A Motion Mapping System for Humanoids that Provides Immersive Telepresence Experiences

Carlos Girard[1], Diego Calderon[1], Ali Lemus[1], Victor Ferman[2] and Julio Fajardo[1,2]

Abstract— **Motion capture and mapping systems have been evolving to enhance the virtual immersion experience in a human-in-the-loop model. In this work, a motion mapping system composed by a 3D printed humanoid robot, built with low-cost materials, an IMU-based motion capture suit, a binaural microphone and a virtual reality headset is presented. The movements of the robot are limited by its reduced amount of degrees of freedom, particularly due to its shoulder configuration, which differs from the human's own biomechanics. Once the motion capture suit's information is extracted, a ROS-based architecture maps orientations from the user to obtain the generalized coordinates of the robot in order to imitate the operator's arm's motion. Additionally, the headset is used to project a stereo vision of the robot's surroundings and to map the operator's head motion. Furthermore, the microphones located on each ear provide the ability to capture 3D sound. This project intends to provide an interactive telepresence puppetry system to encourage the involvement of a targeted audience on engineering subjects. The system shows acceptable results with moderate time response.**

I. INTRODUCTION

Motion control in the field of humanoid robots has been approached with many techniques that involve a vast variety of motion tracking and detection implementations. Since the beginning of robotics, the topic of developing human shaped robots has been a great challenge for engineering, because of the problems that involve replicating human movements, as well as the right acquisition of motion measurements from a human. Therefore, a lot of algorithms and devices had to shape the path for stable and robust systems that can accomplish these goals with acceptable results. One of the first attempts to harness humanoid control was by developing the inverse kinematics models and motion capture (MoCap) systems using cameras [1]. Moreover, having a control system to determine the position of the end effector of a robot is essential for critical applications like surgery assistance where there are human lives at risk [2]. In other applications, such as teleoperated robotic arms, it exists a greater interest on replicating the orientation of the robot links rather than track the cartesian coordinates of the end effector, all this, in the sake that the operator can work more conveniently and intuitively [3].

Furthermore, as telepresence continues expanding to new applications, humanoid robots have been recently added to more immersive forms of human interaction.

[1] Authors are with Turing Research Laboratory, FISICC, Universidad Galileo, Guatemala City, Guatemala, email: {prospektrgirard,diegocdl,alilemus}@galileo.edu

[2] Authors are with Department of Computer Engineering and Industrial Automation, FEEC, UNICAMP, Campinas, SP, Brazil, e-mail: {vferman,julioef}@dca.fee.unicamp.br

The use of virtual reality (VR) headsets can provide the user with a wide view of the information received by the robot's eye-cameras [4]. However, with this principle, comes the problem of latency from the stereo images perceived by the human eyes, producing motion sickness if the frame rate is low or if the images are not well synchronized. Hence, the proper selection of image resolution and frame rate can mitigate those problems [5].

In previous implementations, humanoid robots were designed as attractions on thematic parks with the aim to produce attractions that resemble the human movements, triggering the intention to make a new kind of human-like puppets that can be operated with body movements. Consequently, the methods to acquire human actions were always following the limbs' path's using cameras with certain track points around human body or using motion sensor suits [6,7]. Both approaches have been constantly improved to achieve better and more robust modes of operation. Furthermore, the entertainment industry remains interested in teleoperation and MoCap systems, because the video games and thematic parks are significantly profitable and direct significant resources to generating more research opportunities [8,9]. Otherwise, the use of MoCap systems has grown in other fields such as physiotherapy, where a humanoid robot can provide mechanical assistance to patients [10,11]. The use of Inertial Measurement Unit (IMU) based MoCap systems brings benefits like more information for the interaction between human and robot. Moreover, IMUs contribute with an enhanced representation of each joint (once well calibrated), due to the movement being mapped by a mechanical action and not inferred by a visual model [12].

After achieving a stable motion mapping system, it was possible to think of applications for education and make it accessible for low-income universities to have affordable humanoid robots for their students [13]. On the other hand, it is also reasonable to think new applications for thematic parks or develop human-robot collaborative environments that promote robotic services, aided by virtual reality environments that deliver a more vivid experience. The concept of using humanoid robots to involve people on the robotics field sometimes awakes curiosity and desire to develop more interactive projects of engineering. As a matter of fact, virtual reality devices like the Vibe, HoloLens and Oculus Rift have brought great solutions to head motion tracking and stereo vision, this type of devices have become mainstream, not only for entertainment but also for research applications [14]. Considering that VR headsets like the Oculus Rift offer an easy solution to obtain the head orientation of a human

Fig. 1. The robot with an operator wearing the motion capture system.

operator and, without any mayor processing, send it to a robots' head actuators, not only to imitate its motion, but also to provide a system based on a telepresence system.

This work presents Leonardo GreenMoov, a telepresence humanoid robot that is controlled through a MoCap suit, built with low-cost materials and based on the InMoov project [15]. Furthermore, the robot mapping problem for this specific case is solved using forward and inverse kinematics techniques to deal with the differences, generated by the joint configuration, between the robot's and operator's arms. Moreover, the head of the robot has two high definition cameras, whose images are projected in the Oculus Rift DK2 headset alongside binaural microphones, which provide the teleoperator the sensation of immersion in the environment where the robot is at. Thereby, the system brings a customized experience, interacting with a targeted audience through a telepresence puppetry system [16]; guiding and entertaining people in places like universities, theaters, thematic parks, malls and museums, in counterpart to the role of automated receptionist such as works described in [17,18].

The rest of this work is organized as follows: Section II describes the functions and components that are part of the telepresence system. In Section III, forward and inverse kinematics are explained to give a better understanding of how the problem of robot mapping was approached. Finally, the Sections IV and V list the experimental results and conclusions obtained through simulations and the use of the system in real scenarios.

II. System Architecture

A modular system is proposed in order to control the robot remotely, as well as enabling or disabling specific modules depending on the user's needs (teleoperated or automatic mode). First, the system is composed by the Axis Neuron Pro (MoCap System Software), which is the only way to extract the information from the MoCap suit and it is available for Windows and macOS but not for Linux based distributions; the first option was chosen to keep the cost as low as possible.

The humanoid robot uses a Pine A64 2GB single-board computer (SBC) with Ubuntu MATE 16.04, which runs the

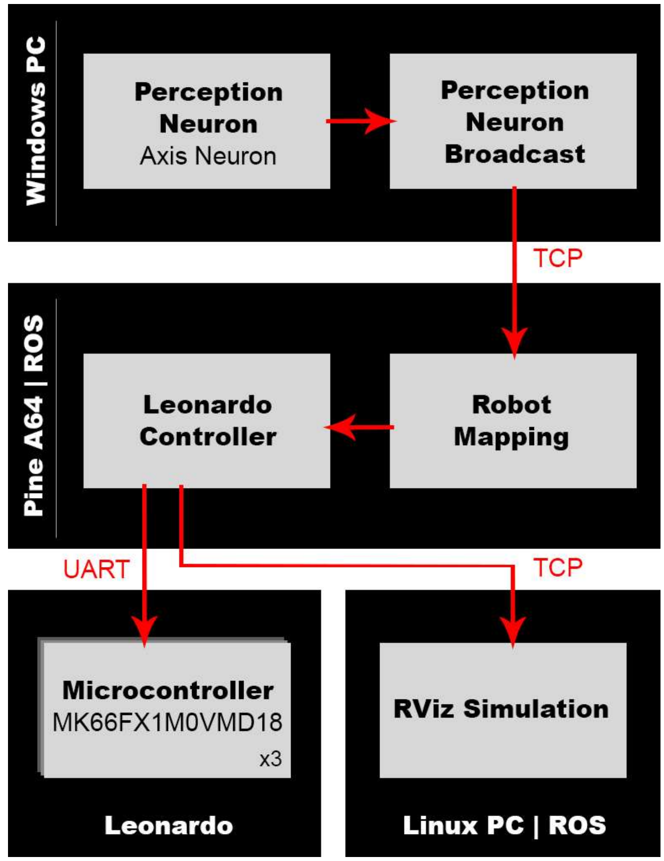

Fig. 2. Block diagram of the system architecture implemented.

master and other nodes of Robot Operating System (ROS) Kinetic to implement the robot mapping and Leonardo's Controller. This SBC was selected, because it is affordable (about US $30) and has low energy consumption. For an easy development, the system also includes an RViz simulation, which allows to run tests without the need to have the robot connected.

Besides, three microcontrollers (MCUs) are connected to control the actuators of the physical robot through three point-to-point UART communication ports using the *rosserial* package. The Perception Neuron Suit is connected to a Windows PC using a USB port or WiFi, this way, the Perception Neuron Broadcast can be connected to the Axis Neuron Pro to acquire the packages and send them to the ROS master. Furthermore an Oculus Rift headset is connected to a Windows PC over HDMI and USB in order to extract user's head movements and sends them to the ROS master in the same way that is done with the Perception. Then, all the modules interact through ROS topics with the aim to update the servomotors' positions using Pulse Width Modulation (PWM) generated by the MCUs.

A. Perception Neuron

The Perception Neuron is a full-body MoCap System developed by Noitom Ltd. It is composed of "neurons", which consist in IMUs located in specific position and orientation of the suit that grants a consistent coordinate system. These

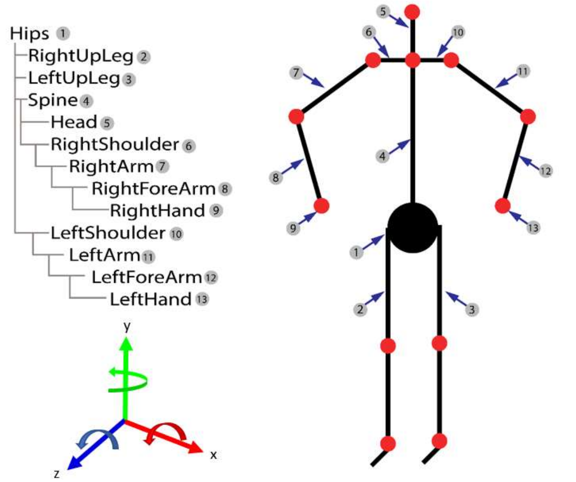

Fig. 3. The BVH format diagram for the Perception Neuron data.

neurons are connected to a HUB that collects, processes and packages information about the motion captured from the operator, transferring it at a frequency of about 60 Hz in the full body configuration. Noitom Ltd. provides a software called Axis Neuron Pro which allows the extraction of sensors' data and the development of third-party software using a C/C++ API.

The Axis Neuron Pro acts as a TCP server, while the Perception Neuron Broadcast node imports the C/C++ API and was developed to establish a connection to this server. The API provides a callback function that is called every time the server publishes a new data frame, with information about the kinematics of the human body represented with the Biovision Hierarchical format (BVH). This is a format for representing MoCap data that is commonly used in the industry of animation and production of movies. It is composed of "bones", where the human body is described using a tree structure with the proper Euler angles of each bone [19]. In this case, the "Hips" represent the ROOT of the tree, as illustrated in Fig. 3. The Perception Neuron Broadcast uses the *rosserial_windows* node to connect to the ROS core running in the Pine A64 and, every time the callback function is called, it takes the data frame and sends it as ROS topics for their use in the robot mapping node.

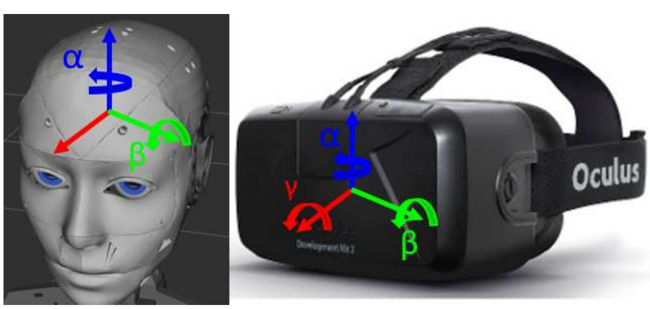

Fig. 4. Representation of coordinate frames for the Oculus Rift headset and for the robot's head; x axis in red, y in green, and z in blue.

B. Oculus Rift

The Oculus Rift is the VR headset used in this system. It allows the operator to view what the robot is seeing through its eye-cameras, bringing an immersing experience to the users [20]. This is achieved using two HD cameras placed into the eyes of the robot, capturing video that is later processed and projected into the Oculus display. Furthermore, the headset is used to get the information needed for imitating the operator's head movements with the robot's head, complementing the generalized coordinates provided by the MoCap system, as shown in Fig. 4. This is obtained using the Oculus SDK using a Windows PC. Then, the information about the Euler angles of the operator's head is sent to the Leonardo controller (Fig. 2) in a similar way as the MoCap data.

C. Hardware Configuration

Leonardo GreenMoov is a 3D-printed humanoid robot based on the open source InMoov project [21], which has 51 degrees of freedom (DOF), 27 degrees of actuation (DOA) (five for each hand, one per finger, five for each arm, two for the torso, two for the hips, two for the neck and one for the jaw). In this manner, the only DOAs that are used for this project involve the actuators for the arms and the head, to make a total of 23 servo motors (5 in each arm, 3 in the head and 2 in the torso). This actuators are controlled using three MK66FX1M0VMD18 MCUs, from NXP. Besides, it has two cameras in the eyes, a speaker in the hips and a set of binaural microphones placed in the robot's ears.

D. Motion Mapping Controller

The Motion Mapping is implemented as a ROS node that subscribes to the topics which have the Euler angles that are sent from the MoCap System and the VR headset. This node processes each message as described in Section III. Then, it sends commands to the Leonardo Controller to execute the movements that the operator is performing. This controller acts as a central module of communication for the physical robot and the RViz simulation (Fig. 6), it receives the motion commands and validates them against the constraints of the physical robot to ensure that the movement will not break the robot for trying to move to an unreachable pose.

III. Motion Mapping

To describe the movements of the robot's arms it is essential to properly allocate the rotation axes into the robot model to have a consistent representation of each DOF. Since the robot needs to perform tasks that require a proper tracking of the orientation of the operator's arms links rather than the precise positions of the elbow and wrist, it is not possible to rely on methods that are only sensitive to changes on the end-effector's position, such as the algorithms based on least squares, e.g. Damped Least Squares (DLS) [22,23]. Moreover, it is imperative to solve the inverse kinematics for the robot's arms configuration with the aim to have a more complying performance. Since there are many options to solve this problem, the most effective one is to manually

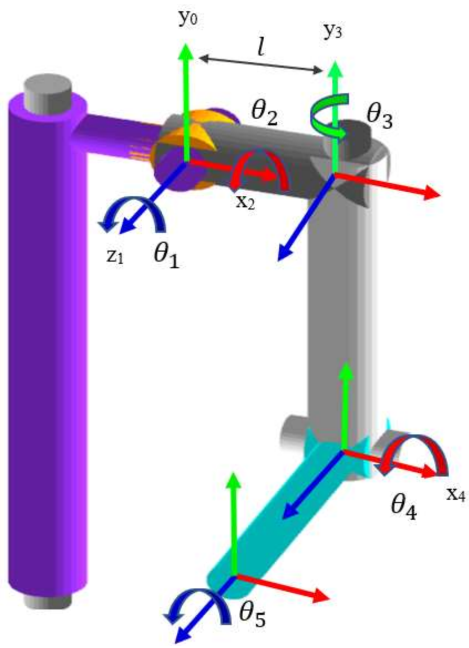

Fig. 5. Leonardo's arms geometry model with its coordinate frames.

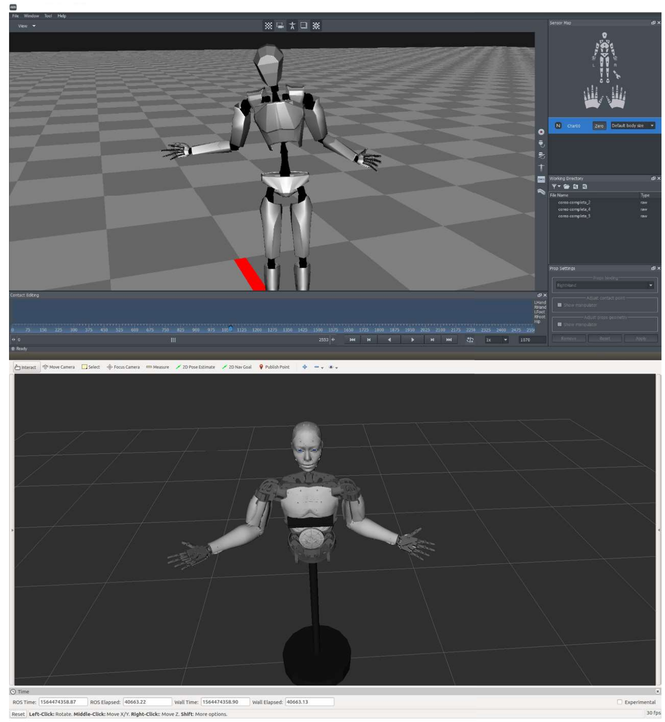

Fig. 6. Axis Neuron visualization and the RViz Simulation.

obtain the equations that describe the rotation of each joint of the robot's arms. To describe the geometry of the motion performed by the link between the shoulder and the elbow of the robot, due the intrinsic and extrinsic rotations of its joints ($\theta_1 - \theta_3$), as shown in the Fig. 5, it was necessary to generate the homogeneous transformation that describes the orientation of the shoulder through its rotation matrix $\mathbf{R_{03}}$, as it is shown in the Equation (1).

In order to obtain the equations that map the generalized coordinates $\mathbf{q}$ of the robot's arms configuration (shoulder and elbow, as shown on Fig. 5) the proper sequence of Euler angles obtained from the MoCap suit were used. For this case of study it was used the traditional notation from the defined rotation matrix $\mathbf{R_{03}}$ described in the equation (1), which transforms the orientation of the elbow from the coordinate frames 0 (shoulder) to 3 (elbow), and that is composed by three column vectors named: normal, slide and approach, where $\mathbf{R}_{03} = [\mathbf{n}, \mathbf{s}, \mathbf{a}]$, with $\mathbf{R}_{03} \in \mathbb{R}^{3x3}$, $\mathbf{n} \in \mathbb{R}^3$, $\mathbf{s} \in \mathbb{R}^3$ and $\mathbf{a} \in \mathbb{R}^3$. Namely, each one of these vectors has an x, y and z component. These vectors compose a 3×3 matrix that describes each one of the rotations that the robot executes, this is shown below:

$$\mathbf{R}_{03} = \begin{bmatrix} C_1C_3 - S_1S_2S_3 & -S_1C_2 & S_1S_2C_3 \\ S_1C_3 + C_1S_2S_3 & C_1C_2 & S_1S_3 - C_1S_2C_3 \\ -C_2S_3 & S_2 & C_2C_3 \end{bmatrix} \quad (1)$$

The presence of the spherical joint that humans have in the shoulder makes it very difficult to map movements directly to the actuators in a humanoid robot. The task of replicate this type of joints brings great difficulties of design and a more expensive actuator selection. Nevertheless in this case it was the robot manage to execute the task needed for the studies that were made with out this type of joints. At first, it was necessary to perform the analysis of forward kinematics (FK) from the robot's arms configuration (shoulder to elbow) to obtain the rotation matrix previously mentioned. Therefore, the orientation of each frame was delimited with the Euler angles' sequence using the Tait-Bryan convention ZXY; it is also relevant to mention that there was no direct mapping configuration to match the proper Euler angles of the MoCap suit. Also, the FK (shoulder to elbow) from the MoCap suit is necessary in order to execute the motion mapping properly by obtaining the orientation of the elbow, since the human body does not have a displacement l on the shoulder like the robot does. Therefore, this guarantees to having an effective way to map the controller actions, since this is also an open loop system that does not have a feedback from the links position. This is the fundamental reason for which the system does not qualify as a control for human operations, instead it is a robot mapping solution that centers in imitating the arms' orientation within the robot's workspace.

The generalized coordinates for the robot's arm's configuration are presented in the Equation (2). After knowing the orientation of the forearm (at the elbow joint), it is possible to directly map the intrinsic rotations of the elbow and wrist of the MoCap system using the following Euler angle sequence ZYX $= [\alpha, \beta, \gamma]$, where $\theta_4 = \gamma$ and $\theta_5 = \alpha$. This also relieves the system from singularities, hence there does not exist a scenario where this type of events can occur. Fig. 5 illustrates which part of the arm is controlled by each one of the angles derived from the rotation matrices. Also this diagram was used as reference to elaborate the FK from the robot's arms. As can be seen in Fig. 3, the same axis frame was used to generate the FK of the MoCap and have a better

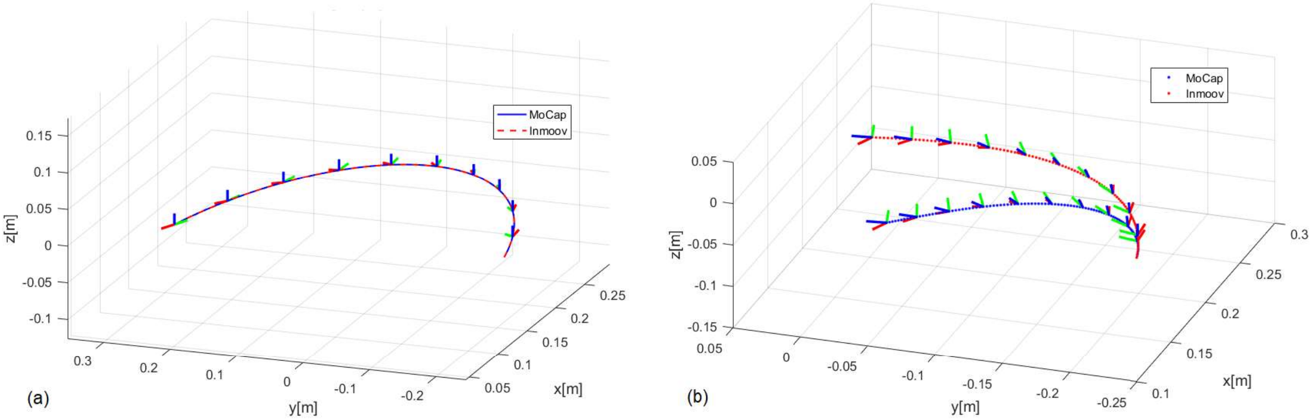

Fig. 7. Simulation's results showing two different trajectories of Leonardo's left elbow.

coherence with the further calculations for the robot's arm orientation.

$$\mathbf{q} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix} = \begin{bmatrix} \operatorname{atan2}\left(-\mathbf{s}_x, \mathbf{s}_y\right) \\ \sin^{-1}\left(\mathbf{s}_z\right) \\ \operatorname{atan2}\left(-\mathbf{n}_z, \mathbf{a}_z\right) \\ \gamma \\ \alpha \end{bmatrix} \tag{2}$$

The fact of keeping the axes in the same orientation makes a simpler implementation than using more general approaches, like the Denavit-Hartenberg algorithm, that grants a particular solution that can be generated by aligning the reference axes in any other orientation, but this may produce further complications at the moment of assigning a positive or a negative rotation to the robot's actuators.

On the other hand, the head of the robot can be oriented utilizing the Euler angles provided by the Oculus Rift as can be appreciated in Fig. 4. In addition to that, the image resolution and the frame rate can be modified to ensure a fluid and vivid experience that does not fatigue the user during operation. Altogether, the system is able to map the orientation of the robot's arms, reacting to actions of opening and closing the hands by setting a threshold in the fingers' orientation. The mouth can also be configured to work synchronously with voice or text-to-speech nodes to provide a more realistic experience for the robot collaborator. The system does not have any type of response for the joints placed on the hips of the robot.

IV. Results

In order to validate how the motion mapping problem is addressed for this specific case, some tests were made under different scenarios using MATLAB and ROS. These tests produced acceptable results obtained by the system in real situations, as shown in Fig. 1. On these tests, different left arm trajectories, composed of a time window of 100 Euler angles for each joint, were generated and then processed. In all tests, the angular velocities from the operator's joints were assumed as constants, as well as the robot workspace's constraints. Moreover, a visual resemblance of the animation generated by the MoCap suit, alongside with the virtual robot controlled by the Motion Mapping node, is shown in Fig. 6. This representation is shown using the ROS Visualization tool (RViz). The Fig. 7(a) shows a scenario where the robot's left elbow tracks the operator's motion with almost a perfect match; this test was performed using a trajectory composed of Euler angles on the ZYX configuration; from the rest position to the elbow's orientation denoted by $\left[\frac{3\pi}{4}, 0, 0\right]$. On the other hand, the Fig. 7(b) describes a movement where the shoulder displacement of the robot makes it impossible to reach the same position as the MoCap system throughout the trajectory; however, it always follows the same orientation as the operator's elbow. This test was performed using $\text{ZYX} = \left[\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{6}\right]$ as final orientation of one of the generated trajectories, also starting from the rest position. Hence, the resulting generalized coordinates for this particular test are represented in a subset of $\mathbf{q}$, described as $\hat{\mathbf{q}} = [\theta_1, \theta_2, \theta_3]^T \approx [1.1832, 0.3614, 0.8571]^T$. Finally, as can be appreciated in Fig. 8, the position error reaches moderate values, about $[x, y, z]^T \approx [0.038, 0.022, 0.071]^T$ meters, as shown in the Fig 7.(b), scenario where the endeffector tracks the trajectory in terms of position is not achievable for the robot.

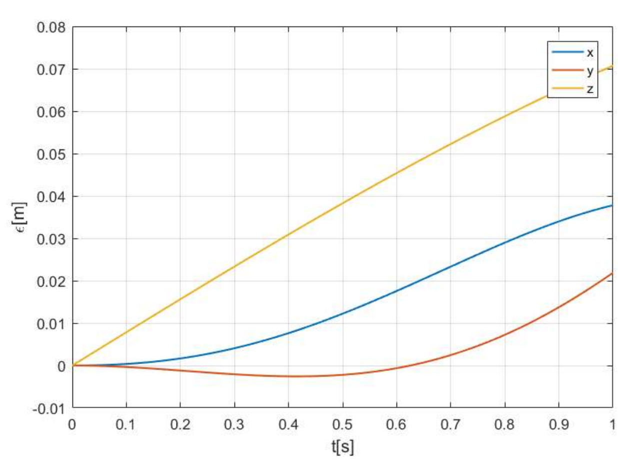

Fig. 8. Elbow position error for the test presented in Fig. 7(b).

Fig. 9. Robot Head controlled using the Oculus Rift DK2.

Besides, satisfactory results were obtained using the Oculus Rift DK2 to map motions from the operator's head, as shown in Fig. 9. The only rotational motion constraint is the absence of the intrinsic rotation about the x axis in the robot's neck (roll angle) due the amount of DOAs in the robots head's configuration, as mentioned in Subsection II.C and as is illustrated in Fig. 4.

V. Conclusions

Since the robot's head provides an immersive VR experience to the operator, providing different and innovative, the robot's surroundings are presented through stereo images and sound. Therefore, it is interesting for future works to measure both user and audience experiences after interacting with the system. In this case, the concept of using humanoid robots to involve people on the tech fields has been shown to different types of audience and achieving great acceptance. This statement inspires to evaluate the human robot interaction on future tests. Moreover, the arms' orientations are replicated at a successful level and, in some cases, could achieve the exact orientation of the operator's arm, depending on the proper implementation of a closed loop control system and on the actuators' constraints. The position error may vary, because of the displacement l in the robot shoulder's configuration. As matter of fact, in some tests, the suit loses precision relatively faster than the VR headset and is necessary to recalibrate it to continue working properly. Finally, the system has been tested during long iterations, of a maximum of 3 hours, without showing fatigue or misalignment.

References

[1] N. S. Pollard, J. K. Hodgins, M. J. Riley, and C. G. Atkeson, "Adapting human motion for the control of a humanoid robot," in *Proceedings 2002 IEEE international conference on robotics and automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 1390–1397.

[2] G. H. Ballantyne, "Robotic surgery, telerobotic surgery, telepresence, and telementoring," *Surgical Endoscopy and Other Interventional Techniques*, vol. 16, no. 10, pp. 1389–1402, 2002.

[3] D. Gong, J. Yu, and G. Zuo, "Motion mapping for the heterogeneous master-slave teleoperation robot using unit dual quaternions," in *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2017, pp. 1194–1199.

[4] K. Theofilis, J. Orlosky, Y. Nagai, and K. Kiyokawa, "Panoramic view reconstruction for stereoscopic teleoperation of a humanoid robot," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 242–248.

[5] U. A. Chattha and M. A. Shah, "Survey on causes of motion sickness in virtual reality," in *2018 24th International Conference on Automation and Computing (ICAC)*. IEEE, 2018, pp. 1–5.

[6] K. Rohr, "Towards model-based recognition of human movements in image sequences," *CVGIP: Image understanding*, vol. 59, no. 1, pp. 94–115, 1994.

[7] D. Fitzgerald, J. Foody, D. Kelly, T. Ward, C. Markham, J. McDonald, and B. Caulfield, "Development of a wearable motion capture suit and virtual reality biofeedback system for the instruction and analysis of sports rehabilitation exercises," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 4870–4874.

[8] J. R. Whitson, "The new spirit of capitalism in the game industry," *Television & New Media*, p. 1527476419851086, 2019.

[9] D. N. Aratuo and X. L. Etienne, "Industry level analysis of tourism-economic growth in the united states," *Tourism Management*, vol. 70, pp. 333–340, 2019.

[10] P. Morasso, M. Casadio, P. Giannoni, L. Masia, V. Sanguineti, V. Squeri, and E. Vergaro, "Desirable features of a "humanoid" robot-therapist," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2009, pp. 2418–2421.

[11] A. Guneysu, B. Arnrich, and C. Ersoy, "Children's rehabilitation with humanoid robots and wearable inertial measurement units," in *Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare*. ICST (Institute for Computer Sciences, Social-Informatics and ..., 2015, pp. 249–252.

[12] N. Miller, O. C. Jenkins, M. Kallmann, and M. J. Mataric, "Motion capture from inertial sensing for untethered humanoid teleoperation," in *4th IEEE/RAS International Conference on Humanoid Robots, 2004.*, vol. 2. IEEE, 2004, pp. 547–565.

[13] H. Yi, C. Knabe, T. Pesek, and D. W. Hong, "Experiential learning in the development of a darwin-hp humanoid educational robot," *Journal of Intelligent & Robotic Systems*, vol. 81, no. 1, pp. 41–49, 2016.

[14] P. Rezeck, F. Cadar, J. Soares, B. Frade, L. Pinto, H. Azpurua, D. G. Macharet, L. Chaimowicz, G. Freitas, and M. F. M. Campos, "An immersion enhancing robotic head-like device for teleoperation," in *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*. IEEE, 2018, pp. 164–169.

[15] G. Langevin, "Inmoov open-source 3d printed life-size robot," *pp. URL: http://inmoov.fr, License: http://creativecommons.org/licenses/by–nc/3.0/legalcode*, 2014.

[16] M. Sakashita, T. Minagawa, A. Koike, I. Suzuki, K. Kawahara, and Y. Ochiai, "You as a puppet: Evaluation of telepresence user interface for puppetry," in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 2017, pp. 217–228.

[17] F. Bazzano and F. Lamberti, "Human-robot interfaces for interactive receptionist systems and wayfinding applications," *Robotics*, vol. 7, no. 3, p. 56, 2018.

[18] R. Karunasenal, P. Sandarenu, M. Pinto, A. Athukorala, R. Rodrigo, and P. Jayasekara, "Devi: Open-source human-robot interface for interactive receptionist systems," 2019.

[19] M. Meredith, S. Maddock *et al.*, "Motion capture file formats explained," *Department of Computer Science, University of Sheffield*, vol. 211, pp. 241–244, 2001.

[20] S. M. LaValle, A. Yershova, M. Katsev, and M. Antonov, "Head tracking for the oculus rift," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 187–194.

[21] W. Xu, X. Li, W. Xu, L. Gong, Y. Huang, Z. Zhao, L. Zhao, B. Chen, H. Yang, L. Cao *et al.*, "Human-robot interaction oriented human-in-the-loop real-time motion imitation on a humanoid tri-co robot," in *2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2018, pp. 781–786.

[22] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Journal of dynamic systems, measurement, and control*, vol. 108, no. 3, pp. 163–171, 1986.

[23] A. A. Maciejewski and J. M. Reagin, "A parallel algorithm and architecture for the control of kinematically redundant manipulators," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 405–414, 1994.