www.czasopisma.pan.pl PAN www.journals.pan.pl

jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2021, VOL. 67, NO. 3, PP. 379-384
Manuscript received September 6, 2020; revised July, 2021 DOI: 10.24425/ijet.2021.137823

Design of a Head Movement Navigation System for Mobile Telepresence Robot Using Open-source Electronics Software and Hardware

Tan Jia Wee, and Herman Wahid

Abstract—**Head movement is frequently associated with human motion navigation, and an indispensable aspect of how humans interact with the surrounding environment. In spite of that, the incorporation of head motion and navigation is more often used in the VR (Virtual Reality) environment than the physical environment. This study aims to develop a robot car capable of simple teleoperation, incorporated with telepresence and head movement control for an on-robot real-time head motion mimicking mechanism and directional control, in attempt to provide users the experience of an avatar-like third person's point of view amid the physical environment. The design consists of three processes running in parallel; Motion JPEG (MJPEG) live streaming to html-Site via local server, Bluetooth communication, and the corresponding movements for the head motion mimicking mechanism and motors which acts in accordance to head motion as captured by the Attitude Sensor and apparent command issued by the user. The design serves its purpose of demonstration with the usage of basic components and is not aimed to provide nor research with regards to user experience.**

Keywords—**Telepresence, head movement control, real-time, video streaming, Bluetooth communication**

I. Introduction

IMMERSIVE telepresence is a technology on the rise in the 21st century. It is used in teleoperations, conferencing, and much more. Unlike virtual reality, for instance the immersive gaming experience, instead of experiencing a virtual, non-existing world, telepresence technology provides stimuli to a person's senses, primarily the sense of sight and hearing that allows one to feel as though they are in another location instead of the one they are really in. In simpler terms, the technology grants one to perceive from a third-person point of view.

The quintessential example of the application of telepresence technology is telepresence video conferencing. Different from conventional video conferencing, the use of the state-of-the-art video telephony can provide users with a significantly enhanced sense of fidelity in both sound and sight. Various benefits have propelled companies into opting for telepresence, and that includes the reduction of travel expenses, reduction of environmental impact and carbon footprints, improving productivity while encouraging efficiency, and improving work life balance [1][7][8]. By the means of a mobile robot incorporated with telepresence, its user will be able to hover around the vicinity without being physically present at a venue. "I can actually just telepresence myself and navigate around the office, speaking to all the employees," says David Merel, the CEO of a small business that uses robots from the telepresence company Double Robotics [2].

Aside from communication purposes, it also allows maneuvering at remote and inaccessible environments, for example in the 2003 Mars Explorer Rovers (MER), Spirit and Opportunity project. Both rovers are tele-robots collaboratively operated by 120 scientists and engineer, which in turn allow those personnel to experience the surface of an unknown world [3][9]. In conjunction with telepresence technology, teleoperation is also widely used in scenarios involving hazardous environments. In the nuclear fallout at the Chernobyl power plant for instance, the construction of the sarcophagus and the attempted clearing of the radioactive wastes involves the usage of teleoperated robots [4][10].

In an ideal case of teleoperation, operators should be provided an immersive first-person perspective of a robot with minimum delay of the information they are to be provided. However, regular remote presence robots and consumer telepresence robots project their camera vision through a video feed on a two-dimensional screen. As depth perception is essential in most operations, the lack of a stereoscopic display will hamper a robot operator's judgement about distance, when in turn might lead to unwanted collisions due to misestimations of the surrounding [5][6][11].

To showcase a teleoperation system, a robot car capable of performing a simple teleoperation, tele-maneuvering, was constructed. The features of the product include immersive live footage, and head movement control for the head motion mimicking mechanism and the robot's directional control. There are two modes, maneuvering unbounded to head movement, and maneuvering bounded to head movement.

II. Methodology

A. System Workflow

Three main modules have been identified for this project as depicted in Fig 1: image transmission, head motion tracking, and robot movement control. For image transmission, the images will come from the camera to the Raspberry Pi, then transmitted to the mobile phone via wireless transmission. The display on the phone will then be viewed by the user through the VR headset.

For head motion tracking, the pitch, yaw, and roll data from the attitude sensor will be collected by the Arduino and transmitted to the Raspberry Pi via Bluetooth transmission. The data will then be processed at the Raspberry Pi and

This work is supported by the Ministry of Higher Education Malaysia under FRGS, Registration Proposal No: FRGS/1/2020/ICT02/UTM/02/5 & UTM.

Authors are with Universiti Teknologi Malaysia, School of Electrical Engineering, UTM Skudai, MALAYSIA (e-mail herman@utm.my).

© The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0, https://creativecommons.org/licenses/by/4.0/), which permits use, distribution, and reproduction in any medium, provided that the Article is properly cited.

then be translated into servomotor movements at the head motion mechanism.

For robot movement control, the direction of the robot will be determined based on the direction where the user's head is facing, which corresponds to the yaw data of the attitude sensor. With the commands received from the remote control, apparent yaw data will be affiliated with the navigation direction actuated by the motors.

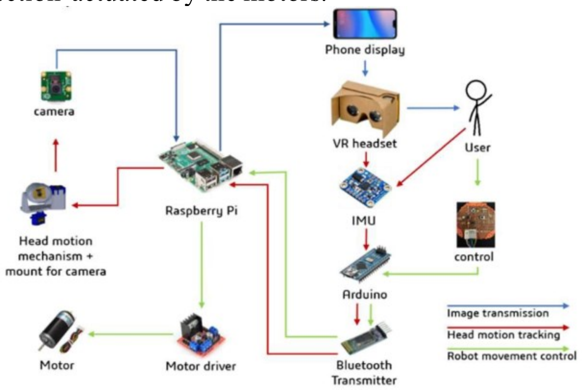

Fig. 1. System Workflow

B. Hardware

Two microcontrollers are used in this project, one responsible for acquiring and transmitting head motion data and user action, while the other for streaming camera footage, receiving the head motion data, and translating them into subsequent actions. The Arduino Maker Uno is chosen at the former, while the Raspberry Pi 4 Model B with 2GB RAM is used as the later due its decent multiprocessing qualities and the availability of four pins capable of hardware pulse-width-modulation (PWM).

For acquiring head motion data, the CJMCU-055 Attitude Sensor is used. It is a 9-axis Intelligent Sensor which directly converts its data into the form of degrees onboard. For Bluetooth transmission, the HC-06 Bluetooth module is opted for.

For the head motion mimicking mechanism, two SG90 Micro Servos are used to mimic the pitch and yaw movement of the human head. Two 5-volt direct current motors are used to drive the wheels. In addition, an l298n motor driver is used to drive the motors.

C. Programming

The programming in this project is divided into two parts which corresponds to the usage of two microcontrollers, which is performed in both C language and Python.

i. Programming the Arduino

The Arduino is responsible for accomplishing two operations: acquiring data from the Attitude Sensor and the controller; and transmitting them via Bluetooth communication. To acquire data from the Attitude Sensor, the default library of the sensor which can be acquired online via open sources is used. The orientation data for the x and y axis are chosen. Data for orientation x is outputted in the range of 0° to 360° whilst the data for orientation y is outputted in the range of -90° to 90°. As the human neck has an effective rotation of approximately 120° in the x-plane, for x, data between 60° to 300° are neglected by limiting angles from 180° to 300° at 300° and 60° to 180° at 60°. Subsequently, both x and y data are translated to corresponding angles within 0° to 180° with 90° as the center with simple if..else's and mathematics to ease the programming with the servos and motors.

Commands are acquired from a wired controller consisting of four buttons in which each are assigned to differing numbers as commands to be read at the Pi which values will be stored. Data x, y, and command are then combined into a string to be transmitted, separated by "/" for the purpose of re-splitting them later on, and ending with "/n" as an identifier marking the end of a transmission instance. For Bluetooth transmission, the SoftwareSerial.h library is used in which the coding is available online.

ii. Programming the Raspberry Pi

Programming at the Raspberry Pi can be further divided into three parts, live streaming, Bluetooth communication, and servo and motor movement. All modules are executed separated in parallel. Two methods are used for that purpose, threading, and multiprocessing. During the earlier phases, only the threading module is used. Variables are shared globally amongst each thread. It is then realized that the quality of the video streaming is hampered due to all four threads, including command processing, being executed in the same process. Hence, the program is spread into three processes, separating all three modules using the Process class within multiprocessing module, whilst the commands are processed at the Arduino instead. Data between the Bluetooth communication process and the servo and motor movement process is accomplished using the Pipe class in the multiprocessing module.

For live streaming, the MJPEG streaming method is used. This is a method that streams footage directly from the Raspberry Pi camera to a html browser that can be accessed via browsers on any devices connected to the same network. As a dual-camera setup has to be stripped from the project due to budget and hardware limitation, the duplication, resizing, and concatenation (as in Fig. 2) of the camera footage is performed in PIL format for the resultant footage to be viewed via the 3D Headset.

Fig. 2. Dimensions of image cropping

Bluetooth communication and servo movement are combined as the servo is to use the x and y data no matter the case. Initially, the PWM signals are provided using the RPI.GPIO module. However, the module has resulted in instability issues such as the servos jittering. Hence, another module by the name of pigpio is opted for. In data receiving via Bluetooth, socket programming is used for communication, where it listens to the receiving port. Data is received in the form of bytes and possibly in fragments. Hence, all data is converted into String and concatenated until the end identifier, "/n", is detected. The variables are split into x, y, and command respectively using data.split with "/" as the splitting identifier. Using the pigpio library's servo function, the pulse width for -90° corresponds to 500, 0° corresponds to 1500, and 90° corresponds to 2500. Hence, their relation is simply calculated using Y=mx+C.

Motor movement involves logic switching at the logic pins at the l298n motor driver for motor direction control. The logic signals and PWM signals are similarly provided by the pigpio library. By acquiring the final command from the command acquiring module, the process determines the current mode, which are 'maneuvering unbounded to head movement', 'maneuvering bounded to head movement', or 'stop'. Each mode translates the command and degree x data into respective actuations of the motor.

D. System Design

The system starts by establishing both Bluetooth and Wi-Fi connection. Subsequently, multiple processes and threads are spawned which flows are as depicted in Fig 3, Fig 4, Fig 5(a) and F ig 5(b).

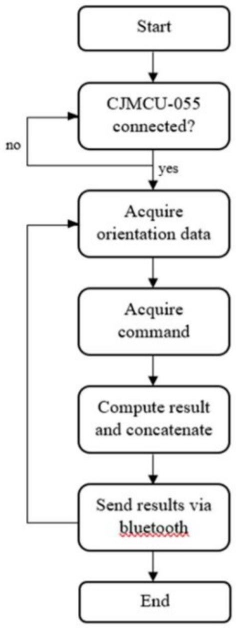

Fig. 3. Process flow at Arduino

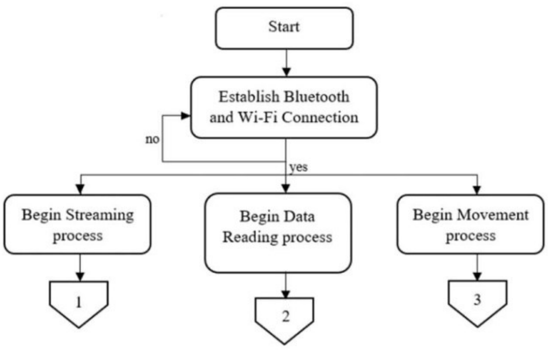

Fig. 4. Process flow at Raspberry Pi

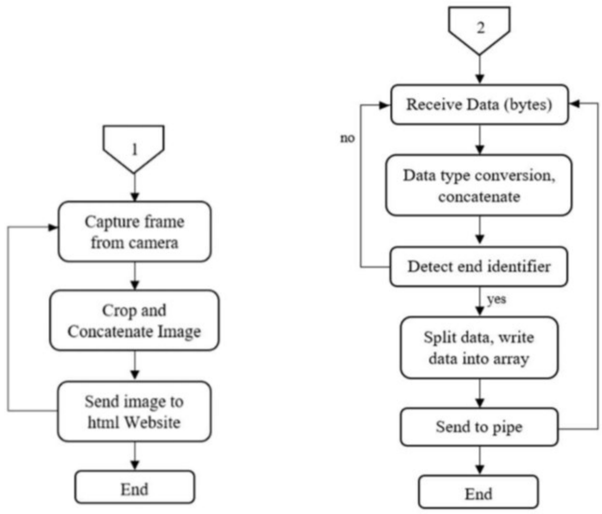

Fig. 5(a). Flowchart extension

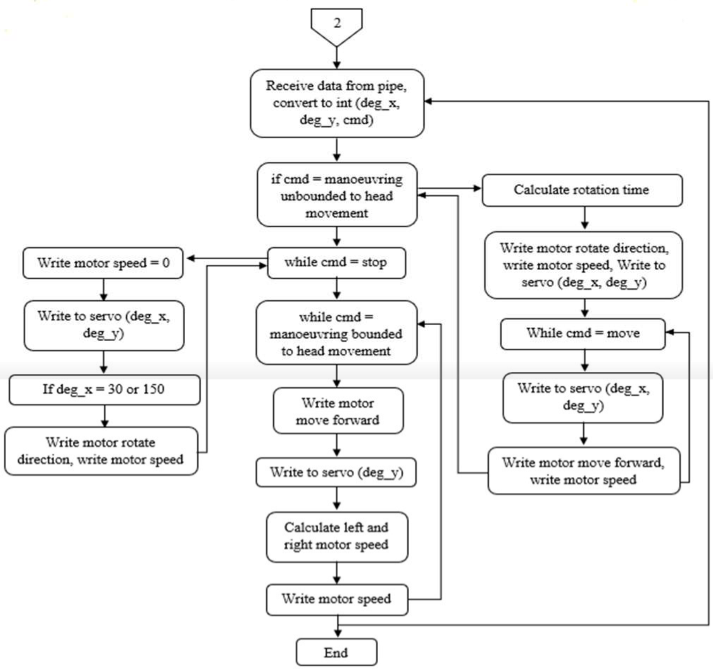

Fig. 5(b). Flowchart extension

E. Hardware Setup

The Arduino Maker Uno is connected to the CJMCU-055 Attitude Sensor, the Bluetooth module, and the wired controller, as shown in Fig. 6. For the CJMCU-055 Attitude Sensor, the TX and RX are connected to A4 and A5 analog input pins on the Arduino, VCC and GND to that corresponding to the microcontroller, whilst COM3 is connected to ground. For the Bluetooth module, RX and TX are connected to pin12 and 13, and VCC and GND to that of the Maker Uno. For the wired controller, the inputs for the four buttons are connected to pin 4 to 7, and VCC and GND similarly to their respective places.

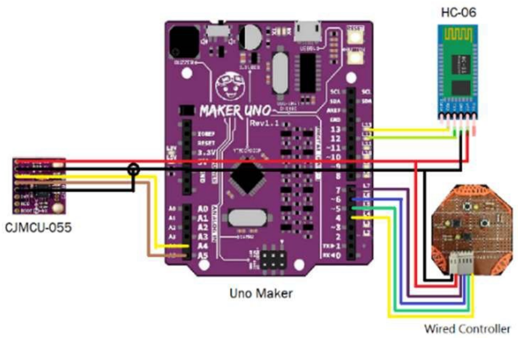

Fig. 6. Schematic diagram of circuit for Arduino Maker Uno

On the Raspberry Pi, the board is connected to 2 Servo Motors and 2 Motors with a l298n motor driver as the interface. For the Servo Motors, VCC is connected to pin 2 and 4 (5V power), grounds to pin 3 and 6 (Ground), and control pins to pin 32 and 33 (GPIO 12 and 13) which corresponds to PWM0 and PWM1. For the l298n motor driver, EN1 is connected to pin (GPIO 18(PCM_CLK), IN1 to pin 31 (GPIO 6), IN2 to pin 29 (GPIO 5), IN3 to pin 18 (GPIO 24), IN4 to pin 16 (GPIO 23), and EN2 to pin 35 (GPIO 19(PCM_FS)). The motors are connected to the l298n motor driver as depicted in Fig. 7.

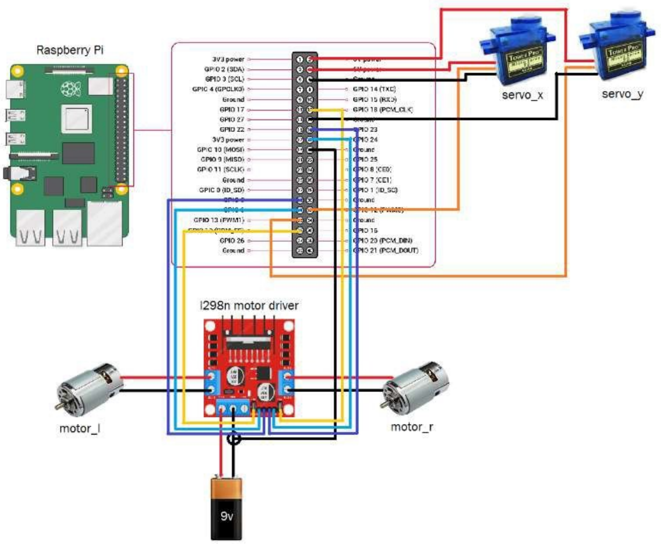

Fig. 7. Schematic diagram of circuit for Raspberry Pi

III. Results and Discussion

A. Resultant of Motor Movement

For mode ‘maneuvering unbounded to head movement’, when the command is issued, robot rotates to angle the user’s head is facing, and moves forward from there until another command is issued, as illustrated in Fig. 8. Rotation time to reach the angle is roughly tuned as the motor used are not encoder motors. The equation of rotation time determined as $(deg\_x \ -90)/180$ upon tuning.

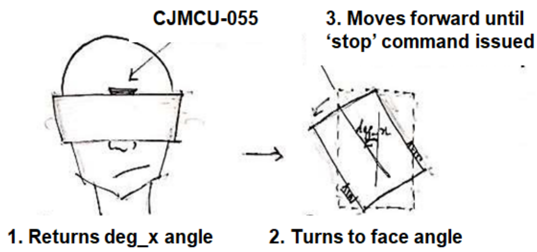

Fig. 8. Illustration of resultant movement from mode: ‘maneuvering unbounded to head movement’.

For mode ‘stop’, the motor will stop, unless user turns head to 30° or 150°, which the robot will continuously spin left or right at the same location, as illustrated in Fig. 9.

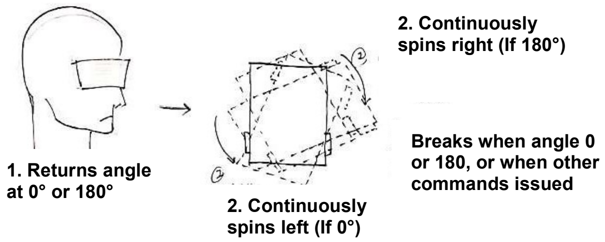

Fig. 9. Illustration of resultant Movement from mode: ‘stop’.

For mode ‘maneuvering bounded to head movement’, the left and right motor speeds up or slows down according to the direction the user’s head is facing, following the simple linear relationship as depicted in Fig. 10. The designated max duty cycle is set at 45% whilst the minimum duty cycle to move the robot is found out to be at 20%, as illustrated in Fig. 11.

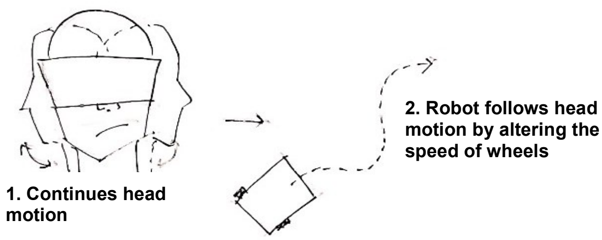

Fig. 10. Illustration of resultant Movement from mode, ‘maneuvering bounded to head movement’.

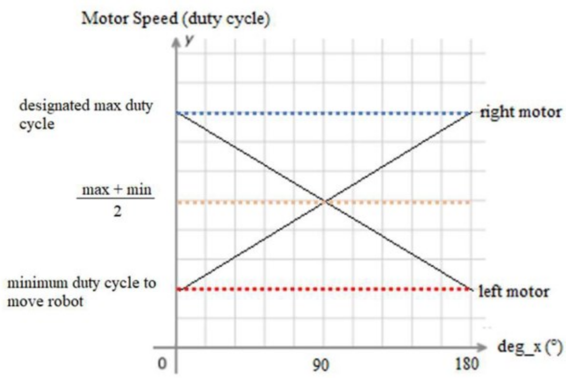

Fig. 11. Graph of motor speed against deg_x

B. Servo Movement

The mechanism for the head movement mimicking system is created using the 3D pen, as shown in Fig. 12. The mechanism can follow head movement almost instantaneously with very minor jitters. The servo movement actively follows head movement, except in during mode 'maneuvering bounded to head movement' (as illustrated in Fig, 13), where it is discovered that it is constantly writing to all four PWM pins which cause massive lag spikes to all servos and motors. Hence, it is set to default position during the operation.

Fig. 12. Mechanism for mimicking head movement created with the 3D pen

Using the set_servo_pulsewidth in pigpio library, the relationship between servo angle and pulse width value goes by a linear relationship, as shown in Figure 14. Hence, the equation for pulse width value at servo x is $100/9 * deg\_x + 500$ while for servo y is $100/9 * deg\_y + 500$.

Fig. 13. Resultant of the head position and the position of the mechanism

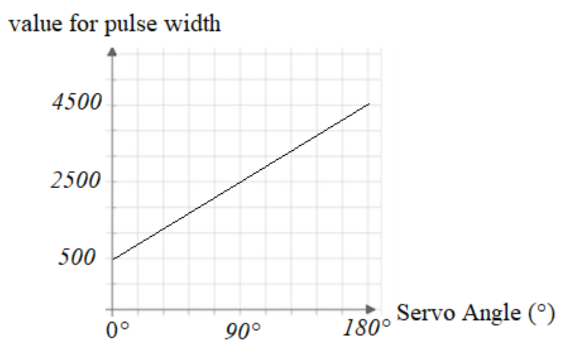

Fig. 14. Graph of servo angle versus pulse width value according to the pigpio library

C. Live Streaming

The live video footage can be displayed on the phone via a full screen browser at 40 fps (example in Fig. 15). Any 'fps' higher than 40 will result in the hue of the image which becoming purple. When viewed through the VR headset, the image is mediocre due to the low image pixel of the Raspberry Pi Camera Module and the occasional jitters caused by running multiple processes in parallel, as illustrated in Fig. 16.

Fig. 15. Footage from live streaming

Fig. 16. Image screenshot from Huawei Nova 3e

CONCLUSION

A system capable of simple teleoperation in the form of navigation is successfully created with all three modules working consecutively. The application of such telepresence robots can serve as substitutes for humans to navigate in hazardous environments with the prerequisite of having good enough connection between the master and slave.

Associating head motion to robot navigation does provide a rather unique experience to users, and further developments to expand this mode of navigation to, for instance, flying drones, may be feasible. With regards to user experience in this study, dizziness is a main issue due to the poor resolution from the 5 Megapixel camera and the delay of the streaming which is already sufficient to induce minor disorientation. Aside from that, having only footage from a single camera is insufficient in navigation through a VR setup due to the lack of depth perception.

To improve the project, precision in various aspects has to be reworked. A better approach should be used in the creation of hardware, for instance, using CAD software to design the head motion mimicking system and creating it using a 3D printer instead of a 3D pen which accuracies in dimensions will be hampered. To enhance user experience, a better range of components should be opted for, for instance, using encoders motors so that the PID control loop can be applied for better speed and angular control, and using a microcontroller with better processing capabilities. At last, dual-camera stereoscopic setup ought to be implemented with cameras of higher resolution on a viable microcontroller for depth perception and better user experience.

REFERENCES

[1] Janice, "What Is Telepresence Technology?", 2017. Retrieved from: https://www.eztalks.com/video-conference/what-is-telepresence- technology.html

[2] Malczewski. K.,"The Rise of Telepresence Robots for Business and Beyond", 2014. Retrieved from: https://www.factor- tech.com/roundup/this-week-facial-recognition-used-to-capture-fugitive- spacex-commits-to-city-to-city-rocket-travel-and-uk-reveals-it-launched-a-cyber-attack-on-islamic-state/

[3] Kerruish, E. "Lessons on telepresence from the Mars explorer Rovers: Merleau-Ponty and the open perceptual circuit. Culture", Theory and Critique, 1-15, 2019.

[4] Mould, R. F., "Chernobyl record: the definitive history of the Chernobyl catastrophe", CRC Press, 2000.

[5] Kratz, S., Vaughan, J., Mizutani, R., & Kimber, D. „Evaluating stereoscopic video with head tracking for immersive teleoperation of mobile telepresence robots". In Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts (pp. 43-44). ACM, 2015.

[6] Wen, M. C., Yang, C. H., Tsai, M. H., & Kang, S. C., "Teleyes: A telepresence system based on stereoscopic vision and head motion tracking". Automation in Construction, 89, 199-213, 2018.

[7] Pinter, M., Lai, F., Sanchez, D. S., Ballantyne, J., Roe, D. B., Wang, Y., ... & Wong, C. W., U.S. Patent No. 9,776,327. Washington, DC: U.S. Patent and Trademark Office, 2017.

[8] Lester, D. F., Hodges, K. V., & Anderson, R. C., "Exploration telepresence: A strategy for optimizing scientific research at remote space destinations". Science Robotics, 2(7), Art-No, 2017.

[9] Nakashima, R., & Shioiri, S., "Why do we move our head to look at an object in our peripheral region? Lateral viewing interferes with attentive search". PloS one, 9(3), e92284, 2014.

[10] Sadik, M. J., & Lam, M. C., „Stereoscopic Vision Mobile Augmented Reality System Architecture in Assembly Tasks". J Eng Appl Sci, 12, 2098-2105, 2017.

[11] Raja, D. DC Motor Control with Raspberry-Pi, 2016 Retrieved from https://circuitdigest.com/microcontroller-projects/controlling-dc-motor-using-raspberry-pi.