Robot Gesture Control Using Online Feedback Data with Multi-Tracking Capture System

Khang Hoang Vinh Nguyen[1], Tri Bien Minh[2], Van Chi Le[3] and Phu Xuan Do*[4]

[1]Mechatronics and Sensor Systems Technology, Vietnamese-German University, Vietnam
`khang.nhv@vgu.edu.vn`
[2]Electrical and Computer Engineering, Vietnamese-German University, Vietnam
`tri.bm@vgu.edu.vn`
[3]Mechatronics and Sensor Systems Technology, Vietnamese-German University, Vietnam
`levanchi1991@gmail.com`
[4]Mechatronics and Sensor Systems Technology, Vietnamese-German University, Vietnam
(Corresponding author)
`phu.dx@vgu.edu.vn`

Abstract. This study uses a Vicon system, a high-speed accurate marker-based motion capture system aiming to develop a wireless low-latency gesture control method that requires no hardware development and no on-board power source. A novel control is designed to track a rigid-body object's motion in the workspace and translate it to robot control signals. The system has five main components: a host PC with a Vicon software named Tracker, a set of eight Vicon Vantage V8 infrared cameras, a client PC that receives Tracker's broadcasted data over the Ethernet, a robot model named youBot provided by KUKA, and a rigid-body object that will act as the gesture controller. Using an Application Programming Interface (API) called DataStream, it is successfully obtained the pose (position and orientation) of the rigid-body object and processed it to remotely control the KUKA youBot robot with high accuracy and no recognizable latency. Although it is similar to a combination of accelerometer-based and gyroscope-based controllers, the gesture controller requires no specific hardware besides a set of retro-reflective markers for tracking purposes. The proposed control method proves beneficial during the robotics prototyping phase, where different control mechanisms can be replicated and tested within the motion capture system's range before deciding the best one for the target application. Furthermore, the robot's odometry can also be tracked during testing, making a motion capture system a valuable tool for the evaluation of motion control.

Keywords: Vicon; motion capture; wireless; gesture control; robotics; ROS.

1 Introduction

In the robotics development, developers can use different devices to input control signals, such as a keyboard, control panel, or joystick. Wireless devices are more favorable and commonly connect via Bluetooth. Although the devices above are easy to use, they are built mainly with push buttons and potentiometers, thus limiting the input options. One of the new input methods is using gesture-related signals, for example, hand gestures, detected by either using sensors (such as accelerometer [1], flex sensor [2], force sensor [3], or a combination of all of those sensors [4]) or image processing. Nowadays, sensors are getting smaller and taking less space on miniature circuit boards. Hence, the sensors can be easily fit on a glove [4] or a wristband [3], together with wireless units and batteries. The sensor output signals will then be programmed to control the robot's actuators, for example, tilting the wrist (alters gyroscope's output) controls steering, and bending the finger (alters flex sensor's output) accelerates the robot [4]. Alternatively, gestures can be recognized using image processing techniques combined with machine learning algorithms to improve accuracy and reliability [5].

However, wireless sensor systems require an on-board power source and connectivity programming, while image processing techniques have a quite low accuracy regarding position tracking [6]. This paper would like to introduce another approach to robot gesture control that requires no addition hardware development, little programming effort, and is easy to experiment. Using a marker-based motion capture system provided by Vicon, which is one of the most popular suppliers in the motion tracking solution market, an object's pose (position and orientation in three-dimensional coordinate system (3D)) can be tracked and processed to generate control signals to drive a KUKA youBot robot to move omnidirectionally with the overall latency of 0.3 s. However, the controller must remain within the capture range of the motion capture system in order to be effective. Therefore, the proposed controller would be more useful during the prototyping and testing phases rather than becoming an actual commercial product.

2 Proposed multi-tracking robot control system

The concept of the proposed controller utilizing motion capture data can be represented in the signal chain as illustrated in Fig. 1, where the system consists of:

- Eight calibrated Vicon Vantage V8 infrared cameras, which is one of Vicon's flagship models, with resolution of 8 Megapixel and frame rate is set at 100 Hz (although the frame rate can be set to a higher value (up to 2 kHz), it is not recommended to do so because the field of view will be significantly decreased)
- A host PC with Vicon Tracker installed (Tracker is Vicon's user interface for monitoring system status and broadcasting capture data over Ethernet [7])
- A client PC to handle the programming and data processing tasks, which is an Acer TravelMate P259-M with Intel Core i5-6200U processor, 8 Gigabytes of RAM, 500 Gigabytes solid state disk drive and uses Ubuntu 16.04 operating system
- A Wi-Fi router for device communications over Ethernet. The router model used in this study is Netis DL4304
- KUKA youBot robot model. The robot communicates with the client PC using ROS messages and it will be discussed further in section 2.2

- A rigid-body object with retro-reflective markers acting as the gesture controller. By rotating the object about the axes, the controller will drive the robot to move at different velocities as shown in Fig. 2

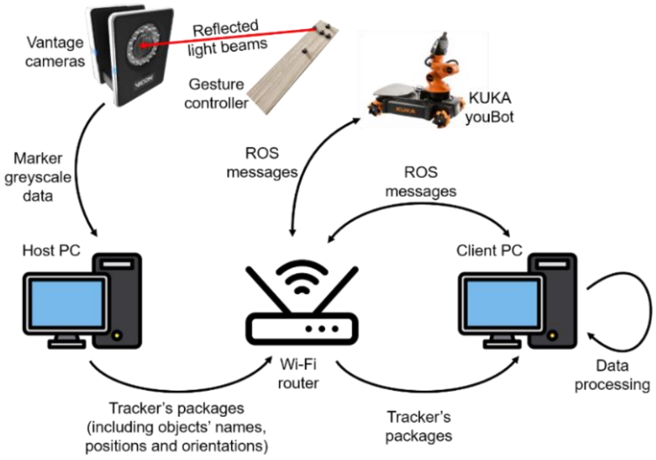

Fig. 1. Proposed multi-track system signal chain

In addition, Vicon also provides a free API called DataStream which enables client PC to obtain the capture data via Ethernet and develop third-party applications. In this study, the capture data includes the motion of the controller, which will be used as control signals, and the movement of the robot in order to verify the system response. The robot is programmed to move at different linear and angular velocities as described in Table 1, where a, b and c are the velocity factors and θ, ψ, ϕ are the Euler angles that will be described further in section 3.1.

Table 1. List of gestures

Controller's rotation axes	Angles	Robot's velocities
x	θ	$v_y = \theta \times a \ (m/s)$
y	ψ	$v_x = \psi \times b \ (m/s)$
z	ϕ	$\omega = \phi \times c \ (rad/s)$

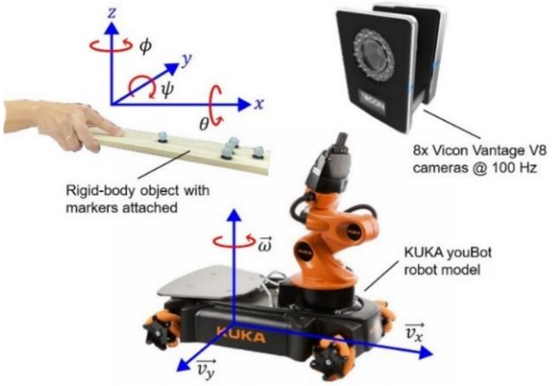

Fig. 2. Experiment setup with the multi-tracking system and robot

2.1 Vicon motion capture system

The motion capture system of this study is a marker-based system; therefore, it requires the markers coated with retro-reflective material for tracking purposes. The makers used in this study have the diameter of 14 mm, which is suitable for small-scale robotics development and clinical analysis.

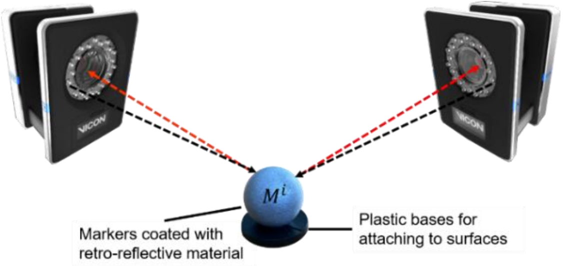

Fig. 3. Retro-reflective marker reflects light beams to Vantage cameras

To track the marker M^i in 3D (where i is the number of markers visible in the workspace), the Vantage cameras emit infrared light to the workspace using a strobe ring equipped on each unit as shown in Fig.3. The cameras are adjusted to receive only the reflected light from the markers only. Fig. 4 visualizes the positions of the cameras as well as the capture range of the system and the objects such as the controller and the robot are required to be visible in this field in order to be tracked properly. When the markers are attached to the objects, the centroid of each marker can be estimated as a position in the 3D using the multi-view reconstruction methods [8]. A Vicon system is capable of achieving errors lower than 0.02 mm in static analyses and approximately 0.2 mm in high-speed dynamic analyses with proper camera calibration [9]. Further details of the calibration procedure can be found on the official webpage [10]. Once the system goes live, the tracked pose data can be processed locally or broadcast over Ethernet for client PCs to develop custom applications using Vicon DataStream API. Although the API supports various programming languages such as C, C++, dotNET, Python [11], this study implements mainly robotics libraries written in Python for programming, which suits the simplicity of the proposed controller where new control mechanisms can be quickly developed and experimented.

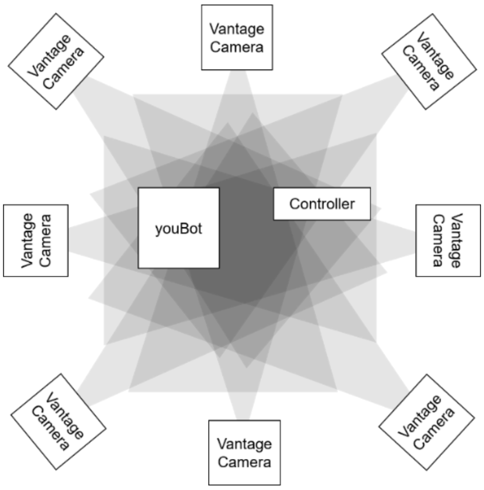

Fig. 4. Camera positionings and objects in the workspace

2.2 KUKA youBot

As mentioned above, the target robot of this study is KUKA youBot, a mobile robotic arm developed by KUKA for research and education. The robot has five degrees of freedom (5-DoF) arm and a two-finger linear gripper. The base has four Mecanum wheels that allow youBot to move omnidirectionally as shown in Fig. 4. The robot also has an onboard PC and it is capable of executing stored programs as well as being remotely controlled by another computer [12].

Fig. 5. KUKA youBot system

The robot can be programmed using open-source robotics middleware such as Robot Operating System (ROS), Open Robot Control Software (Orocos), or using Robotics Module of LabVIEW. This paper will mainly focus on developing the robot using ROS, as its existing packages for youBot, which are widely available, community-supported and proofed to be more reliable in comparison to the other development platforms. ROS is installed on the client PC and the compatible ROS version is called Kinetic Kame.

The client PC communicates with youBot by trading ROS messages [13]. The content of the messages can be sensor data, motor control commands, state information, actuator commands, etc. Therefore, the motion capture data package of the Vicon system cannot be sent directly to the client PC and requires additional processing. Besides installing ROS, there is a package called `vicon_bridge` is required to be included in the build system for translating the messages received from Vicon Tracker to typical ROS message formats prior to sending commands to the KUKA youBot robot [14].

3 Controller design

Fig. 6 shows a simple 400mm*90mm*9mm wooden piece with five markers attached that is used as a gesture controller as well as its representation in the Tracker interface. The markers are linked together and form an object with a local coordinate system, where its origin can be automatically placed by Tracker or manually defined by user. In order to teleoperate the robot, the controller rotates about the axes and the Euler angle values will determine the velocities as described in Table 1. The Euler angles will be discussed in the following subsection.

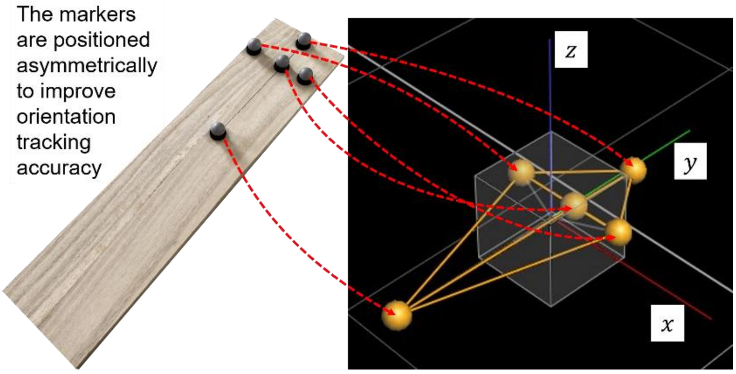

Fig. 6. The gesture controller

3.1 Controller kinematics

Vicon DataStream API represents the pose of an object in each frame as a translation vector and rotation matrix. When an object i is detected in the workspace, its position and orientation in the global coordinate system $\{0\}$ are represented as follows:

$$^{0}t_i = [t_x \quad t_y \quad t_z] \tag{1}$$

$$^{0}R_i = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2}$$

Where, $^{0}t_i$ is the translation vector of object i from the global coordinate and $^{0}R_i$ is the object's rotation matrix from the global coordinate. The transformation of an object i from the local coordinate to the global coordinate is:

$$^{0}T_i = \begin{bmatrix} ^{0}R_i & (^{0}t_i)^T \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

However, it is more convenient to have the control signal only composed of the translation vector and the rotation is described as the zyx sequenced three-angle representation, or Euler angles [15]. The controller's rotation angle about the z-axis can be calculated from the obtained transformation matrix (3) as:

$$\phi = atan2(r_{21}, r_{11}) \tag{4}$$

The rotation angle about the y and x-axes can be determined as follows:

$$\psi = atan2(\sin\phi * r_{13} - \cos\phi * r_{23}, \cos\phi * r_{22} - \sin\phi * r_{12}) \tag{5}$$

$$\theta = atan2(-r_{31}, \cos\phi * r_{11} + \sin\phi * r_{21}) \tag{6}$$

The calculated Euler angles are later used for determining the robot's linear and angular velocities as previously described in Table 1 and the command messages will be sent to the robot via Ethernet to drive it accordingly.

4 Evaluation

In order to verify the robot's movement, an additional set of five markers was attached to the robot and defined in Vicon Tracker similar to the process of defining of the controller in Tracker mentioned in section 3. Tracker is capable of accurately tracking multiple rigid objects simultaneously. Therefore, introducing a new object to the system does not affect the accuracy of the gesture controller. From the translation vectors obtained from Vicon DataStream API, the linear velocity along a particular axis can be calculated using the finite difference method as:

$$v_n = \frac{x_n - x_{n-1}}{t_s} \tag{7}$$

Where, v_n is the linear velocity along an axis (for example x-axis), calculated by dividing the difference in the sampled positions from the global origin with the sampling interval t_s. Identically, the angular velocity (for example rotation about z-axis) can be determined as:

$$\omega_n = \frac{\phi_n - \phi_{n-1}}{t_s} \tag{8}$$

Figs. (8-10) illustrate the comparison between the controller's rotations and the robot's velocities during one identical record: Fig. 8 represents the robot is driven back and forth, moved sideways in Fig. 9 and turned around in Fig. 10. It can be seen that the robot moves as programmed, where the velocity factors are $a = 0.6, b = 0.6, c = 0.4$ (see Table 1). There are several mismatches between the signals, for example around the 50th second in Fig. 8 and Fig. 9, but it is not because of the incorrect robot's response. Due to the rotation about the z-axis, the displacement of the robot's base consequently changes the velocities along x and y axes, thus, it is not an error of the system.

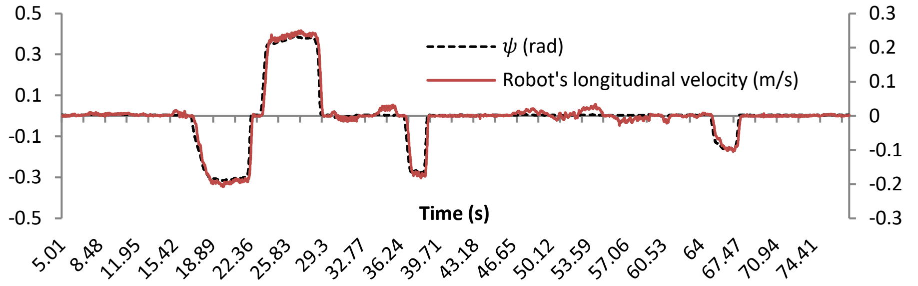

Fig. 7. Comparison between controller's angle value ψ and robot's movement

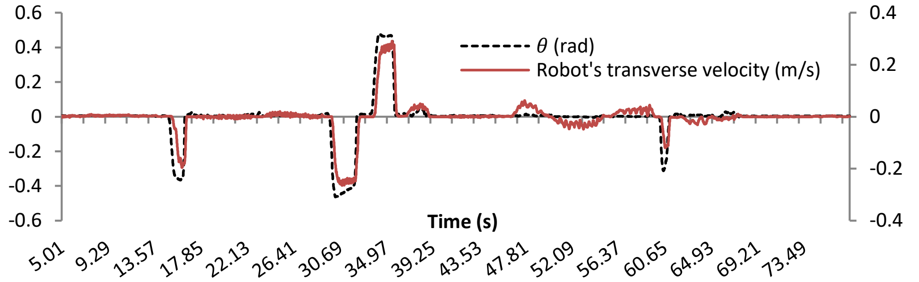

Fig. 8. Comparison between controller's angle value θ and robot's movement

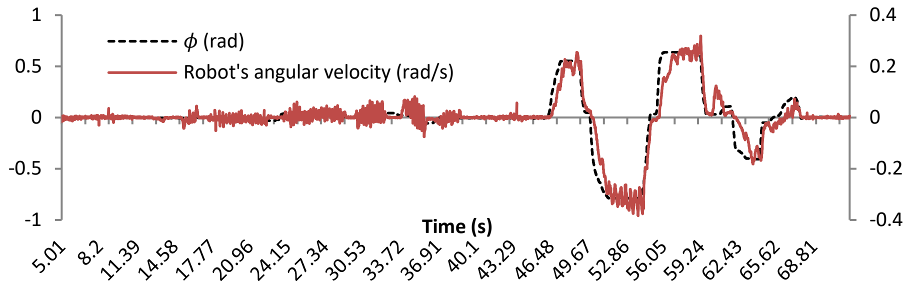

Fig. 9. Comparison between controller's angle value ϕ and robot's movement

By inspecting the measurements above in a small duration of 4 s, the overall latency of the system can be determined as approximately 0.3 s, as shown in Fig. 11. Although this latency does not suitable for hard real-time applications where delays are fatal, as ROS is not capable of doing so and ROS 2 is in development to improve the real-time performance [16], the proposed system still delivers an acceptable user experience, since human's perception only has the typical reaction time of 0.2 s to 0.3 s [17].

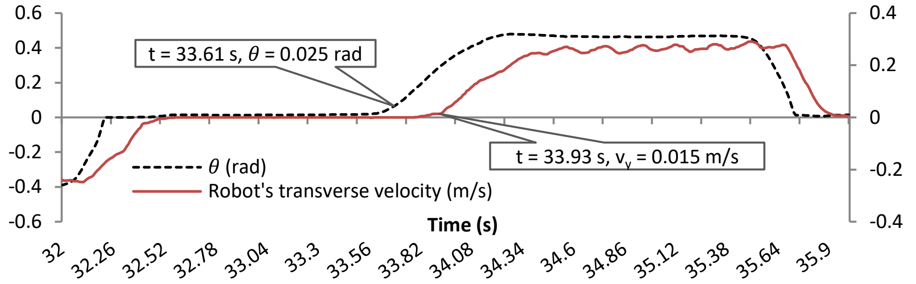

Fig. 10. Overall system latency

5 Conclusion

This study has verified that the motion capture data of an object in Vicon Tracker can be retrieved via Ethernet and processed as wireless control signals for robotics applications. Through evaluation, the proposed controller was capable of controlling the KUKA youBot robot model to move as programmed with overall latency of 0.3 s. Furthermore, the proposed system has been proven to be a suitable platform for new controller designs as it requires no complex hardware development and takes little programing effort. Thus, developers can quickly experiment with different control mechanisms prior to decide which one is the best for the target application and proceed to design corresponding hardware.

References

1. D. O. Anderez, L. P. Dos Santos, A. Lotfi and S. W. Yahaya, "Accelerometer-based Hand Gesture Recognition for Human-Robot Interaction," *2019 IEEE Symposium Series on Computational Intelligence (SSCI),* pp. 1402-1406, 2019.
2. A. K. Panda, R. Chakravarty and S. Moulik, "Hand Gesture Recognition using Flex Sensor and Machine Learning Algorithms," *2020 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES),* pp. 449-453, 2021.
3. M. Anvaripour and M. Saif, "Hand gesture recognition using force myography of the forearm activities and optimized features," *2018 IEEE International Conference on Industrial Technology (ICIT),* pp. 187-192, 2018.
4. A. Dekate, A. Kamal and K. Surekha, "Magic Glove - wireless hand gesture hardware controller," *2014 International Conference on Electronics and Communication Systems (ICECS),* pp. 1-4, 2014.
5. J. L. Raheja, R. Shyam, U. Kumar and P. B. Prasad, "Real-Time Robotic Hand Control Using Hand Gestures," in *Proceedings of the 2010 Second International Conference on Machine Learning and Computing,* USA, 2010.
6. M. Oudah, A. Al-Naji and J. Chahl, "Hand Gesture Recognition Based on Computer Vision: A Review of Techniques," *Journal of Imaging,* p. 73, 2020.
7. "Tracker: Delivering precise real-world data: Motion Capture Software," Vicon, [Online]. Available: https://www.vicon.com/software/tracker/. [Accessed 19 09 2022].
8. R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge: Cambridge University Press, 2004.
9. P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur and X. Savatier, "A Study of Vicon System Positioning Performance," *Sensors,* p. 1591, 2017.
10. "Vicon Vantage Cameras Documentation," Vicon, [Online]. Available: https://docs.vicon.com/display/Vantage/Vicon+Vantage+camera+range. [Accessed 19 09 2022].
11. "DataStream SDK Documentation," Vicon, [Online]. Available: https://docs.vicon.com/display/DSSDK111. [Accessed 19 09 2022].
12. "youBot User Manual," KUKA, [Online]. Available: http://ftp.youbot-store.com/manuals/KUKA-youBot_UserManual.pdf. [Accessed 19 09 2022].
13. A. Mahtani, L. Sanchez, E. Fernandez and A. Martinez, Effective Robotics Programming with ROS - Third Edition, Packt Publishing, 2016.
14. M. Achtelik, "vicon_bridge ROS package," 2021. [Online]. Available: https://github.com/ethz-asl/vicon_bridge. [Accessed 19 09 2022].
15. B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, Robotics: Modelling, Planning and Control, Springer Publishing Company, Incorporated, 2008.
16. Y. Yang and T. Azumi, "Exploring Real-Time Executor on ROS 2," *2020 IEEE International Conference on Embedded Software and Systems (ICESS),* pp. 1-8, 2020.
17. A. R. Jensen, "Chapter 3 - Reaction Time as a Function of Experimental Conditions," in *Clocking the Mind,* Oxford, Elsevier Science Ltd, 2006, pp. 43-54.