Simulation And Real Time Of VR Controlled Robotic Manipulator Using ROS

Sasidharan Vairavasamy
Department of Mechatronics engineering
Hindustan Institute of Technology and Science
Chennai, India
sasidharanvairavasamy02@gmail.com

Noel Innocent J
Department of Mechatronics engineering
Hindustan Institute of Technology and Science
Chennai, India
noelinnocent369@gmail.com

Hamin MJ
Department of Mechatronics engineering
Hindustan Institute of Technology and Science
Chennai, India
haminmjudy@gmail.com

Syed Ahmed
Department of Mechatronics engineering
Hindustan Institute of Technology and Science
Chennai, India
sdkool77@gmail.com

Seenu N
Centre for Automation and Robotics, Department of Mechatronics Engineering
Hindustan Institute of Technology and Science
Chennai, India
seenu.n@gmail.com

Ramya MM
Dean
Agurchand Manmull Jain College
Chennai, India
ORCID: 0000-0003-2076-3390
dean@amjaincollege.edu.in

***Abstract* - This paper focuses on developing a virtually controlled robot by integrating Virtual Reality (VR) and Robot Operating System (ROS). Robots can be used in certain environments where humans cannot be physically present to undertake a task. To increase the safety of humans, controlling a robot virtually is one of the best solutions. This paper shows how the VR-controlled manipulator will help in different fields like hazardous environments, underwater research, and the medical field for surgical operations. Unity 3D is used to develop the virtual environment, and ROS is used for communication with the physical robot. A Virtual environment would be developed where an exact robot (URDF) model can be designed. The interaction with the virtual environment is done with the help of VR headsets and controllers. ROS acts as the communication bridge between virtual and physical robots. A prototype has been developed to be controlled through virtual controllers that can interact through ROS.**

***Keywords* - Virtual Reality, ROS, Robot, Environment, Manipulator, Robotic Arm**

I. Introduction

In recent years, Robotic manipulators are being used in many fields such as research sectors, Industries etc. The future of VR technologies is slowly being developed in teleoperated Robotics to solve some of the issues that are faced by Humans. However, using a human to operate a teleoperated manipulator can facilitate tasks and increase safety because a human wouldn't have to physically do a dangerous operation. This Teleoperated Manipulator can be further used in surgeries, hazardous environments, etc. A person can avoid some of the dangerous tasks by being present in a safe environment and completing the tasks using the manipulator, which can be controlled and teleoperated to do the work by the user. A teleoperated VR manipulator allows the user to work from a Virtual environment and control the manipulator physically.

A. Virtual Reality

Virtual reality is a simulated experience to interact with a virtual model, and according to the needs of this paper, virtual reality is the best source for creating the things that augmented reality will not be able to in some situations. Virtual reality is quite beneficial in many ways. For instance, a person can use VR to take a virtual surgery training with the help of the VR manipulator and rehabilitation program using Virtual Reality [1,2], since this paper focuses on the virtual reality being used to control the manipulator inside the virtual environment and movements can be mimicked by the prototypes in the real world concerning the virtual world.

B. Virtualization of the manipulator

Unity3D is a cross-platform game engine that can create 3D and 2D games, interactive simulations, and other experiences. The software helps us make a VR environment where it can apply physics applications and complex geometry, give lighting to the object, and render surfaces as well [3]. To initiate the VR, the connection between the VR headsets and the Unity engine has to be made. The Unity engine realizes that there is VR plugged in, and then the game view changes to a VR view that can only be seen through the VR headsets. Scripts are used to control and select the joints to move in a forward and backward direction. There is another script to determine the position of the end effector based on the location of the joints. Unity 3D is ideal for interacting with a VR and ROS, as VR controllers are easily accessible via the Steam VR and XR plugins [4].

C. Robotic operating system (ROS)

ROS is a combination of tools, libraries, and conventions to help create numerous robot application tasks. It is used to establish and control communication between a virtual and a physical robot. Unified Robotic Description Format (URDF) is an Extensible Markup Language (XML) file format used in ROS to describe all elements and physical property of a robot. It is used by various tools for simulation and visualization [5]. This paper focuses on teleoperating robots using virtual reality (VR) devices. Using the hand controllers in VR devices, users can match their movements to the robot's movements to complete various tasks. Such systems could eventually help humans supervise robots from a distance like hazardous environments, underwater research, and the medical field for surgical operations.

978-1-6654-9291-1/22/$31.00 ©2022 IEEE

II. LITERATURE REVIEW

The anatomy of the manipulator can be easily inspected, visualize and analyze in a Virtual Environment. VR gives a replica of the real world, allowing us to put forward conditions for simulation in VR. With this, it can predict outcomings of the physical environment which enables us to prevent them from getting damaged [6]. It planned the trajectory of a manipulator using two algorithms, compared them, and plotted the results on a graph using ROS. In a complex environment, the visualization tool performed trajectory planning with collision avoidance [7]. The usage of the Teleoperation Manipulator has brought forth a lot of opportunities due to its feature of being present in one environment and working in a different environment. It can replicate the same workspace of the real world and gives the feature of moving in a 360-degree Virtual environment that enables one to do work in the Virtual world [8]. The manipulator being present in a different workspace is controlled by the Hand Controllers of the Virtual Reality. The work done in a Virtual Environment using the manipulator is replicated in the real world using the manipulator which is present in a different real workspace environment [9,10]. As non-technical users find it more effective by using a VR teleoperation interface, which makes them more convenient than other interfaces [11]. Virtual reality enhances users' motor function and balance skills, providing distraction for patients undergoing acute pain treatment [12].

III. VIRTUAL REALITY CONTROLLER

VR is a new generation of computer games and technologies, where its headset allows one to see the virtual world and its controllers enable one to interact with the virtual world. Its feature of requiring movement in a 360-degree environment simulates being in the real world [13]. When a person interacts with the robot, the controller provides position and orientation input to the VR model as well as feedback in the form of vibration.

IV. PROPOSED METHODOLOGY

A. Block diagram

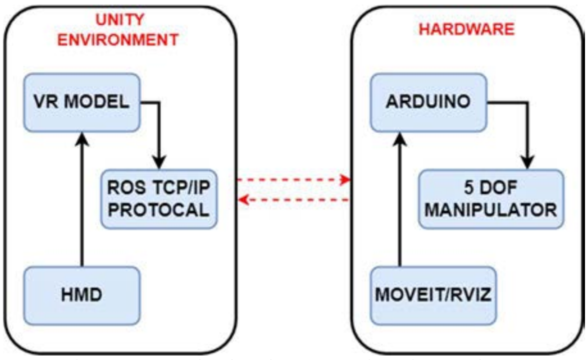

Figure 1: - Block Diagram of Unity and ROS

This [Figure 1] explains the two main fields of this paper, i.e., In a unity environment, first the user will give input through the VR controllers, then the VR model will move to that position with the help of a forward kinematic script. The other ROS subscriber script will keep sending the position and orientation of the manipulator through the ROS TCP/IP protocol. The microprocessor will take care of all low-level actuators. With help from the ROS publisher script, the position and coordination of the VR model are sent to Moveit planner. The Moveit will take the shortest route to its destination. Orientation is transformed to degree value and publisher script publish value in ROS topic. The Arduino will subscribe to the position degree value and convert it into PWM (Pulse-width modulation) value that is sent to the servo. The servo will rotate as per the value and take up the same pose in the real world.

B. Flowchart

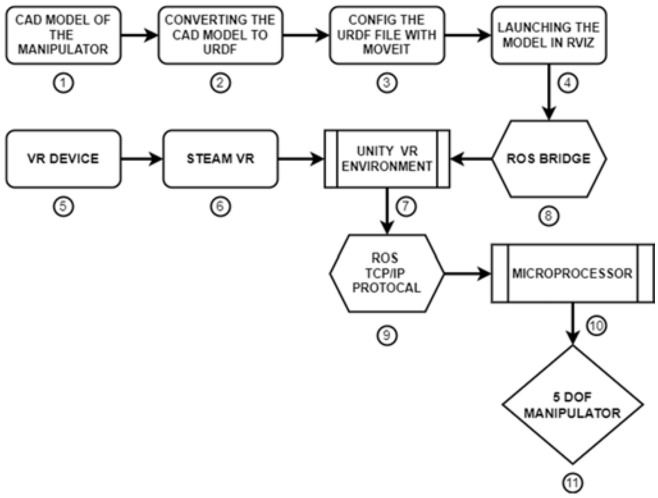

Figure 2: - Flowchart

1. The design of the 3D model of the manipulator in Solid works to give a specific material property such as PLA (Polylactic Acid) and torque, damping with theoretical calculations.

2. The conversion of the design of the Solid works to URDF is done by specifying the links and joints names in the Solid works.

3. The Move IT is utilised for manipulating each joint to develop systems and ROS message files at various angles and positions.

4. Launch RViz (ROS visualisation) when the URDF file has been successfully modified in Moveit.

5&6. The VR Controller and Steam VR are used to interface with the Virtual environment presented in Unity.

7. All the properties defined in the URDF are still present when the imported file is loaded into Unity's 3D environment.

8. When connecting Unity with ROS, ROS Bridge serves as the gateway. In this instance, the manipulator enters the 3D environment that Unity has created by importing the URDF file via ROS Bridge into the Unity Engine.

9. For data transfer using protocols like TCP, JSON API, etc., ROS WebSocket is utilised. The socket uses an IP address to find the client and host. In the TCP protocol, the host publishes the data, and the client subscribes to the data.

10. The microcontroller i.e., "Arduino" takes the signal values from the ROS WebSocket and gives the signal to the prototype.

11. According to the signals the VR controller receives, the 5 DOF manipulator moves.

V. VR SIMULATION

The VR simulation is used to recreate the entirety of the real-world model in virtual reality. In [Figure 3], the model is spammed into the VR environment with help of a ROS web socket using Unity3D. Now we can add all the graphical input and controlling script to the model to make it dynamic. This will force the manipulator to accomplish its goal.

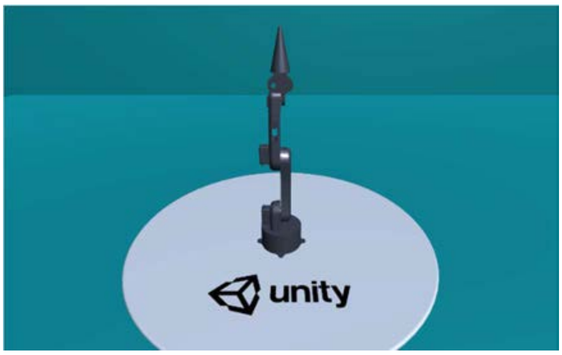

Figure 3: - Virtual Robot in Unity3D

A. ROS web socket

The ROS web socket is used as a bridge between ROS and unity. It uses the JSON API, fusing this with the URDF from ROS, which has been implemented through the help of ROS web socket to unity. As a result, Unity can deploy the robot model and add all functions to it, and ROS web socket allows Unity to connect with ROS and share data between two different frameworks.

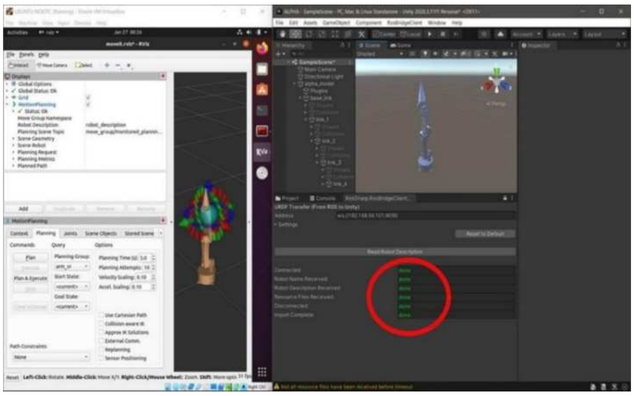

Figure 4: - ROS Web-socket Connection

VI. MATHEMATICAL MODELLING

A. Degree of Freedom (DOF)

Grubler's formula

$$DOF = m(N-1-J) + \sum_{i=1}^{J} f_i$$

Where, m = Number of bodies moving in a given plane, N = Number of links including ground, J = Number of joints, f_i = Individual freedom of joints

m = 6(spatial);N = 6;J = 5; $\sum_{i=1}^{J} f_i$ = 5 * 1(revolute joint) = 5

DOF = 6(6 – 1 – 5) + 5 = 5

Thus, Degree of freedom for this manipulator is **5**.

B. DH Parameters

The link transform is given by a homogeneous transformation matrix in the Denavit-Hartenberg parameter .It contains four parameters where [14]

Joint angle (θ) describes rotation around z_{n-1} required to obtain an x_{n-1} axis coordinate with x_n axis.

Angle Twist (α) describes rotation around x_n required to obtain a z_{n-1} axis coordinate with z_n.

Link length (r or a) describes the length between the centers of (n-1) and (n) frames along the direction x_n.

Link offset (d) describes the length between the centers of (n-1) and (n) frames along the direction z_{n-1}.

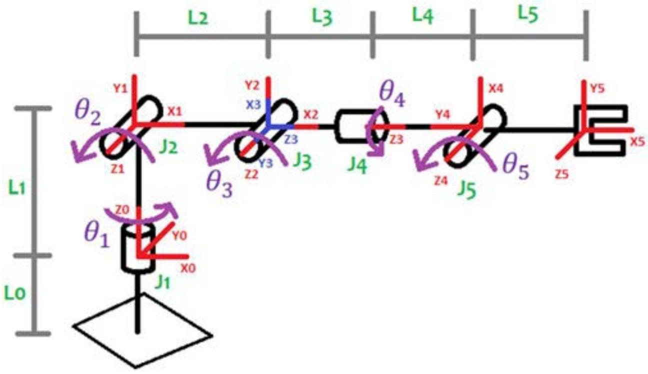

Figure 5:- Kinematic diagram for manipulator

Joints	θ	α	r	d
1	θ_1	90	0	L_1
2	θ_2	0	L_2	0
3	$\theta_3 + 90$	90	0	0
4	θ_4	-90	0	$L_3 + L_4$
5	$\theta_5 - 90$	0	L_5	0

Table 1:- DH Parameter Table

In [Figure 5], the kinematic diagram is utilised for describing DH parameters. The red axis represents the coordinate axis of the respective joints, the blue axis represents the imaginary coordinate axis of joint 3, which is drawn according to rules, and the purple arrow indicates rotation along the axis of the joint. In [Table 1], All parameters are used to convert any DH tables into homogeneous matrix.

$$H_n^{n-1} = \begin{bmatrix} \cos\theta_n & -\sin\theta_n\cos\alpha_n & \sin\theta_n\sin\alpha_n & r_n\cos\theta_n \\ \sin\theta_n & \cos\theta_n\cos\alpha_n & \cos\theta_n\sin\alpha_n & r_n\sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where, (n-1) frame indicate previous frame and (n) frame indicates the current frame. In upper left corner matrix shows a rotation matrix, the upper right corner shows a displacement vector or matrix, the lower left corner matrix shows a projection matrix and lower right corner matrix shows global scaling in computer graphics. The position and orientation of the end effector are determined by applying parameters to the matrix.

C. Torque Calculation

Consider the kinematic scheme for a manipulator with five links, five joints, and one gripper from [Figure 6]. The distance between the joints is represented as the length of the link, which is shown in [Table 2], and the corresponding weight of the joints is shown in [Table 3].

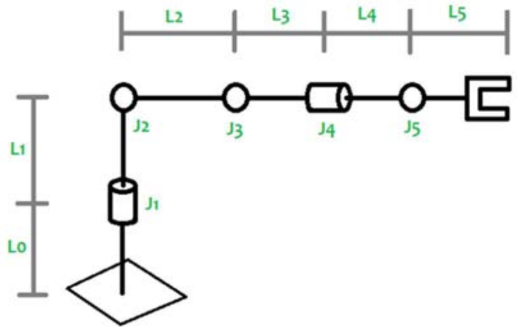

Figure 6: - Kinematic Scheme for Manipulator

Links	Length (cm)	Weight (kg)
Base link	5	0.058
Link 1	4.355	0.050
Link 2	14	0.076
Link 3	10.30	0.047
Link 4	3.04	0.017
Link 5 (gripper)	13.85	0.048

Table 2:- Length and weight of Links

Joints	Weight (kg)
Joint 1	0.055
Joint 2	0.055
Joint 3	0.055
Joint 4	0.009
Joint 5	0.009

Table 3: - Weight of Joints

Gripper

- The centre of mass of the gripper is located at a distance $L_5 \,/\, 2$ from the axis of the rotation of Joint 5
- The weight of link 5 is 0.048 kg
- Thus, the Torque due to the gripper (link 5) is

$$\tau_{Link5} = \frac{L_5}{2} * W_{Link5}$$

Link 4

- The centre of mass of link 4 is located at a distance of $L_4 \,/\, 2$ from the axis of rotation of joint 4 and the centre of mass of the Joint 5 is located at the distance of L_4.
- The weight of link 4 is 0.017 kg and joint 5 is 0.009 kg
- Thus, the torque required to lift the gripper is

$$\tau_{Link_4} = \left(\frac{L_4}{2} * w_{Link_4}\right) + \left(\left(L_4 + \frac{L_5}{2}\right) * w_{Link_5}\right) + \left(L_4 * w_{joint}\right)$$

Link 3

- The centre of mass of link 3 is located at a distance of $L_3/2$ from the axis of rotation of joint 3, and the centre of mass of the Joint 5 is located at the distance of $(L_3 + L_4)$ and the centre of mass of the Joint 4 is located at a distance of $L3$
- Thus, the torque required to lift the gripper and link 4 is

$$\tau_{Link_3} = \left(\frac{L_3}{2} * w_{Link_3}\right) + \left(L_3 + \frac{L_4}{2}\right) * w_{Link_4} + \left(\left(L_3 + L_4 + \frac{L_5}{2}\right) * w_{Link_5}\right) + \left((L_3 + L_4) * w_{joint5}\right) + \left(L_3 * w_{joint5}\right)$$

Link 2

- The centre of mass of link 2 is located at a distance of $L_2 \,/\, 2$ from the axis of rotation of joint 2, and the centre of mass of the Joint 5 is located at a distance of $(L_2 + L_3 + L_4)$, the centre of mass of the Joint 4 is located at a distance of $(L_2 + L_3)$
- Thus, the torque required to lift the gripper, link 4 and link 3

$$\tau_{Link_2} = \left(\frac{L_2}{2} * w_{Link_2}\right) + \left(L_2 + \frac{L_3}{2}\right) * w_{Link_3} + \left(\left(L_2 + L_3 + \frac{L_4}{2}\right) * w_{Link_4}\right) + \left(\left(L_2 + L_3 + L_4 + \frac{L_5}{2}\right) * w_{Link_5}\right) + \left(L_2 * w_{joint3}\right) + \left((L_2 + L_3) * w_{joint_4}\right) + (L_2 + L_3 + L_4) * w_{joint_5}$$

Link 1

- The centre of mass of the link 1 is located at a distance of $L_1 \,/\, 2$ from the axis of rotation of joint 1, and the centre of mass of the Joint 5 is located at a distance of $(L_1 + L_2 + L_3 + L_4)$ and centre of mass of the Joint 4 is located at a distance of $(L_1 + L_2 + L_3)$
- Thus, the torque is required to lift link 2, link 3, link 4 and gripper.

$$\tau_{Link_1} = \left(\frac{L_2}{2} * w_{Link_1}\right) + \left(L_1 + \frac{L_2}{2}\right) * w_{Link_2} + \left(\left(L_1 + L_2 + \frac{L_3}{2}\right) * w_{Link_3}\right) + \left(\left(L_1 + L_2 + L_3 + \frac{L_4}{2}\right) * w_{Link_4}\right) + \left(\left(L_1 + L_2 + L_3 + L_4 + \frac{L_5}{2}\right) * w_{Link_5}\right) + \left(L_1 * w_{joint2}\right) + \left((L_1 + L_2) * w_{joint_3}\right) + \left((L_1 + L_2 + L_3) * w_{joint_4}\right) + \left((L_1 + L_2 + L_3 + L_4) * w_{joint_5}\right)$$

LINK	TORQUE (kg-cm)
τ_{link1}	6.236
τ_{link2}	4.750
τ_{link3}	1.628
τ_{link4}	0.531
τ_{link5}	0.332

Table 4:- Torque Calculation

As a result, torque computation from [Table 4] is utilised to select the servo motors for manipulators, increasing their effectiveness.

VII. SIMULATION & REAL TIME TESTING OF THE PROPOSED SYSTEM

A. 3D Solid works Design

Figure 7: - Solid works design Manipulator

The Grab Cad [15] model has been used in [Figure 7], however certain changes have been made to the model to make it more adaptable for VR applications. After the model has undergone extensive mathematical study, it is converted into URDF (Unified Robotic Description Format) utilizing solid works plugin for URDF conversion.

B. ROS Simulation

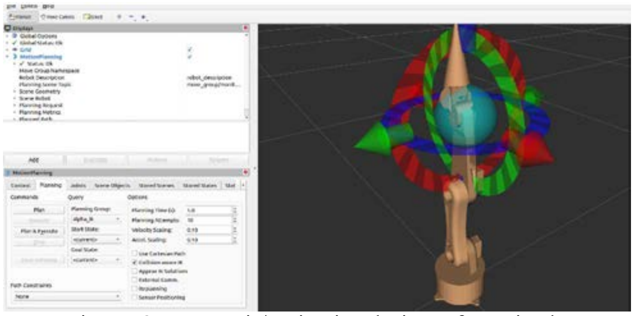

Figure 8: - Moveit/Rviz simulation of manipulator

In [Figure 8], the converted URDF file is launched with the help of the ROS-visualizer tool to cross-check with CAD model. Then the Moveit tool is used to convert the Unified Robotic Description Format (URDF) file to Semantic Robot Description Format (SRDF). This will add the inverse kinematics and the Moveit plugin, so the model will become more usable.

C. Outputs of Manipulator control in Real Time

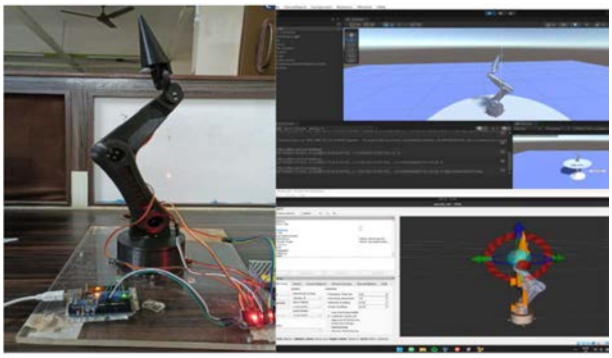

Figure 9: – Movement of the manipulator

The VR model and the real world are connected via the ROS TCP/IP Protocol, and when a user moves the VR model using a VR controller, the same position is transformed into the real world via Moveit and Arduino. Testing has been carried out with various positions, and quick transformations have been obtained. The real model has been accurately transposed to all test inputs. The few test outcomes are shown in [Figure 9,10].

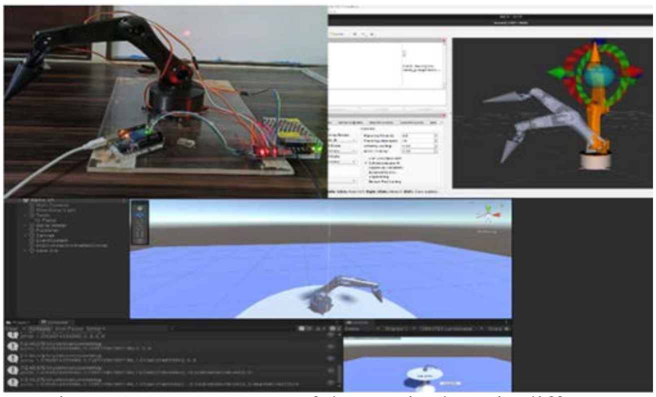

Figure 10: - Movement of the manipulator in different positions

VIII. RESULTS & CONCLUSION

Teleoperated manipulators are on the rise. The use of this technology enables the user, who is present in a safe virtual environment, to control the manipulator present in the physical environment using VR hand controllers. This type of method can help us solve problems faced in many domains, such as medical operations. Emergency operations can easily be done without the doctor being physically present in the hospital. So, this paper mainly focuses on solving the societal issues that cause drawbacks in some work like hazardous environments, space research, medical operations, etc. URDF (Unified Robotic Description Format) was designed to correspond to manipulators that are used in simulation and real-time operations. Moveit is used to apply the kinematics, which are then utilised to establish the location and orientation of the manipulator. Once the connection between Unity3D and ROS was established using the ROS (Robotic Operating System) TCP/IP protocol, the positions and orientations of the simulated model and the real-time model were synchronised. The reliability of the system was tested when, given the position and orientation of a manipulator, a simulated model and a real-time model were achieved, for which it seems effective. In the future, we intend to implement it for a specific application and conduct analysis.

IX. FUTURE WORK

Future work on this article will focus on leveraging ROS to integrate the model with mixed reality. To ensure that the robot is aware of its surroundings, image processing using OpenCV is carried out. Using OpenCV and AI, the robot determines which end-effector will be appropriate for a given environment for a variety of tasks. The robot will be mounted on an AGV to perform its tasks in a mobile style, enabling anyone needs to approach an environment to do so via the AGV, observe the environment, and then complete their mission. Using OpenCV, the robot will recognise a human for their protection and then prevent a collision with them.

REFERENCES

[1] H. de Sousa and S. Balbino, "Development of Immersive Virtual Reality Interface for Lower-Limb Robotic Rehabilitation," 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE). IEEE, Nov. 2018. doi: 10.1109/lars/sbr/wre.2018.00031.

[2] O. Heidari and A. Perez-Gracia, "Virtual Reality Synthesis of Robotic Systems for Human Upper-Limb and Hand Tasks," 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 2019, pp. 966-967, doi: 10.1109/VR.2019.8798321.

[3] R. Codd-Downey, P. M. Forooshani, A. Speers, H. Wang and M. Jenkin, "From ROS to unity: Leveraging robot and virtual environment middleware for immersive teleoperation," 2014 IEEE International Conference on Information and Automation (ICIA), 2014, pp. 932-936, doi: 10.1109/ICInfA.2014.6932785.

[4] J. J. Roldán et al., “Multi-robot Systems, Virtual Reality and ROS: Developing a New Generation of Operator Interfaces,” Studies in Computational Intelligence. Springer International Publishing, pp. 29–64, Jul. 06, 2018. doi: 10.1007/978-3-319-91590-6_2.

[5] J. Chen, H. Deng, W. Chai, J. Xiong and Z. Xia, "Manipulation Task Simulation of a Soft Pneumatic Gripper Using ROS and Gazebo," 2018 IEEE International Conference on Real-time Computing and Robotics (RCAR), 2018, pp. 378-383, doi: 10.1109/RCAR.2018.8621702.

[6] C. Xue Er Shamaine, Y. Qiao, J. Henry, K. McNevin and N. Murray, "RoSTAR: ROS-based Telerobotic Control via Augmented Reality," 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP), 2020, pp. 1-6, doi: 10.1109/MMSP48831.2020.9287100.

[7] H. Tao, Y. Fang, and Z. Zhou, “Research on Trajectory Planning of ROS-based Manipulator,” Proceedings of the 2019 International Conference on Robotics, Intelligent Control and Artificial Intelligence - RICAI 2019. ACM Press, 2019. doi: 10.1145/3366194.3366260.

[8] L. Gammieri, M. Schumann, L. Pelliccia, G. Di Gironimo, and P. Klimant, “Coupling of a Redundant Manipulator with a Virtual Reality Environment to Enhance Human-robot Cooperation,” Procedia CIRP, vol. 62. Elsevier BV, pp. 618–623, 2017. doi: 10.1016/j.procir.2016.06.056.

[9] D. Whitney, E. Rosen, D. Ullman, E. Phillips and S. Tellex, "ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1-9, doi: 10.1109/IROS.2018.8593513.

[10] B. Illing, B. Gaspers and D. Schulz, "Combining Virtual Reality with Camera Data and a Wearable Sensor Jacket to Facilitate Robot Teleoperation," 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), 2021, pp. 649-650, doi: 10.1109/VRW52623.2021.00207.

[11] D. Whitney, E. Rosen, E. Phillips, G. Konidaris, and S. Tellex, “Comparing Robot Grasping Teleoperation Across Desktop and Virtual Reality with ROS Reality,” Springer Proceedings in Advanced Robotics. Springer International Publishing, pp. 335–350, Nov. 28, 2019. doi: 10.1007/978-3-030-28619-4_28.

[12] S. K. Swee, L. Z. You, B. W. W. Hang and D. K. T. Kiang, "Development of rehabilitation system using virtual reality," 2017 International Conference on Robotics, Automation and Sciences (ICORAS), 2017, pp. 1-6, doi: 10.1109/ICORAS.2017.8308045.

[13] D. M. Roach and I. Demirkiran, "Computer aided drafting virtual reality interface," 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC), 2017, pp. 1-13, doi: 10.1109/DASC.2017.8102142.

[14] J. -D. Sun, G. -Z. Cao, W. -B. Li, Y. -X. Liang and S. -D. Huang, "Analytical inverse kinematic solution using the D-H method for a 6-DOF robot," 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 2017, pp. 714-716, doi: 10.1109/URAI.2017.7992807.

[15] “CAD designs, Files & 3D models,” Free CAD Designs, Files & 3D Models | The GrabCAD Community Library. [Online]. Available: https://grabcad.com/library/5-dof-robotic-arm-6. [Accessed: 11-Nov-2022].