Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-25 (2019) 203–207

An application to simulate and control industrial robot in virtual reality environment integrated with IR stereo camera sensor ⋆

Youshaa Murhij * **Vladimir Serebrenny** **

* *Robotic Systems and Mechatronics Department, Bauman Moscow State Technical University, Moscow, Russia (e-mail: murkhizhyu@student.bmstu.ru)*

** *Robotic Systems and Mechatronics Department, Bauman Moscow State Technical University, Moscow, Russia (e-mail: vsereb@bmstu.ru)*

Abstract: The main goal of this research is to test for potential ways to control KUKA KR10 industrial arm manipulator using Virtual Reality technology and check for the advantages of applying this control methods. The final version of this application aims to achieve this goal by establishing an interaction between the user and the manipulator inside a virtual environment developed using the game engine Unity3D and the HTC VIVE Pro headset for the virtual visualization. By applying this control method, the user does not have to operate on site and instead he can work remotely. In addition to the ability to use off-line programming of the manipulator. The application is designed to simplify the controlling ways by displaying a complete virtual environment where the tridimensional model of the robotic arm can be visualized and programmed according to the real manipulator's parameters and specifications. All the movements and parameters in the virtual environment can be synchronized with the real robot in an on-line or off-line path planning depending on the application or the task. The system integrates a set of virtual reality controllers and Leapmotion sensor as options to allow the user to control and see the robot and its parameters in the virtual environment. As a result of this research, the manipulator moves on the planned trajectory in a smooth way after applying some filtering techniques without losing its accuracy.

Keywords: Virtual Reality; Robot; KUKA; Unity3D; HTC VIVE, Leapmotion.

1. INTRODUCTION

Virtual reality is an interactive interface that provides the user with visually realistic images created by computer graphics with depth buffer. Its techniques are one of the modern technologies that are considered as tools to enhance the user experience. Graphical simulation is an evolving technique being used in robotic systems to generate robot command with no risk and in a cost effective way. Most of these systems are Robotics Teleportation Systems which are widely used in: industry, science, medicine, education, entertainment and military applications.This paper discusses the possibility to develop and implement a virtual reality application to control and simulate a manipulator, which could be considered as a helpful tool to improve robot usage by enabling an interaction with the robot and workspace features.

1.1 Related Work

A project to test the benefits of virtual reality technology by creating a simulator for a Mitsubishi robot (1) was done to decrease the students learning curve using Oculus Rift and Unity. A complex control system of a teleported mobile robot using virtual reality assisted by a human operator was achieved in a cost-effective way (2). In (3) they described how consumer grade Virtual Reality headsets and hand tracking hardware can be used to naturally teleoperate robots to perform complex tasks. Virtual reality technology is used to improve human-robot interface (4) and the system included teaching task, graphical simulation and real task execution using a data glove attached to a sensor tracker. A simulation and manipulation of teleportation system for remote control of mobile robot using the Virtual Reality (VR) was presented in (5). They allowed the operator to control and supervise a unicycle type mobile robot.

1.2 Hardware and Software

This research is based on HTC VIVE VR instruments in addition to KUKA KR10 manipulator with the integration of Leapmotion sensor. The main used platforms and programs are Unity3D, Matlab, SteamVR and Autodesk MAYA in addition to Robot drivers that use TCP/IP

⋆ The reported study was funded by RFBR according to the research project 19-01-00767.

2405-8963
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.12.473

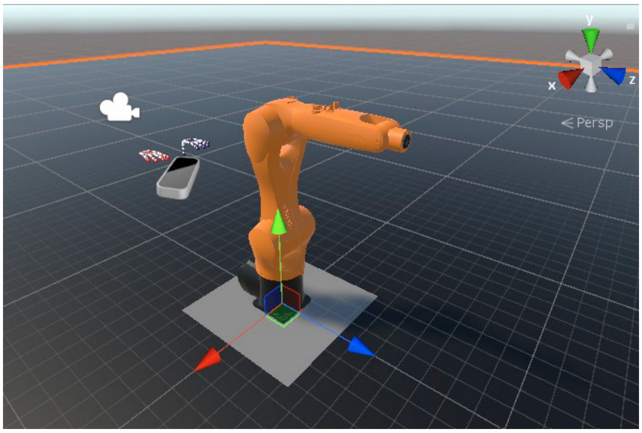

Fig. 1. Virtual model of KUKA KR 10 in Unity3D

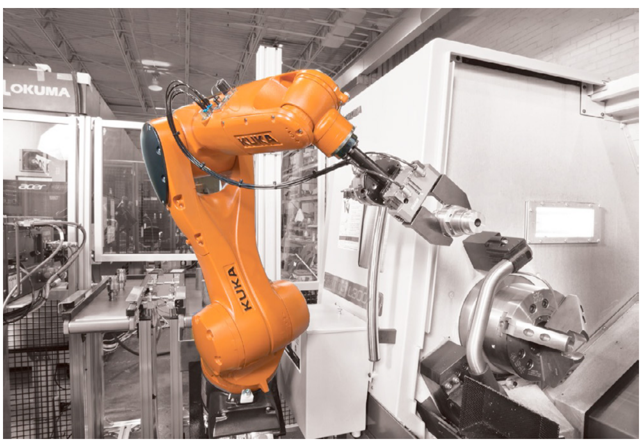

Fig. 2. Real KUKA KR 10 robot in a production line

socket communication protocol for exchanging data with the real robot.

2. . METHODOLOGY AND MAIN PROCEDURE

2.1 Manipulator 3D Modelling

A complete 3d model of the manipulator to be controlled (KUKA KR10 in our case) with real-world details was assembled so it can be imported easily to the project scene in Unity3D as *.fbx* file extension or a *Prefab*. This task has been done using external CAD and design program (Autodesk MAYA) and the model is imported as a tree of connected joints and links with constrains between them. So that each part is a child of the previous part starting from the base part. See Fig. 1 and Fig. 2.

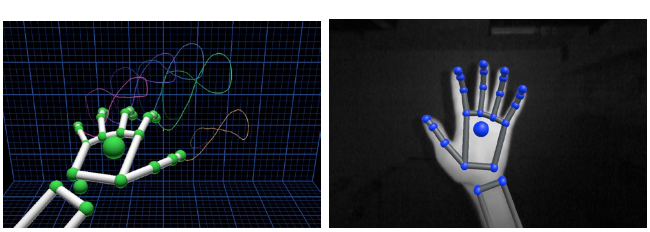

Fig. 3. Extracting Palm's position from Leapmotion's Raw Data

2.2 Reading External Sensors

In one hand, using Leapmotion SDK allows us to read the gestures of the hands and extract all the position and orientation data of them. In our case, we are interested in the 3D coordinates of the hand's palm (Center of mass) as shown by a big blue sphere in Fig. 3 where the right image represents a gray photo from one the cameras of Leapmotion sensor and the left image represents the extracted data from the sensor. In the other hand we can extract the headset sensors data and the controllers input of HTC VIVE instruments to get enough parameters as the target position and orientation of the end effector of the manipulator, in addition to the 6DOF values of the VR camera in the scene inside Unity environment.

2.3 Integration and programming in Unity3D

All the Leapmotion and HTC VIVE files can be imported as *prefabs* to the Unity environment. Once in Unity, each part of the robot was assigned to a different Game Object. Finally, the robot was constructed by embedding each Game Object in a hierarchical way according to the hierarchical order of the DOF's links in the real-world robot. In other words, to make each joint of the robot move with the respect with other joints, every joint should be a parent to the joint which it carries so, we can apply the direct kinematics using the total transformation matrix (Transition + Rotation) of the manipulator as shown below:

$$T_0{}^6 = \prod_{i=0}^{5} T_i{}^{i+1} \tag{1}$$

where,
$T_0{}^6$: The total transformation matrix of the manipulator.
T_i^{i+1} : The transformation matrix between link i and $i+1$.
And in order to control and update each joint's angle in every frame we should attach a script file (C# code) to each joint so we can write angle updating commands in the scope of the update() function. The mainly used function is the transformation function (build-in function in unity) which can update the position, orientation and scale of every joint in the scene.

2.4 Robotic Movement Programming

We need to program the movement of every joint according to the manufacturer specifications and this required to change the system of angles to avoid the gimbal lock around the X axis (in the second, third and fifth joints). So, applying inverse kinematics on the input data will give us the generalized parameters of the Virtual robot and by applying forward kinematics we can mirror the virtual robot movements to the real one in the lab.

2.5 Data Control

As shown in Fig. 4, the input to the scene is the input data from Leapmotion sensor and from the left and right controllers of the HTC VIVE instruments. The user has the option to choose the input or the control method (either using hands by enabling the Leapmotion sensor or

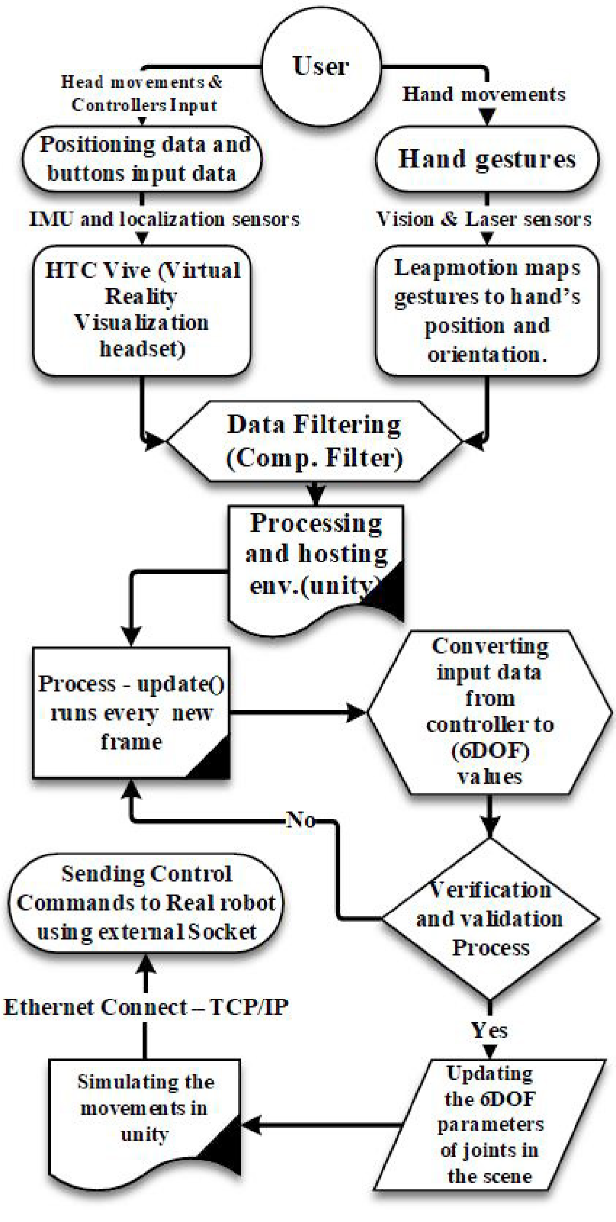

Fig. 4. General Procedure and Data-Flow

using the HTC VR controllers as input devices). The left controller's data is used to control the position and the orientation of the end-effector of the robot by receiving the 6DOF data of the controller and mapping them to the local system coordinates of the manipulator in the VR environment.

Using inverse kinematics, we can compute the generalized parameters of each joint in the robot with respect to the actual parameters (the coordinates of its end-effector). The last step is to update the parameters of the robot in the virtual environment and send the generalized parameters to the actual robot using an Ethernet connection (TCP/IP). The right controller is used to control the navigation and the movements of the main camera in the 3D VR environment. The orientation of the main camera is mapped to the orientation of the HTC VIVE headset which give a perfect VR experience while controlling the manipulator.

3. FILTERING DATA FROM LEAPMOTION AND VIVE CONTROLLERS

The data samples are taken once per frame in the main Unity procedure. So, the sampling rate is quite high which could cause some problems like latency and flickering in the 3D virtual robot and affects the smoothness of the movements in real world application. The sensors record the positions, orientations of the user's hand (In case of using Leapmotion) or the position of controllers (in case of HTC VIVE Controllers). These data are used to generate the robot's parameters which are used to define the dynamic behaviors of the robot. So, a data filtering process is required to reduce the wide changes of data values and eliminate the outliers. A Low-Pass complementary filter has been added to the input data in order to smooth the sharp changes in positioning data and to enhance the trajectory of the manipulators end-effector. The formula of complementary filter is given in this equation:

$$X_{desired}[n] = (1-\alpha).X_{actual}[n] + \alpha.X_{desired}[n-1] \quad (2)$$

Where:
$X_{desired}[n]$: is the current target position of the end-effector. $X_{actual}[n]$: is the current position of the user's hand or the current position of the VR controllers.
α: Filter coefficient. It controls how much you want the output to depend on the current value or a new value that arrives. Both the alpha's have to be the same. α is usually > 0.5 using the definitions above. Figures from 5 to 8 represent the actual path of the manipulator's end-effector without and with applying data filtering method. The first graph shows the 3D trajectories and the others show the values on each axis.

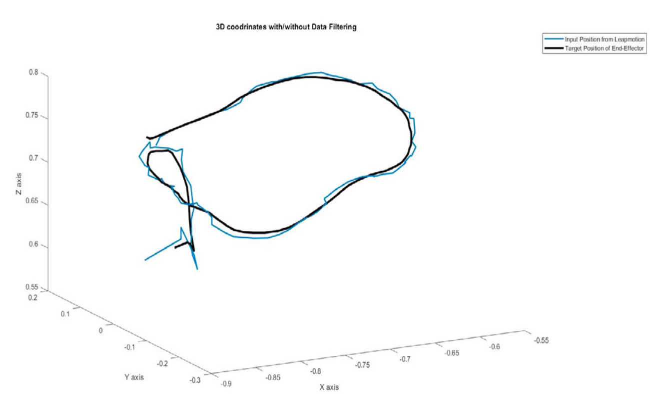

Fig. 5. End-Effector 3D coordinates with/without Data Filtering

Other methods were tested like taking the average of the recent n coordinates and updating the robot's parameters based on the resulting values. This method gives good results when the hands or the controllers are not moving (slightly shaking in space). The formula of this method:

$$X_{desired}[i] = \sum_{k=i-n}^{i} X_{actual}[k]/n \quad (3)$$

Where:
n: represents the number of samples that were averaged

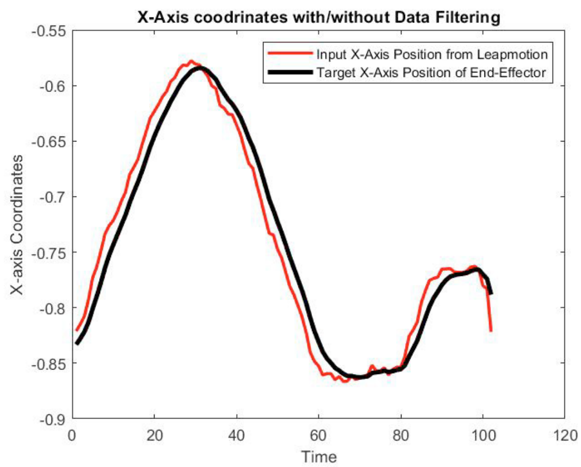

Fig. 6. X-Axis Coordinates

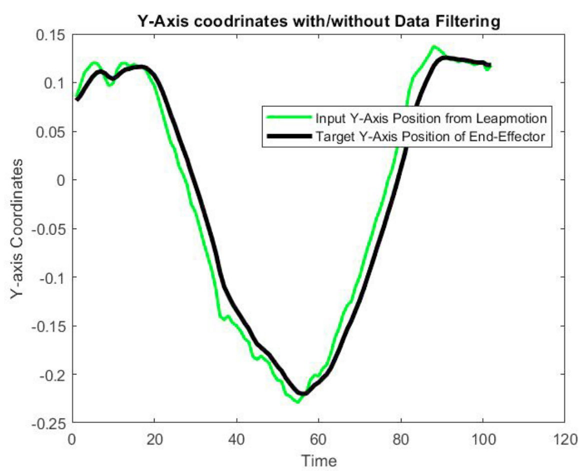

Fig. 7. Y-Axis Coordinates

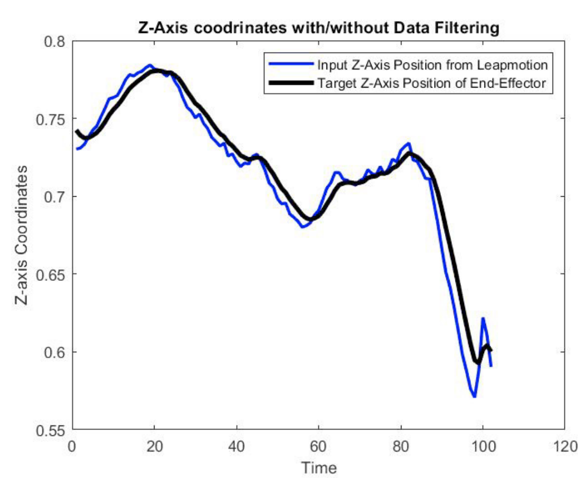

Fig. 8. Z-Axis Coordinates

to estimate the current position of the end-effector. As there are two VIVE controllers, one of them is associated to the real time tracking mission and the other one to maintain a trajectory like `straight path` (linear line or point-to-point) which required only two points to define the line and `Curving path` that is developed based on the Bézier curve. If the task to be achieved is a teleportation mission, there would be no need to use any kind of data filtering. Because, filtering data in this case would make teleportation system loose its reliability.

4. RESULTS

The completed version of the application that is exported from the Unity environment allows the user to interact with HTC VIVE, the virtual version of the KUKA KR10 Robot and the Leapmotion sensor in a way that provides an online or offline path and motion planning of the KUKA robot. By comparing between the actual generated trajectory which is computed using this system and the trajectory that is done by the input devices (Leapmotion sensor or HTC VIVE controllers), We can compute the reliability of this controlling system. In other words we compare between the virtual and real trajectories which are done by the virtual and real robots. In Fig. 9, where the black trajectory represents the processed input data from Leapmotion and the blue one represents the trajectory of the manipulator, we can notice that the movement of the robot's end-effector is a scaled version of the trajectory generated using the two input methods, which gives the system enough reliability to be used in order to accomplish different tasks.

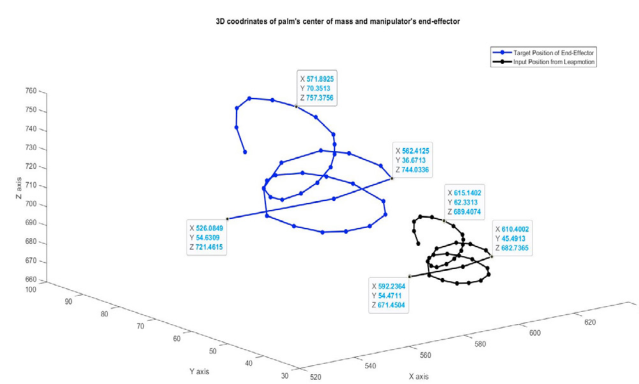

Fig. 9. 3D coordinates of Palm's center of mass and the end-effector in the same system coordinates

5. ABBREVIATIONS AND ACRONYMS

VR: Virtual Reality.
DOF: degree of freedom.
CAD: Computer Aided Design.
TCP/IP: Transmission Control Protocol/Internet Protocol.

6. CONCLUSION

We can conclude that integrating VR technology in specific control systems will enhance the process and provide an interactive experience to the user, in addition to the remotely operating process which is preferred in certain tasks.

ACKNOWLEDGEMENTS

This research was supported by the *Robotic Systems and Mechatronics* Laboratories at Bauman Moscow State Technical University, Russia

REFERENCES

[1] R. Crespo, R. García and S. Quiroz. *"Virtual reality simulator for robotics learning,"* International Conference on Interactive Collaborative and Blended Learning (ICBL), Mexico City, 2015, pp. 61-65. doi: 10.1109/ICBL.2015.7387635. (2015)

[2] Kot, T., & Novák, P. *"Application of virtual reality in teleoperation of the military mobile robotic system TAROS"*. International Journal of Advanced Robotic Systems. (2018)

[3] T. Zhang et al. *"Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation,"* IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, 2018, pp. 1-8.doi: 10.1109/ICRA.2018.8461249. (2018)

[4] Jen, Y., Taha, Z., & Vui, L. *"VR-Based Robot Programming and Simulation System for an Industrial Robot"*. International Journal Of Industrial Engineering: Theory, Applications And Practice, 15(3), 314-322. (2008)

[5] E. Baklouti, M. Jallouli, L. Amouri and N. Ben Amor. *"Remote control of mobile robot through 3D virtual reality environment,"* 2013 International Conference on Individual and Collective Behaviors in Robotics (ICBR), Sousse, pp. 9-14. doi: 10.1109/ICBR.2013.6729276. (2013)