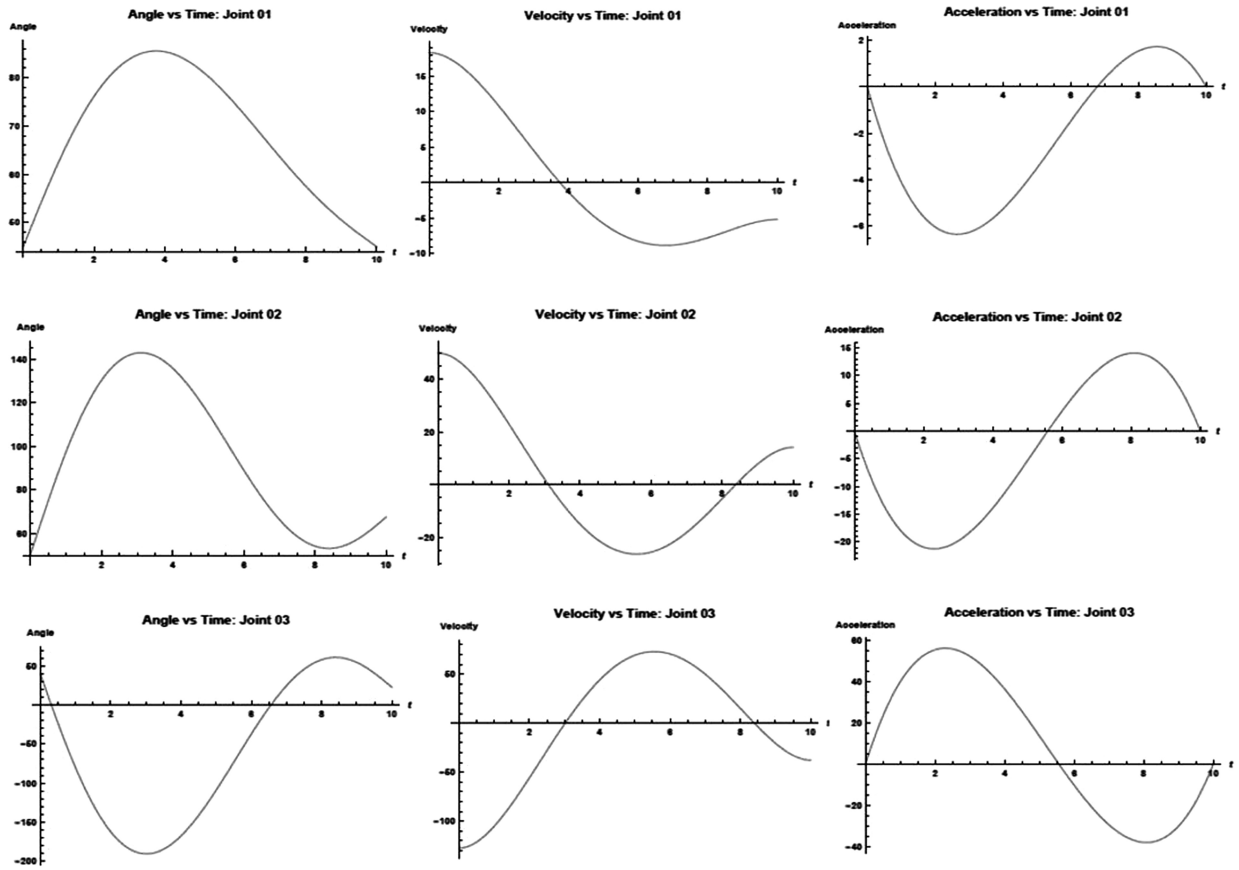


Fig. 14 Mathematica generated angle, velocity and acceleration plots.

$$G_{21} = \frac{1}{2} m_2 g\, l_2 \cos(\theta_2) + \frac{1}{2} m_3 g\, l_3 \cos(\theta_2 + \theta_3) + m_3 g\, l_2 \cos(\theta_2) \quad (30)$$

$$G_{31} = \frac{1}{2} m_3 g\, l_3 \cos(\theta_2 + \theta_3) \quad (31)$$

6 Simulation Results

The mathematical model for the 3DOF robot manipulator was developed in Mathematica 10, which generated the angle, velocity and the acceleration plots for a known case, as shown in Fig. 15. The Cartesian displacement of the fingertip was taken as 1 unit along x, y, z axes, and -π/6, π/12, -π/12 rotations in respective axes with 15% reduction in velocity in all three axes. Initial and final joint accelerations were taken as zero. Then the joint torques were calculated. This model was tested with real fingertip data. Robotics toolbox for MATLAB (RTB 9.10 by Peter Corke) was used to develop the 3DOF robot model for the simulation. Link objects and the SerialLink constructer were used to develop the model. LMC detects the position, orientation and the velocity parameters of the fingertip, and then MATLAB matleap library was used to read the parameters. From the position and orientation data, the composite transformation matrix was generated and used with the inverse kinematics model to find the joint angle values. Also, Cartesian velocity was used to find the required joint space velocities. Based on these values, robot trajectory was generated, and then the simulation was able to successfully imitate the fingertip motion. Fig. 15 and Fig. 16 show the 3DOF robot model used for the simulation.

Fig-17 shows the block diagram of the MATLAB simulation. The design and development of a robot can become a challenge due to possible singularities. The program was designed by considering both interior and boundary singularities. As shown in the pseudo code, if the LMC extracted data is situated outside or on the boundary, then those values will be excluded to avoid boundary singularities. To avoid interior singularities, after performing inverse kinematics, values are checked and the abnormal ones are excluded. Then with the aid of this algorithm, proposed system was tested.

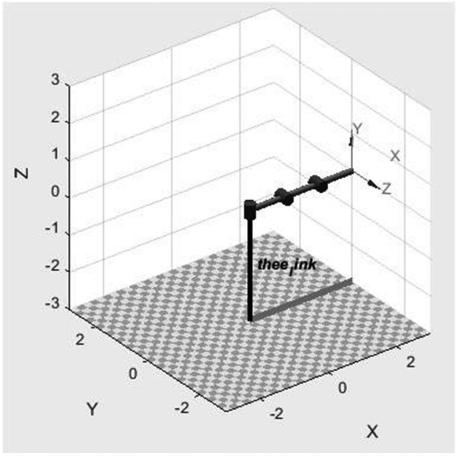


Fig. 15 3DOF robot model.

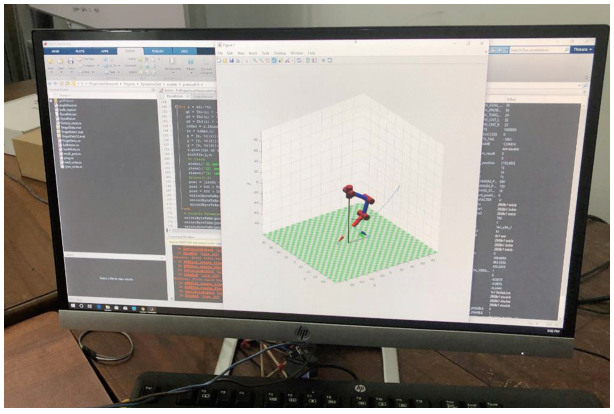


Fig. 16 3DOF robot model used for the simulation.

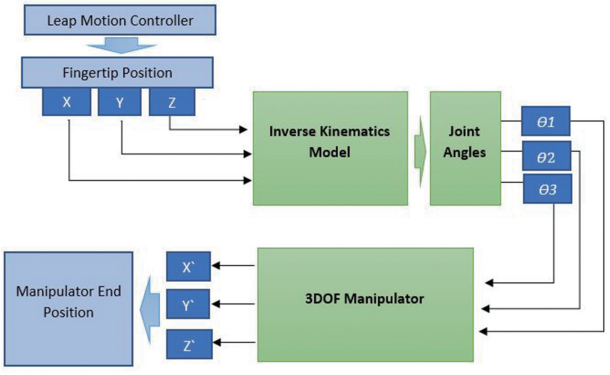
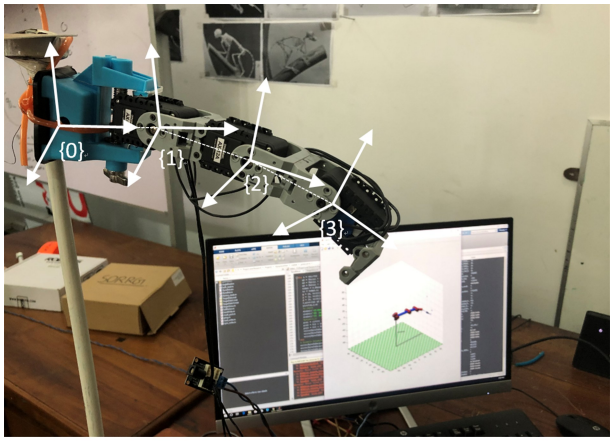


Fig. 17 Functional diagram of the gesture-controlled robot

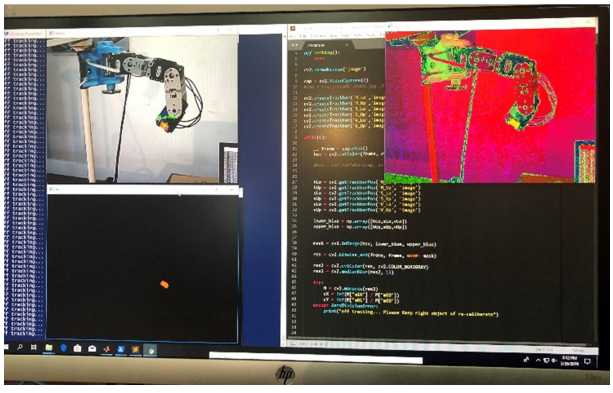
7 Results and Conclusion

According to the developed robot model, using

the program, a test simulation was performed and compared with the experimental results (Fig.18).



(a)



(b)

Fig. 18 Experimental setup.

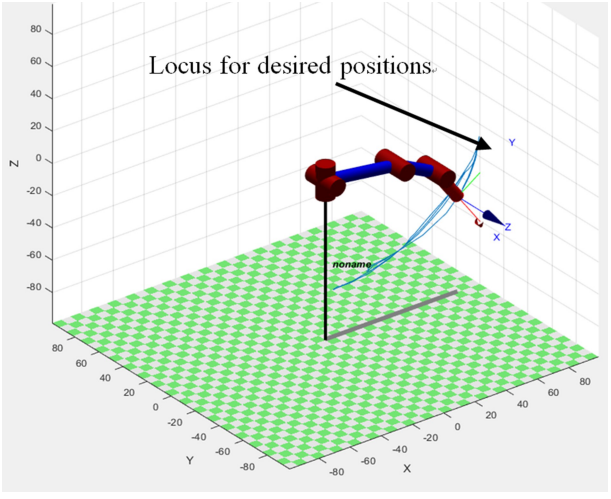


Fig. 19 Results of simulation.

The position of the fingertip was tracked by the Leap Motion Controller and thex, y and z position data were extracted. These x, y, z position data were provided to the derived inverse kinematic model and the corresponding joint angles $\ominus 1$, $\ominus 2$ and $\ominus 3$ in each joint of the 3DOF Manipulator were determined.

The calculated joint angles were provided to the 3DOF manipulator simulation, and the position of the end-point of the manipulator was compared with the tracked fingertip position, by the Leap Motion Controller, as shown in Fig.19 and Fig.20.

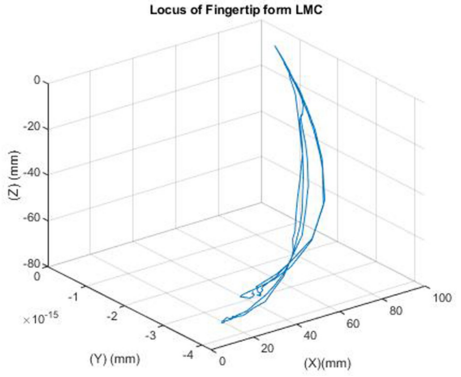


Fig. 20 Locus of fingertip for LMC.

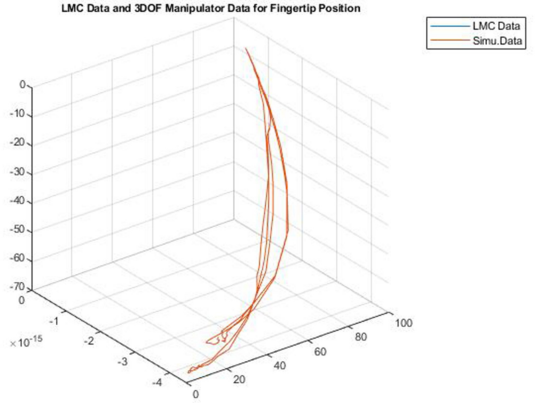


Fig. 21 LMC and robot manipulator simulated data for fingertip position.

Fingertip data that was tracked from Leap Motion controller, and the same data that was obtained from the 3DOF manipulator simulation, were compared (Fig.21). The deviation of the results between the Leap Motion data and the simulated data was found to be quite small.

Fig. 18 shows the experimental setup and in this hardware implementation end-point of the 3DOF ma-

nipulator was tracked through computer vision using HSV color model-based object tracking techniques (Fig22). Fingertip position data that was obtained from the Leap Motion Controller and the End-point data of the 3DOF manipulator that was obtained by the computer vision system were compared to determine the deviation between the theoretical (LMC) data and the experimental data (Fig. 23).

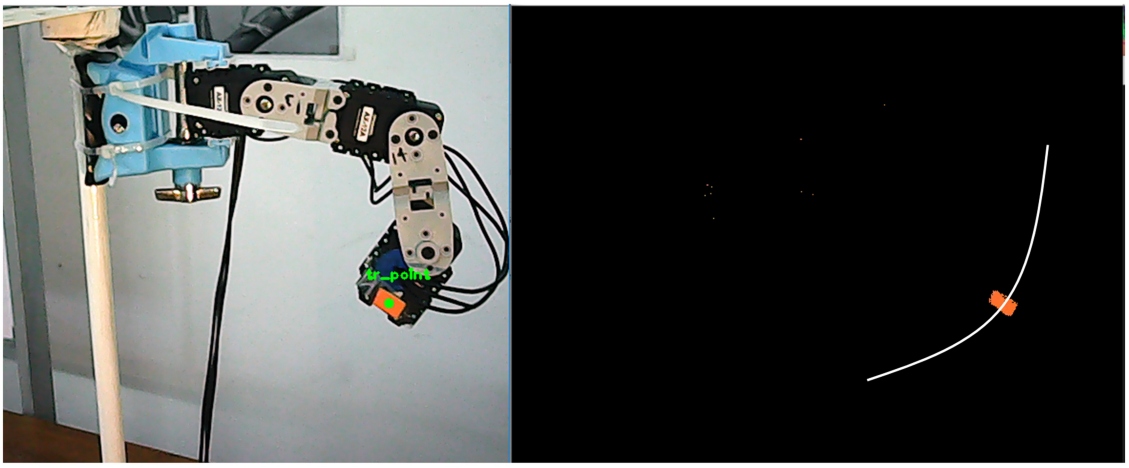


Fig. 22 Tracking the position of the end-effector.

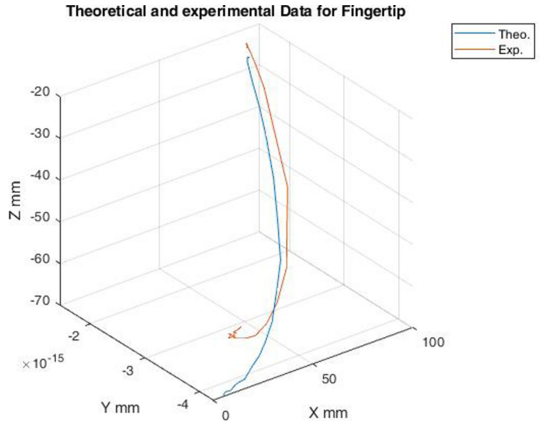


Fig. 23 Behavior of the LMC and robot manipulator in real time.

Table 5 Percentage deviation of the position.

Axis	Deviation (%)
X	6.7
Y	5.5
Z	7.9

The variation of the real time data of the LMC and the Robot manipulator was plotted along the X, Y and Z directions as shown in Fig.23. The percentage of deviation of the robot manipulator relative to the LMC was calculated along the all three directions, and the results are given in Table 5.

According to the results, there is no variation between the position of the LMC and the robot manipulator simulated data for the fingertip position. But it was found that in real time data, the deviation of the position between the robot manipulator with respect to the LMC was 6.7% along the X direction, 5.5% along the Y direction, and 7.9% along the Z direction. This variation may be due to various errors in the system.

References

[1] Baranyi P., Csapó Á.,(2012). Definition and Synergies of Cognitive Info communications. Acta Polytechnica Hungarica, Vol. 9, No. 1, pp. 67-83.

[2] Baranyi, P., Csapo, A., & Sallai, G. (2015). Synergies Between CogInfoCom and Other Fields. Cognitive Infocommunications (CogInfoCom), 39-55.

[3] Cooper, R. (1995). Intelligent control of power wheelchairs. IEEE Engineering in Medicine and Biology Magazine, 14(4), 423-431.

[4] Spaeth, D., & Cooper, R. Variable compliance joystick for control interface research. Proceedings of the First Joint BMES/EMBS Conference. 1999 IEEE Engineering in Medicine and Biology 21st Annual Conference and the1999 Annual Fall Meeting of the Biomedical Engineering Society (Cat. No.99CH37015).

[5] Posada-Gomez, R., Sanchez-Medel, L. H., Hernan-

dez, G. A., Martinez-Sibaja, A., Aguilar-Laserre, A., & Leija-Salas, L. (2007). A Hands Gesture System Of Control For An Intelligent Wheelchair. 2007 4th International Conference on Electrical and Electronics Engineering.

[6] Richert, D., & Macnab, C. (2009). Direct adaptive force feedback for haptic control with time delay. 2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH).

[7] Kooij, H. V. D., Veneman, J., & Ekkelenkamp, R. (2006). Design of a compliantly actuated exo-skeleton for an impedance controlled gait trainer robot. 2006 International Conference of the IEEE Engineering in Medicine and Biology Society.

[8] An, H.-S., Kim, J.-W., & Lee, S.-W. (2016). Design of an asynchronous brain-computer interface for control of a virtual Avatar. 2016 4th International Winter Conference on Brain-Computer Interface (BCI).

[9] Ming L., Wang Z. (2001) The Study Advances and Prospects of Processing Surface EMG Signal in Prosthesis Control, Chinese Journal of Medical Instrumentation. Vol.25, No 3.

[10] Hughes, J., Rhodes, S., & Dunne, B. E. (2017). Eye gaze detection system for impaired user GUI control. 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS).

[11] Hwang, B.-W., Kim, S., & Lee, S.-W. A Full-Body Gesture Database for Automatic Gesture Recognition. 7th International Conference on Automatic Face and Gesture Recognition (FGR06).

[12] Sziladi, G., Ujbanyi, T., & Katona, J. (2016). Cost-effective hand gesture computer control interface. 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom).

[13] Gunawardane, P. D. S. H., & Medagedara, N. T. (2017). Comparison of hand gesture inputs of leap motion controller & data glove in to a soft finger. 2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS).

[14] Centre for Sports Engineering Research, Sheffield Hallam University. Depth Biomechanics. [online] Available at: http://www.depthbiomechanics.co.uk/? p=100.

[15] Zhang, Z. (2012). Microsoft Kinect Sensor and Its Effect. IEEE Multimedia, 19(2), 4-10.

[16] Gunawardane, P. D. S. H., Medagedara, N. T., & Madhusanka, B. G. D. A. (2015). Studying the competitiveness of master designs in bilateral systems for an epidemic environment. 2015 8th International Conference on Ubi-Media Computing (UMEDIA).

[17] Kinect for Windows Sensor Components and Specifications (2014) Previous versions of Microsoft products, services and technologies [online] Available at: https://msdn. microsoft. com/en-us/library/jj131033.aspx. [Accessed 7 Jan. 2015]

[18] Difilippo, N. M., & Jouaneh, M. K. (2015). Characterization of Different Microsoft Kinect Sensor Models. IEEE Sensors Journal, 15(8), 4554-4564.

[19] SR400 (Imaging, MESA),(2014) [online] Available at: http://www. mesa-imaging. ch/products/sr4000/. [Accessed 7 Jan. 2015]

[20] Reference Design BriefCamBoardpicoflexx, [online] Available at: http://static6.arrow.com/aropdfconversion/309ae5444d793b84a974e5325122494b5ac27ab8/irs1145c.pdf. [Accessed March 7, 2015]

[21] Bumblebee XB3 1.3 MP Color FireWire 1394b 3.8mm (Sony ICX445), (2014) [online] Available at: http://www. ptgrey. com/bumblebee-xb3-stereo-vision-13-mp-color-firewire-1394b-38mm-sony-icx445-camera. [Accessed March 7, 2015]

[22] Technology, (2014) Leap Motion Product, [online] Available at: https://www. leapmotion. com/product [Accessed March 7, 2015]

Authors' Biographies

P. D. S. H. Gunawardane graduated from The Open University of Sri Lanka with a B.Tech. Honors in Engineering, specializing in Mechatronics Engineering. He is currently a postgraduate student in the Microelectromechanical Systems (MEMS) Laboratory and the Industrial Automation Laboratory (IAL), Department of Mechanical Engineering, The University of British Columbia, Vancouver, Canada. He is a Lecturer and a Co-Investigator in the Soft Robotics Research Group of The Open University of Sri Lanka.

E-mail: pdgun@ou.ac.lk, hiroshan@ieee.org

T. M. D. N. T. Medagedara graduated with a B.Sc.Eng.(Hons) degree in Mechanical Engineering from the University of Peradeniya, Sri Lanka, and completed her postgraduate studies, specializing in Materials and Manufacturing Engineering, at Sheffield Hallam University, United Kingdom. She is a Senior Lecturer and a Principal Investigator in the Soft Robotics Research Group of The Open University of Sri Lanka.

E-mail: tmmed@ou.ac.lk