Journal of Physics: Conference
Series

Real-time Motion Control of Robot Using Kinect

To cite this article: R Dilip *et al* 2021 *J. Phys.: Conf. Ser.* **2115** 012010

Real-time Motion Control of Robot Using Kinect

Dilip R[1], Dharani J[2], K J Jai Viknesh[2], Lohith D S[2], Yogesh Prakash[2]

[1]Associate Professor, Department of Electrical and Electronics Engineering, Global Academy of Technology

[2]Department of Mechatronics, Acharya Institute of Technology

[*]**dilipr@gat.ac.in.**

Abstract – The goal of Industry 4.0 is to maximize efficiency and reduce human efforts. Artificial Intelligence, Cloud Computing, Robotics, Mobile Technologies and many other technological advances generate Industry 4.0. Humanoid robots resemble humans and their primary goal is to perform any task programmed by the user. Our goal is to create a portable and safer system that can be used in any dynamic environment to control the robot and avoid any sort of complexity. With the rapid technological advances, there is no need to worry about unemployment since it requires a technical expert in the field to control the robot. This paper discusses how this control system processes gesture data along with a brief discussion of certain useful applications.

Keywords- LabVIEW, Kinect, Joint angles

1. INTRODUCTION

Over the last few decades, various intriguing technologies have been developed in the field of HCI. One of the reasons for developing such technologies is to safeguard humans. But such technologies remove the need of human employees in certain fields. So we developed a HCI (Human Computer Interface) control system where a humanoid robot can be remotely controlled by a human [1]. In dynamic workspaces, the usage of manipulator robots controlled by a shared human-computer interaction opens up new possibilities for the execution of difficult tasks. The behavior of many manipulator robots is comparable to that of human arms. When the robotic arm has an equivalent degree of freedom to the human arm, which is largely based on skeletal detection, motion planning is simplified thanks to gesture recognition using the Kinect sensor. This project is a control system that includes a small-scale 17 Degrees of Freedom(DOF) Humanoid Robot chassis which can mimic human gestures with the help of the Xbox 360 Kinect Sensor. The Kinect captures the gesture data and will enable the control system to mimic human gestures in real-time.

2. LITERATURE SURVEY

We have done a comprehensive study of the latest technological trends and efficient systems. Our study includes the current knowledge, findings, as well as theoretical and methodological contributions for development of modern robot control systems. Industries still use human labour to move objects or products around. To reduce human labour, an obstacle avoidance robotic arm that can be controlled using NI myRIO (National Instruments Reconfigurable Input Output) programmed in LabVIEW (Laboratory Virtual Instrument Engineering Workbench) [2]. An algorithm for 6 DOF robotic arms controlled by Arduino [3] was developed using the LabVIEW programming language along with the LabVIEW interface for Arduino (LIFA) toolkit. This is a cheaper solution compared to the previous study since Arduino micro-controllers are cheaper than NI myRIO. A study [4] presents a humanoid robot being controlled by capturing gestures of a human through a Microsoft Xbox 360 Kinect Sensor. A medical service robot [5] consisting of an arm mounted on a wheeled platform that uses a Kinect Sensor to capture hand gestures and is programmed using LabVIEW.

3. HARDWARE AND SOFTWARE

- **Xbox 360 Kinect** - Array of sensor used to capture human skeletal points.
- **Arduino Mega 2560** - It is used for interfacing LabVIEW to the robot.

- **DC Servo motor metal gear MG996r** - Fitted on to the robot joints.
- **National Instruments LabVIEW** - Graphical programming environment.
- **Microsoft Kinect SDK (Software Development Kit)** - Supporting files for interfacing Kinect to the computer.
- **MakerHub LINX** – API's (Application Programming Interface) for interfacing Arduino with LabVIEW.

4. METHODOLOGY

The block diagram below explains the methodology steps involved to achieve the working control system.

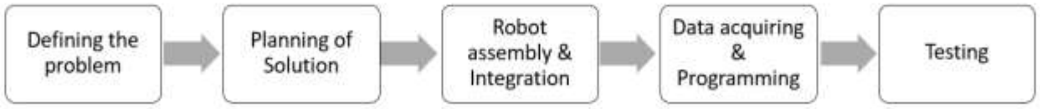


Figure 1 Methodology

The current model employs a maneuvering system with torque control motors attached to an exoskeleton which makes mounting the suit on a human hard. While it increases the sensitivity, it also makes it harder for the user to wear and is heavy to maintain as well, so in a dynamic environment the user cannot do any other tasks which can reduce the productivity of that environment.

Figure 2 Toyota T-HR3 humanoid robot.

To ensure minimum usage of equipment, we planned to implement our idea using an Xbox 360 Kinect sensor which comprises 2 depth sensors and an RGB camera. The depth sensors ensure accurate motion capture and the recorded motion will be processed by a microcontroller and then pushed to the robot which will mimic the same.

A multiple DOF robot body made of aluminum and metal gear servo motor is used to implement the gesture based control of the robot's movements. The Kinect SDK provides an API, using which the HMI (Human Machine Interface) is developed, the LabVIEW uses this API to extract the skeletal data of the human user and this skeletal tracking data is used to calculate the joint angle required to move the links in the desired way [10]. The LabVIEW does this operation using supporting toolkits and libraries called "KINESTHESIA".

The working requires multiple PWM (Pulse Width Modulation) for proper functioning of the servo motors, so an Arduino Mega 2560 is used which provides 15 PWM pins. The Arduino is interfaced with the LabVIEW software using a toolkit called "DILIGENT LINX ". The toolkit provides a firmware to Arduino which enables it to work synchronously with the LabVIEW environment. The HMI is developed in the LabVIEW and the data or user movements are obtained and processed using the Kinect and joint angles are calculated and converted to PWM signals to transmit to servo motors through Arduino.

Figure 3: Assembled multiple DOF Humanoid Robot

The joint angle data is calculated using the skeletal data Acquired from the Kinect [8]. The program is designed to create a human machine interface, which is intermediate between the user and the robot. It calculates the angle values and converts it into a PWM signal for servo motor movements to get the desired outputs. The concepts of calculating the joint angles are explained as follows.

4.1. Camera Coordinates (X, Y, Z)

The Kinect's camera coordinates look for 3D points of joints in space using the Kinect's infrared sensor. These are the coordinates to utilize in 3D projects when determining joint locations. Color and depth coordinates are treated differently than camera space coordinates [12].

From the SDK:

The 3D frame of reference used by Kinect is referred to as camera space. The following is the definition of the frame of reference:

On Kinect, the origin (x=0, y=0, z=0) is at the centre of the IR sensor.

From the sensor's POV (Point of View), X increases to the sensor's left.

The direction of Y's growth is determined by the sensor's tilt.

Z expands in the direction of the sensor.

1 unit = 1 meter

In camera space, the coordinates in the Kinect space is measured in meters. The coordinates (x, y) are often positive or negative, as they extend in both directions from the sensor. The z coordinate will always be positive, because it grows out from the sensor.

The Kinect has an eight-meter depth range, but its skeleton tracking range is 0.5 to 4.5 metres, and it has problems locating skeletons closer than 1.5 metres because to the camera's sector of view. As a result, the camera's Z value will normally be between 1.5 and 4.5.

Because 0 is at the sensor's center, and joints are frequently tracked to the sensor's left (positive) or right (negative), the x coordinate value is frequently negative or positive (negative). Camera X's range is determined by the joint's distance from the camera, however it can span up to 6 metres.

The Y coordinate values also can be negative or a positive value: The Y coordinate is positive when it's above the sensor, and negative when it's below the sensor. The camera Y range will depend upon the space from the camera, but can sense about five meters tall.

4.2. Depth Coordinates (X, Y)

The depth coordinates (x, y) are the coordinates of the picture taken by the depth camera at the joint. The following is an excerpt from the Software Development Kit [7]:

The phrase "Depth space" is used to denote a 2D position on a depth image. Consider this a pixel's row and column location, where x represents the column and y represents the row. As a result, x=0, y=0 corresponds to the image's highest left corner, whereas x=511, y=423 (width-1, height-1) relates to the image's bottom right corner.

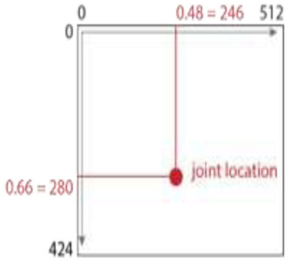


Figure 4: Joint Location

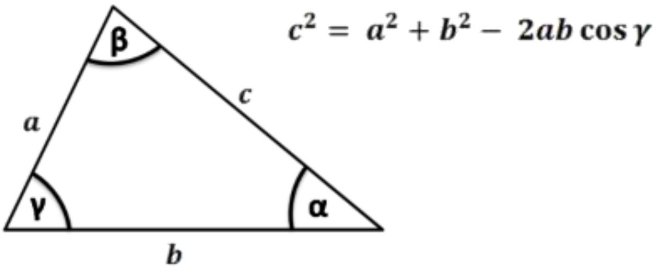
4.3. Inverse Kinematics

The study of inverse kinematics has become more popular as robot kinematics and dynamics have become more complex. As a result, it's not unexpected that some of the initial inverse kinematics solutions were limited to specific robot designs.

Although this approach claims to find a solution if one exists, it does so after multiple iterations. Because they alter all the joint variables simultaneously on a path that would move the end effector towards the goal, methods employing the Jacobian Inverse matrix are generally not cost effective and can be numerically unstable [6]. Furthermore, the Jacobian Matrix can be single, making finding an inverse for the solution impossible. As a result, the concept of cosines law is applied to calculate the joint angle using Kinect coordinates [11].

4.4. COSINES LAW

The law of cosines in trigonometric calculations relates the length of a triangle's sides or edges to the cosine of one of its angles [6]. The law of cosines simplifies the Pythagorean theorem, which is only true for right-angled triangles. When any two sides and their enclosed angles are already known, the cosines law can be used to compute the third side of a triangle, as well as to compute the angles of a triangle if all three sides are known.



$$c^2 = a^2 + b^2 - 2ab\cos\gamma$$

Figure 5: Cosines Law formula

The length of ***a***,***b***,***c*** is extracted from the input coordinates with the vector calculation as shown in the below figure.

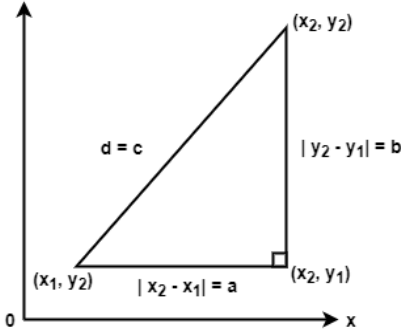


Figure 6: Vector based length calculation

Take the square root of both sides of the Pythagorean Theorem to find the length ***c***

$$C^2=a^2+b^2 \rightarrow c=\sqrt{a^2+b^2} \tag{1}$$

It follows that the distance formula is given as

$$d^2=(x2-x1)^2+(y2-y1)^2 \tag{2}$$

$$d=\sqrt{(x2-x1)^2+(y2-y1)^2} \tag{3}$$

4.5. Joint angle calculation

The joint angle angulation required to send signal to the motor for desired motion is as shown in the below figures [8].

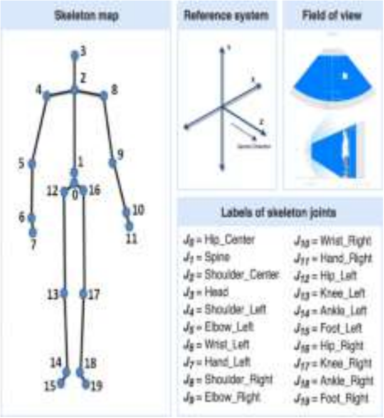


Figure 7: Pictorial representation of skeletal data and joints

The joint angles for respected joints are as described in the following table, and a few samples are briefed for clear understanding [5].

1.Knee angle left

- Angle between the vector defined by the hip left and
- Knee left joints and by the knee left and ankle left joints.

2. Knee angle right

- Angle between the vector defined by the hip right and
- Knee right joints and by the knee right and ankle right joints

Table 1: List of Joint angles and the respected joints used for calculation

Joint Name and Number	Joint 1	Joint 2	Joint 3
Right shoulder(j8)	Shoulder centre(j2)	Right Shoulder(j8)	Right Elbow(j9)
Right Elbow(j9)	Right Shoulder(j8)	Right Elbow(j9)	Right Wrist(j10)
Left Shoulder(j4)	Shoulder centre(j2)	Left Shoulder(j4)	Left Elbow(j5)
Left Elbow(j5)	Left Shoulder(j4)	Left Elbow(j5)	Left Wrist(j6)
Right hip(j16)	Hip centre(j0)	Right hip(j16)	Right Knee(j17)
Right Knee(j17)	Right hip(j16)	Right Knee(j17)	Right Ankle(j18)
Left Hip(j12)	Hip centre(j0)	Left hip(j12)	Left Knee(j13)
Left knee(j13)	Left Hip(j12)	Left Knee(j13)	Left ankle(j14)

The following figure represents the concepts of joint angle calculation using the coordinates obtained from Kinect and these values are used to obtain the desired motion of the robot joint [9].

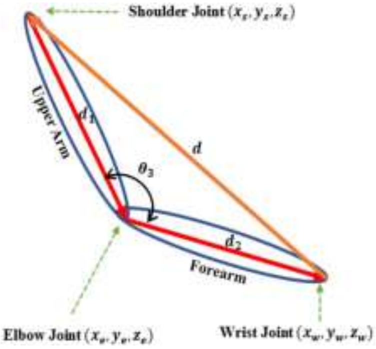


Figure 8: Pictorial representation of joint angle calculation of elbow.

A sample of the program developed in the LabVIEW environment is as shown in the figure below.

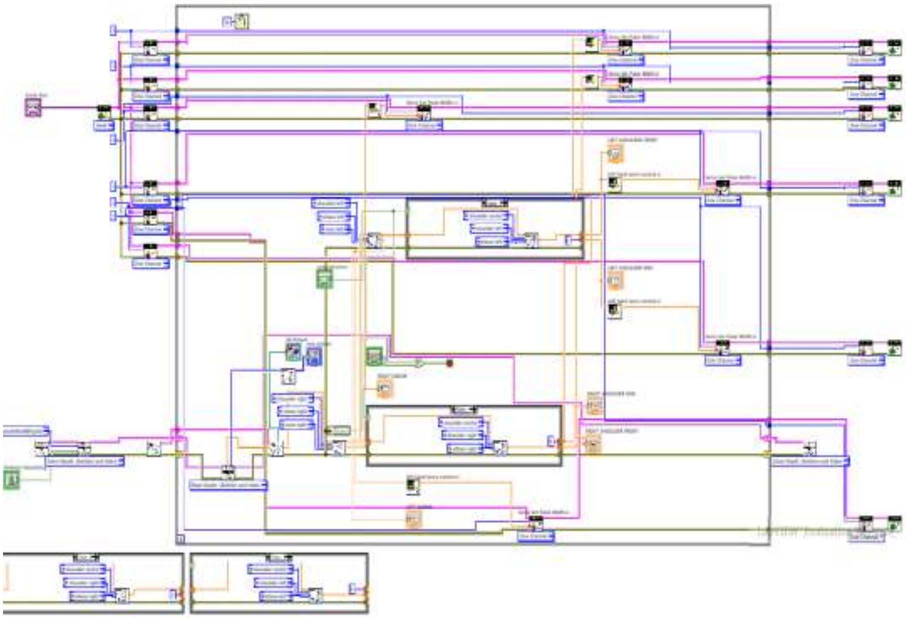


Figure 9: LabVIEW program of the HMI

The software program is designed and tested for error correction and clear understanding of concepts and movements of the robot. The hardware connection is made and the working is executed to verify the outputs in order to obtain the desired results. The testing phase plays a key role in implementation of the control system. It is the time period where the complete working of the module and its accuracy to the desired result is deduced, and finally a working prototype is done.

5. WORKING OPERATION

The working of the control system, the HMI designed and developed using LabVIEW and the robot is as described in the following steps:

Step 1: The skeletal data is acquired from the Kinect sensor using the Kinect SDK and LabVIEW toolkit using KINESTHESIA.

Step 2: The Arduino mega Microcontroller is interfaced with LabVIEW using the LINX toolkit.

Step 3: The robots Servo motors are connected to the Arduino with respective PWM pins.

Step 4: As the user comes into the frame of the Kinect sensor, their movement is captured and processed to obtain the joint angle data in Real time.

Step 5: Then the obtained joint angle data is converted to a PWM signal using an algorithm created and used as a subVI to transmit it to the servo motor in order to obtain the desired movements or results.

Step 6: The PWM signal values calculated are transmitted to the robot via Arduino which is synchronously connected to the HMI developed or programmed in the LabVIEW environment to execute the task and acquire the desired results.

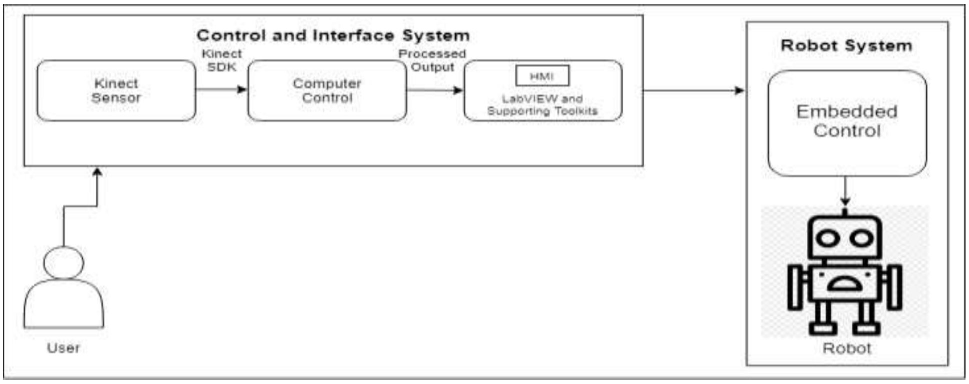


Figure 10: The overall working block diagram.

6. RESULTS AND DISCUSSION

The following figure shows the working and demonstration of the robot's movements and the Front Panel Design of the HMI. The Front Panel shows the mirrored skeletal tracking of the user. The following figures show the simulation and the control of the humanoid robot. The gestures made by the user are mimicked by the robot.

The Fig 11 shows the simulated and mirrored skeletal tracking of the user's gestures.

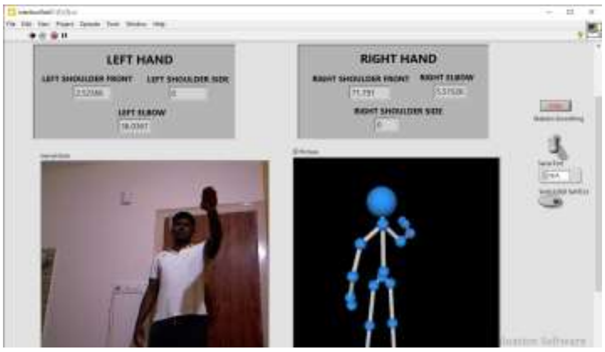


Figure 11: Front panel view of the Control system of HMI

Fig 12 demonstrates controlling the right shoulder joints of the humanoid robot.

Figure 12: Right shoulder movement tracking.

Fig 13 demonstrates controlling the left shoulder joints of the humanoid robot.

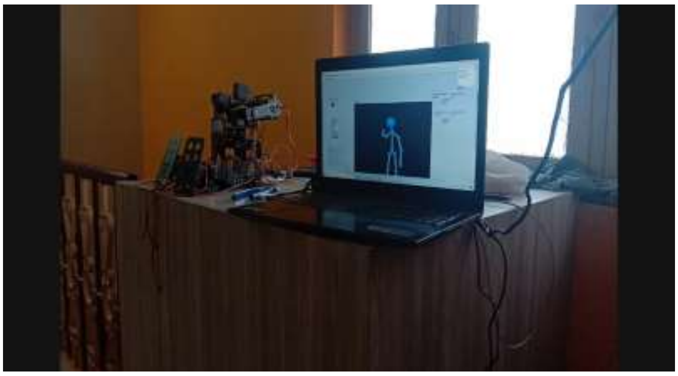


Figure 13: Left shoulder movement tracking

Fig 14 demonstrates controlling the left arm of the humanoid robot. It must be noted here that the arm of this humanoid robot has 3 joints. 2 of these joints make up the shoulder movements and the other makes up the elbow.

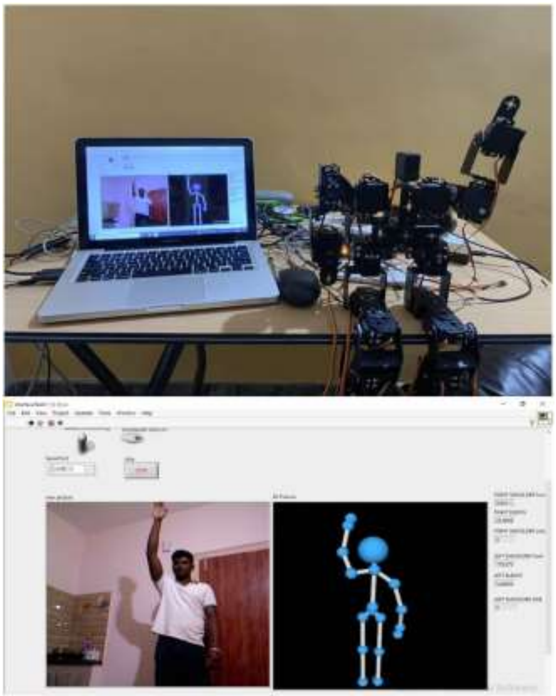


Figure 14: The front panel and the robot's movement.

Fig 15 demonstrates controlling the right elbow of the humanoid robot.

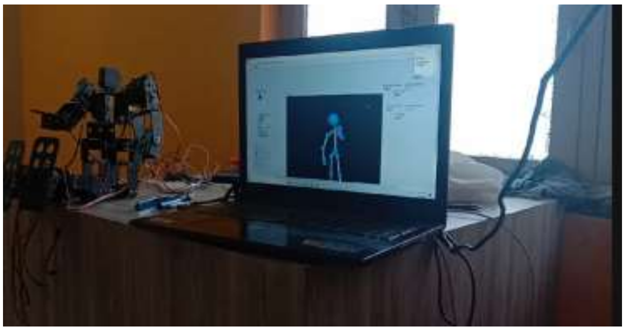


Figure 15: Right elbow movement tracking

Fig 16 demonstrates controlling both the elbows of the humanoid robot.

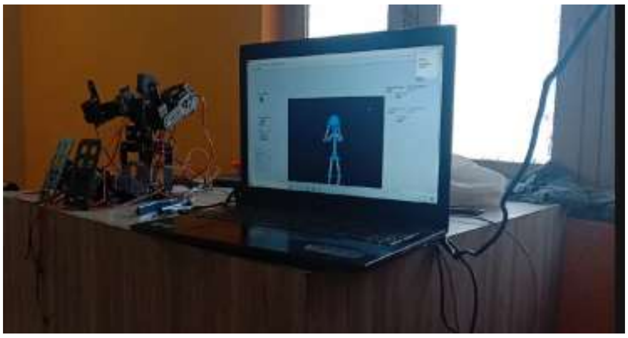


Figure 16: Both Elbows motion tracking.

7. CONCLUSION

In this paper we presented a humanoid control system that allows to mimic its motion by human demonstration, using the affordable Kinect sensor to capture the motion and NI LabVIEW software to design the algorithm. The joint angles are calculated using this algorithm, which collects data from joints in the skeletal image and uses it to mimic gestures in real time.

This control system can also be enhanced to function as a humanoid robot capable of performing appropriate tasks. With a more high-end processing power and a sufficient battery pack installed, as well as wireless technologies such as Bluetooth and Wi-Fi module, this system can be sent to hazardous environments to perform tasks that ensure human safety. It can also be used as a surveillance robot in confined areas that humans cannot access.

REFERENCES

[1]. Hussein M A, Ali A S, Elmisery F A and R. Mostafa 2014 Motion Control of Robot by using Kinect Sensor Research Journal of Applied Sciences, Engineering and Technology 8(11).

[2]. Angal Y and Gade A 2017 LabVIEW controlled robot for object handling using NI myRIO IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT)

[3]. RATHY G A 2018 Arduino based 6DoF Robot using LabVIEW International Journal of Advanced Research Ideas and Innovations in Technology 4

[4]. Wasielica M, Wąsik M, Kasiński A and Skrzypczyński P 2013 Interactive programming of a mechatronic system: A small humanoid robot example IEEE/ASME International Conference on Advanced Intelligent Mechatronics

[5]. M. Limin and Z. Peiyi 2017 The medical service robot interaction based on Kinect IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS) [6] Muller-Cajar, R. & Mukundan, Ramakrishnan. (2007). Triangulation: A new algorithm for Inverse Kinematics. Image and Vision Computing - IVC.

[6]. Muller-Cajar R and Ramakrishnan M 2007 Triangulation: A new algorithm for Inverse Kinematics. Image and Vision Computing

[7]. Wiedemann, Lukas & Planinc, Rainer & Nemec, Iris & Kampel, Martin. (2015). Performance Evaluation of Joint Angles obtained by the Kinect v2. 10.1049/ic.2015.0142.

[8]. J. A. Tumialán Borja, E. Bernal Alzate and D. L. Mariño Lizarazo, "Motion control of a mobile robot using kinect sensor," 2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC), 2017, pp. 1-6, doi: 10.1109/CCAC.2017.8276474.

[9]. Vasiljevic, Goran. (2012). Kinect-Based Robot Teleoperation by Velocities Control in the Joint/Cartesian Frames. IFAC Proceedings Volumes. 45. 805-810. 10.3182/20120905-3-HR-2030.00087.

[10]. Babaians, N. Khazaee Korghond, A. Ahmadi, M. Karimi and S. S. Ghidary, "Skeleton and visual tracking fusion for human following task of service robots," 2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM), 2015, pp. 761-766, doi: 10.1109/ICRoM.2015.7367878.

[11]. Constantin, Daniel. (2015). KINEMATIC ANALYSIS OF THE LYNX6 ROBOTIC ARM. XXV. 65-76.

[12]. Afthoni, Rizqa & Rizal, Achmad & Susanto, Erwin. (2013). Proportional Derivative Control Based Robot Arm System Using Microsoft Kinect. 2013 International Conference on Robotics, Biomimetics, Intelligent Computational Systems (ROBIONETICS). ROBIONETIC 2013. pp 25-29. 10.1109/ROBIONETICS.2013.6743572.

[13]. Anurag Shrivastava, Chinmaya Kumar Nayak, R. Dilip, Soumya Ranjan Samal, Sandeep Rout, Shaikh Mohd Ashfaque,Automatic robotic system design and development for vertical hydroponic

farming using IoT and big data analysis,Materials Today: Proceedings,2021,ISSN 2214-7853,https://doi.org/10.1016/j.matpr.2021.07.294.