



Virtual training system for robotics applications in industrial processes

Cobo Freire, Erick Bryan

Departamento de Energía y Mecánica

Carrera de Ingeniería Mecatrónica

Artículo académico, previo a la obtención del título de Ingeniera Mecatrónica

Ing, Ph.D Andaluz Ortiz, Víctor Hugo

8 de agosto del 2022



Virtual Training System for Robotic Applications in Industrial Processes

Erick B. Cobo^(✉) and Víctor H. Andaluz

Universidad de Las Fuerzas Armadas ESPE, Sangolquí, Ecuador
{ebcobo, vhandaluz1}@espe.edu.ec

Abstract. This work presents the development of a virtual reality training system applied to industrial robotics and oriented to the teaching learning processes. The system considers a manipulator anthropomorphic industrial robot of 6DOF to perform autonomous object manipulation tasks in two virtual environments. The virtual environments are developed in the graphic engine Unity3D, the environment exchanges information with MATLAB by shared memory to execute position and trajectory control algorithms for the operating end of the manipulator robot. The virtual training system connects with an Android mobile app that considers the following control panel functions: articular coordinate system, velocity configuration and dead man switch. Finally, evaluations of the developed virtual environments, the mobile application and the control algorithm are carried out to guarantee the usability of the virtual training system in the teaching-learning processes.

Keywords: Virtual reality · Manipulator robot · Trajectory and position control · Mobile app

1 Introduction

Industrial processes have evolved over the years due to improvements and development in: productivity, product standardization, reduction of production costs and the application of new technologies related to the Industry 4.0 [1]. The Industry 4.0 includes different digital technologies in conventional industry in order to increase productivity and efficiency using the Internet of things to supervise industrial processes in real time to allow decentralized decision making [2]. The most relevant technologies in the Industry 4.0 are: *i) Cloud Computing*, is an intelligent and programmable tool that allows users to access to different services and repositories to obtain and share large quantities of information [3]; *ii) Big Data*, consists of the analysis of large amounts of data to transform them into information or knowledge [4]; *iii) Robotics*, is an engineering branch that combines knowledge of electronics, mechanics, mathematics and control to build machines that perform repetitive or dangerous work faster and more efficiently than humans [5]; *iv) Immersive Technologies*, these are technologies that allows user interaction with digital recreated worlds, the most important are: Augmented reality “AR” that allows to add graphic information to the real world using technologic devices [6], and

Virtual Reality “VR”, is an immersive technology that provides simulated experiences in 3D environments projected by specialized glasses [7].

It is evident that Augmented Reality and Virtual Reality systems are widely used in industrial applications. Using AR systems in human-robot collaborative object manipulation tasks reduces the risk and time to perform the work [8]. User performance in robot programming improves when AR is used in educational training systems [9]. VR systems facilitates the development of industrial Mechatronics systems by allowing prototype testing analysis [10]. Creating a digital bidirectional robot twin to program the robot in the same HMI as the real one using VR, enables performance and security tests in the simulation before the implementation of the program in the real robot [11]. An automotive assembly environment in VR is a useful tool for collaborative product design because it allows the users to identify and solve problems in the process in order to decrease manufacturing costs [12]. An application of VR environments in the industry for the ship pipeline system assembly helps the operators to identify the most efficient assembly sequence for the process [13]. The previously described works shows that the AR and VR systems applied to industrial processes generate a big interest in the scientific community because it allows the users to interact with simulated industrial robotics systems in order to decrease operation risks and increase productivity.

Actually, due to Covid-19 lockdown restrictions the students do not have access to the laboratories in their Universities, this affects directly to the teaching-learning processes. Immersive technologies are a viable and accessible option for users to interact with simulated environments. For what is described it is proposed the development of an immersive and interactive system that allows free access to virtual environments to boost the teaching-learning processes. This work considers the development of two virtual environments: *i) Industrial Robotics Virtual Laboratory*, where the user can handle a 6DOF industrial manipulator anthropomorphic robot using a mobile application for Android with wireless connection to the environment that includes the following control panel functions: articular coordinate system, velocity configuration and dead man switch in order to learn and recognize the movement characteristics and restrictions of the manipulator robot; *ii) Industrial Virtual Environment*, that allows the user to implement autonomous control tasks for the robot in industrial environments. This environment considers a palletizing robotic cell that interchanges data by shared memory with the mathematic software MATLAB in order to perform position and trajectory control tasks to manipulate boxes that came from a conveyor belt. Performance tests of the control algorithm will be carried out in this environment and also, users can develop their own control algorithm for the industrial robot in order to carry out performance and functionality tests.

This document is divided in five sections including the Introduction. Section 2 presents the System Structure and the mathematic modeling of the manipulator robot. Section 3 shows the development of the Virtual Training System, the Control Algorithm design and the Robot Control Panel development. Section 4 presents the performance results of the Virtual Training System and the controller developed. Finally, conclusions are presented in Section 5.

2 System Structure

Robotics oriented VR environments are a viable alternative to boost teaching-learning processes in a didactic and entertaining way. Applying Robotics in VR environments allows the users to visualize and understand the characteristics and functionalities of industrial robots without the need to purchase real robots. Using industrial robots in VR reduces the risk of using real robots for inexperienced users and allows the users to do performance and functionality tests in virtual environments that are similar to a real one. Therefore, a Virtual Reality Training System that includes a 6DOF industrial manipulator anthropomorphic robot will be carried out. The industrial manipulator robot to use is the ABB-IRB 140, which can be used for many industrial applications such as: palletizing, welding, assembly, classification, among others [14].

In order to perform the manipulation tasks desired for the industrial manipulator robot, in Fig. 1 is presented the system proposed in this work, which considers a closed loop advanced control algorithm to define the position and trajectory of the operating end of the industrial robot.

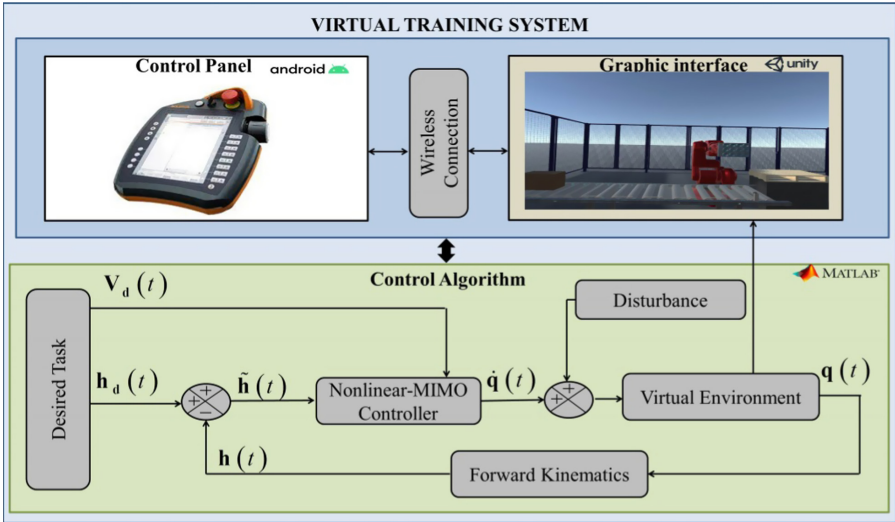


Fig. 1. Proposed system scheme.

The proposed Virtual Training System contains: the graphic interface developed in Unity3D, the control panel mobile app developed in Android Studio and the control algorithm implemented in MATLAB. It is necessary to obtain the mathematic model of the manipulator robot in order to apply it to the control algorithm. The controller uses the kinematic model of the robot to define the articular movements. The kinematic model of a manipulator robot defines the position of the end-effector as a function of its joint configuration, or the operational coordinates of the manipulator robot as a function of its generalized coordinates [15].

$$\mathbf{q} \mapsto \mathbf{h} = f(\mathbf{q}).$$

The forward kinematic model is obtained by the Denavit-Hartenberg parameters (DH), which represents the translation and orientation of the manipulator robot by using articular transformation matrices. The complete homogenous transformation matrix ${}^0\mathbf{T}_n$ of the manipulator robot can be calculated as a string of the individual articular transformation matrices ${}^{n-1}\mathbf{A}_n$ [16]

$${}^0\mathbf{T}_n = {}^0\mathbf{A}_1\mathbf{A}_2\dots{}^{n-1}\mathbf{A}_n \quad (1)$$

where n represents the amount of DOF of the manipulator robot. The position and orientation vector of the end-effector of the manipulator robot

$$\mathbf{h} = [h_x, h_y, h_z, h_\alpha, h_\beta, h_\gamma]^T \quad (2)$$

is obtained from the homogenous transformation matrix.

The instantaneous kinematic model describes the derivative of the end-effector position of a manipulator robot as a function of the derivative of the robotic arm configuration [17]

$$\dot{\mathbf{h}}(t) = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}(t) \quad (3)$$

where $\dot{\mathbf{h}}(t) \in R^m$ is the velocity vector of the end effector of the manipulator robot, $\dot{\mathbf{q}}(t) \in R^n$ is the vector that contains the joint velocities of the robot, and $\mathbf{J}(\mathbf{q})$ is the Jacobian matrix of the manipulator robot. The Jacobian matrix of a robotic arm defines a linear mapping between the joint velocity vector and the end-effector velocity vector. Because the Jacobian is a function of the joint configuration of the manipulator robot, the configurations where $\mathbf{J} \in R^{m \times n}$ is rank-deficient are called singularities. The singularities of a manipulator robot represent configurations at which the mobility of the robot is reduced and is not possible to impose an arbitrary motion to the end-effector. The singularity causes the existence of infinite solutions in the inverse kinematics problem for the manipulator robot. With small velocities in the operational space, large velocities in the joint space shall appear in the neighborhood of a singularity [18]. Singularities are classified into: Boundary Singularities, appears when the robot is out retracted or out stretched, this singularities can be avoided without driving the manipulator into the boundaries of its reachable workspace; and the Internal Singularities that are caused by the alignment of two or more axes of the robot inside its reachable workspace [17].

3 Training Virtual Environment

In this Section it is presented the Training Virtual Environment and the Control Algorithm development. Figure 2 presents the proposed scheme for the Training Virtual Environment, which is developed in the graphic engine Unity3D. The training system contains: *i) External Resources*, that considers the CAD models of the environment that are developed in the Inventor software, post processed in 3DS-Max, and exported in (.fbx) format to be compatible with the graphic engine; *ii) Graphics Engine*, contains all of the graphic resources used in the environment, and the scripts that are the code lines that configures all of the system elements; *iii) Controller*, contains the control algorithm

developed in the mathematical software MATLAB and the communication between the mathematical software and the graphics engine. The Unity3D-MATLAB communication is performed by using the dynamic link library (DLL) to assign a shared memory between both software's, in order to exchange data to implement the control algorithm developed in MATLAB [18, 19]; iv) *Virtual Devices*, that considers the input and output devices to allow the immersive experience to the user.

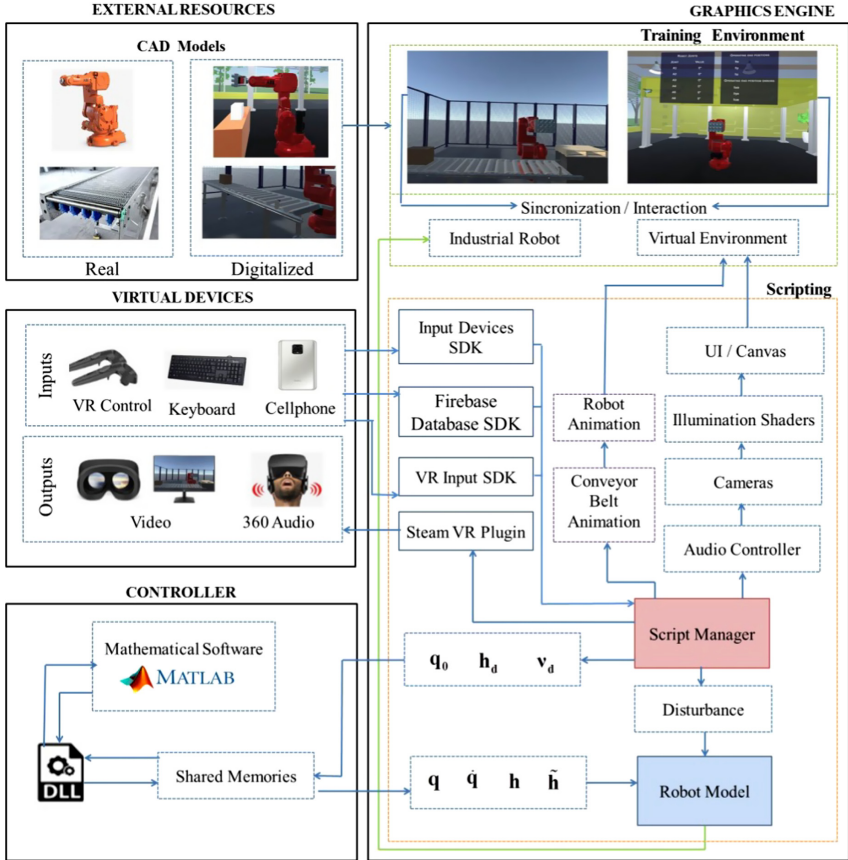


Fig. 2. Proposed virtual environment scheme.

Furthermore, it is important to emphasize the functionality of the Cellphone input device, because it contains the Robot Control Panel application. The Robot Control Panel is developed in Android Studio. In the application the user can define the movement of each joint of the robot and its velocity. Figure 3 shows the Robot Control Panel user interface.

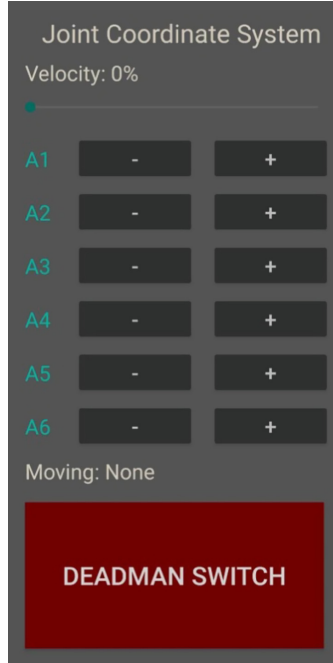


Fig. 3. Robot control panel user interface.

The Robot Control Panel uses Firebase Real-time Database to allow wireless connection with the Industrial Robotics Virtual Laboratory environment, where the information sent by the user with the Robot Control Panel application to the database is received by the virtual environment in real time to perform the desired actions with the robot.

3.1 Control Algorithm

3.1.1 Movement Problems

In order to perform object manipulation tasks with the robot, there are two fundamental problems to solve with the control algorithm: point stabilization and trajectory tracking [15]. Figure 4 represents the point stabilization problem of a manipulator robot, which consists in moving the end effector of the robot to a fixed target location that is defined by the desired position and orientation of the end effector of the robot

$$\mathbf{h}_d = [h_{dx}, h_{dy}, h_{dz}, h_{d\alpha}, h_{d\beta}, h_{d\gamma}]^T. \tag{4}$$

The desired position and orientation are constant values, so the desired velocity for point stabilization is

$$\mathbf{v}_d(t) = \dot{\mathbf{h}}_d(t) = 0. \tag{5}$$

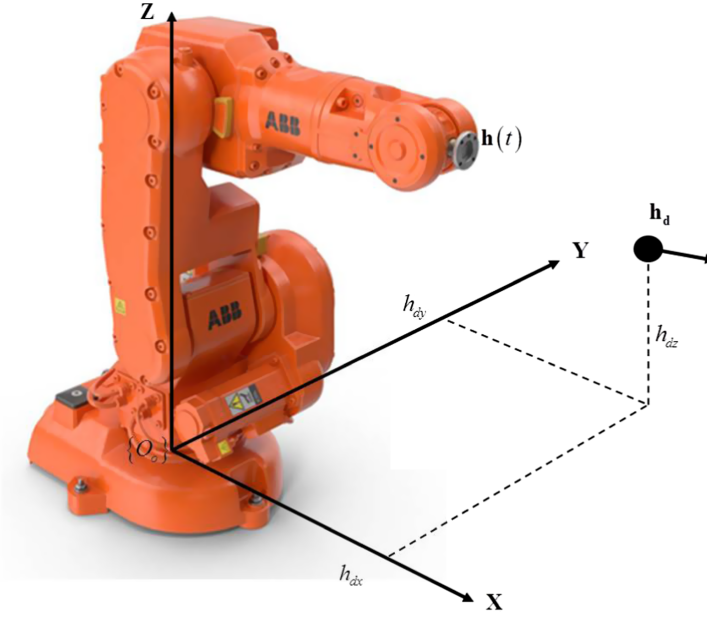


Fig. 4. Point stabilization problem for the manipulator robot represented in the world framework $\{O_0, X, Y, Z\}$.

The point stabilization problem consists in finding a closed loop control law, where the position and orientation errors are [17]

$$\tilde{\mathbf{h}}(t) = \mathbf{h}_d - \mathbf{h}(t), \quad (6)$$

hence

$$\lim_{t \rightarrow \infty} (\mathbf{h}_d(t) - \mathbf{h}(t)) = 0. \quad (7)$$

The second problem for the object manipulation with the robot is the trajectory tracking, which consists in following a desired time-varying trajectory and its derivative that describes the desired velocity [15]. This desired trajectory is independent of the instantaneous position of the robot. The desired trajectory for the manipulator robot is defined by a vector

$$\mathbf{h}_d(t) = [h_{dx}(t), h_{dy}(t), h_{dz}(t), h_{d\alpha}(t), h_{d\beta}(t), h_{d\gamma}(t)]^T. \quad (8)$$

The desired velocity $\mathbf{v}_d(t)$ is the time derivative of $\mathbf{h}_d(t)$, *i.e.*, $\mathbf{v}_d(t) = \frac{d}{dt} \mathbf{h}_d(t)$.

Figure 5 shows the trajectory tracking problem of a manipulator robot, which should consider the position and orientation of the operating end of the manipulator robot $\mathbf{h}(t)$ and the desired positions $\mathbf{h}_d(t)$ and velocities $\mathbf{v}_d(t)$ for the operating end, in order to achieve

$$\lim_{t \rightarrow \infty} (\mathbf{h}_d(t) - \mathbf{h}(t)) = 0. \quad (9)$$

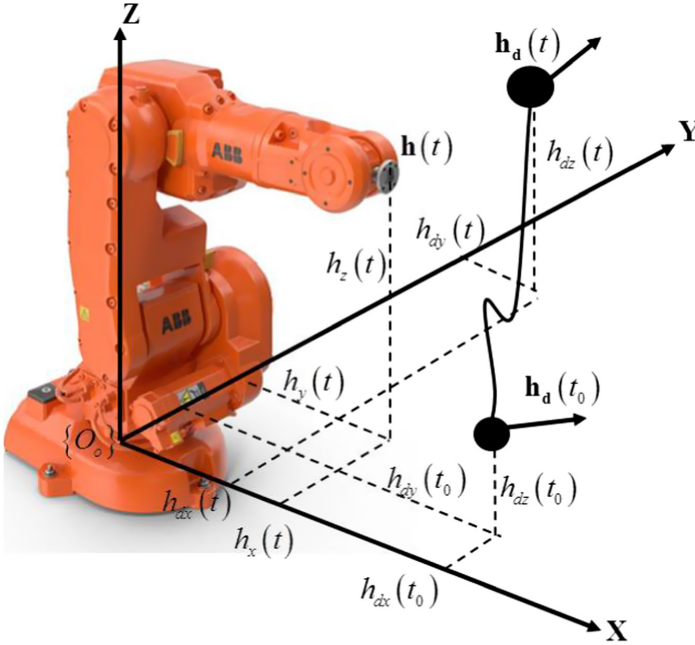


Fig. 5. Trajectory tracking problem for the manipulator robot represented in the world framework $\{O_o, X, Y, Z\}$.

3.1.2 Proposed Control Law

To solve the point stabilization and trajectory tracking problems with the manipulator robot, it is proposed a feedback control law that considers the requirements of both problems to perform manipulation tasks with the robot. The controller considers the redundancy characteristics of the manipulator robot to perform the control actions. The control algorithm uses the minimal norm solution, in order to reach the target with a small number of movements [17]. The controller is developed in MATLAB, and uses the inverse differential kinematics method in order to achieve the desired tasks by saturating the position and orientation errors $\tilde{\mathbf{h}}(t)$ to calculate the articular velocities with the following control law:

$$\dot{\mathbf{q}}_{\text{ref}} = \mathbf{J}^{-1}(\mathbf{q}) \left(\mathbf{v}_d + \mathbf{K} \tanh(\tilde{\mathbf{h}}) \right) \quad (10)$$

where $\dot{\mathbf{q}}_{\text{ref}}(t)$ is the robot joint angular velocity vector, $\mathbf{J}^{-1}(\mathbf{q})$ is the inverse Jacobian matrix of the manipulator robot and \mathbf{K} is the definite positive diagonal gain matrix. The proposed control law considers the redundancy characteristics of the manipulator robot. The minimal norm solution is used for the design of the controller that means that the manipulator robot will follow the desired task at any time with the smallest number of possible movements. The $\tanh(\cdot)$ function is proposed to perform analytical saturations for the velocities of the manipulator robot [20].

3.1.3 Stability and Robustness

Using the kinematic model of the manipulator robot (3), the proposed control law (10) and considering the errors caused by the disturbance applied to the system

$$\dot{\tilde{\mathbf{q}}} = \dot{\mathbf{q}}_{\text{ref}} - \dot{\mathbf{q}} \quad (11)$$

and also considering that the desired velocity is $\mathbf{v}_d = \dot{\mathbf{h}}_d$ and the end effector velocity errors are $\dot{\tilde{\mathbf{h}}} = \dot{\mathbf{h}}_d - \dot{\mathbf{h}}$ the resultant closed loop equation is

$$\dot{\tilde{\mathbf{h}}} = -\mathbf{K} \tanh(\tilde{\mathbf{h}}) + \mathbf{J}\dot{\tilde{\mathbf{q}}}. \quad (12)$$

Using the candidate Lyapunov function $\mathbf{V}(\tilde{\mathbf{h}}) = \frac{1}{2}\tilde{\mathbf{h}}^T\tilde{\mathbf{h}}$ for the stability analysis, and replacing the closed loop equation in its time derivate $\dot{\mathbf{V}}(\tilde{\mathbf{h}}) = \tilde{\mathbf{h}}^T\dot{\tilde{\mathbf{h}}}$ the resultant expression is [15]

$$\dot{\mathbf{V}}(\tilde{\mathbf{h}}) = -\tilde{\mathbf{h}}^T\mathbf{K} \tanh(\tilde{\mathbf{h}}) + \tilde{\mathbf{h}}^T\mathbf{J}\dot{\tilde{\mathbf{q}}}. \quad (13)$$

To be negative definite the Eq. (13) should accomplish $\dot{\mathbf{V}}(\tilde{\mathbf{h}}) < 0$, then.

$$\|\mathbf{K} \tanh(\tilde{\mathbf{h}})\| > \|\mathbf{J}\dot{\tilde{\mathbf{q}}}\|, \quad \text{so,} \quad (14)$$

$$\|\tilde{\mathbf{h}}\| < \frac{\|\mathbf{J}\dot{\tilde{\mathbf{q}}}\|}{\lambda_{\min}(\mathbf{K}) \tanh(k_{aux})} \quad \text{with } 0 < k_{aux} < 1 \quad (15)$$

where $\lambda_{\min}(\mathbf{K})$ are the minimum auto-values and considering that if $\dot{\tilde{\mathbf{q}}} = 0$ then $\tilde{\mathbf{h}} \rightarrow 0$ when $t \rightarrow \infty$ for which it can be concluded that the system is asymptotically stable for the point stabilization and trajectory tracking problems.

4 Experimental Results

This section presents the experimental results of both of the virtual training environments, the control algorithm and the Robot Control Panel application. The hardware used for the tests of the virtual environment and the control algorithm is a mid-range computer that contains (4th generation Intel Core I7 processor, 8 GB DDR3 RAM and 64-bit Windows 8.1 operating system); and for the Robot Control Panel it is used an Android cellphone with Snapdragon 720G processor and 6 GB RAM. The experimental results are divided in two parts. The first part considers the Industrial Virtual Environment tests, where the control algorithm is applied to perform manipulation tasks using different point stabilization locations. The second part contains the Industrial Robotics Virtual Laboratory and the Robot Control Panel performance tests, where the user can define the desired joint values for the manipulator robot in the environment. Lately the usability of the virtual environment will be presented.

4.1 Industrial Virtual Environment Tests

The Industrial Virtual Environment tests, considers a palletizing robotic cell, where the robot performs box manipulation tasks. Considering the point stabilization problem for the manipulation tasks are defined five desired positions and fixed orientations that will be reached by the end-effector of the robot, in order to perform the desired manipulation task avoiding crashes with the other objects in the virtual environment. The initial conditions for the robot joints are $\mathbf{q}_0 = [0 \frac{\pi}{9} -\frac{\pi}{6} \frac{\pi}{18} \frac{\pi}{12} -\frac{\pi}{18}]$ [rad]. Figure 6 presents a box palletizing sequence performed by the manipulator robot in the virtual environment.

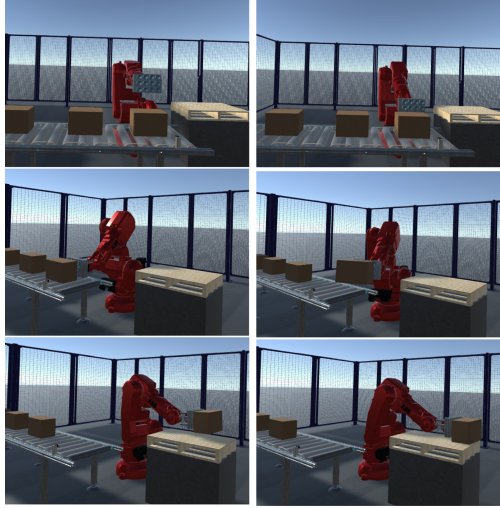


Fig. 6. Manipulator robot performing palletizing tasks.

To confirm the proper functionality of the robot in the palletizing application in Fig. 7 will be shown the position and orientation errors of the end-effector of the robot.

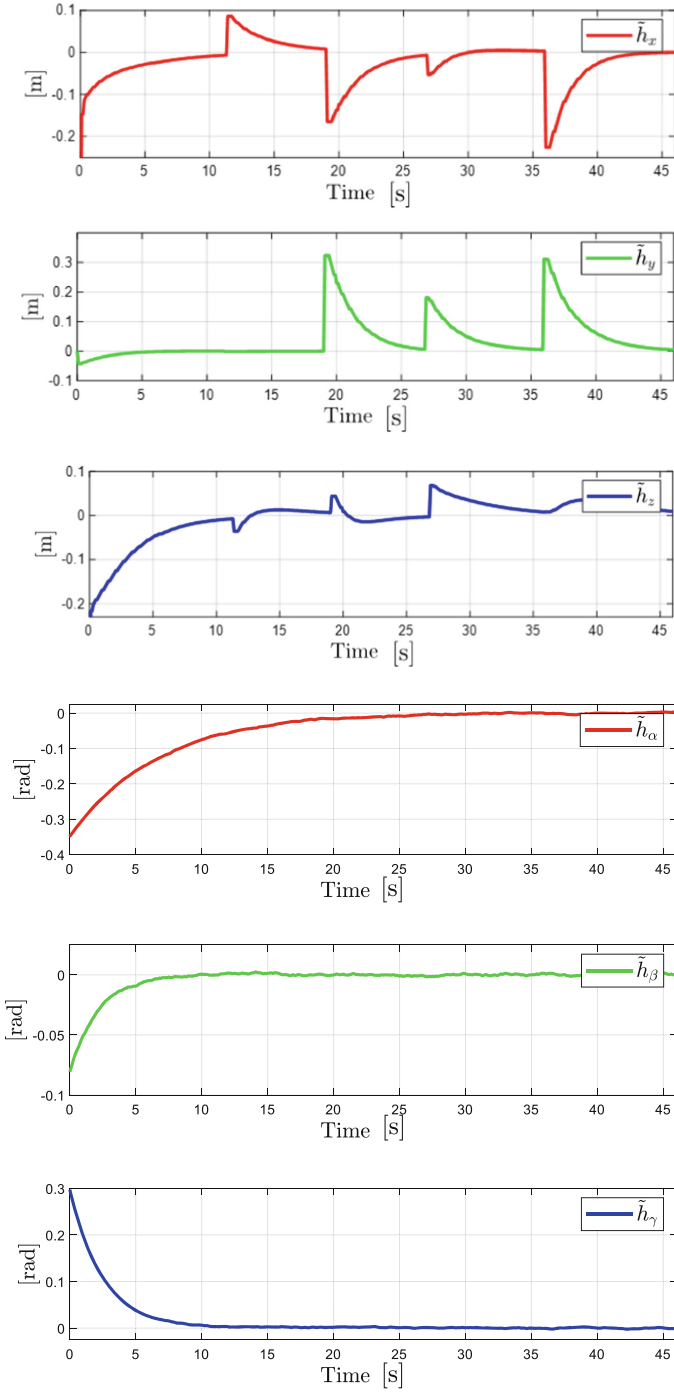


Fig. 7. Time graphics of the robot end-effector position and orientation errors.

The position and orientation error graphics presented shows that the desired task is accomplished with the smallest amount of movement possible and the errors converge to zero when $t \rightarrow \infty$.

Furthermore, the desired values for the trajectory tracking test are.

$\mathbf{h}_d = \left[0.4 + 0.05 \sin(0.2t) \quad 0.4 + 0.2 \sin(0.2t) \quad 0.4 + 0.05 \sin(0.1t) \quad 0 \quad \frac{\pi}{4} \quad 0 \right]$ and the initial conditions for the robot joints are $\mathbf{q}_0 = \left[0 \quad \frac{\pi}{9} \quad -\frac{\pi}{6} \quad \frac{\pi}{18} \quad \frac{\pi}{12} \quad -\frac{\pi}{18} \right]$. Figure 8 shows the trajectory followed by the manipulator robot.



Fig. 8. Trajectory tracking task performed by the manipulator robot.

Figure 9 shows the obtained position and orientation errors for the trajectory tracking test.

As seen in the previous images it's accomplished the propose of the controller because the robot follows the desired trajectory and the position and orientation errors converge to zero.

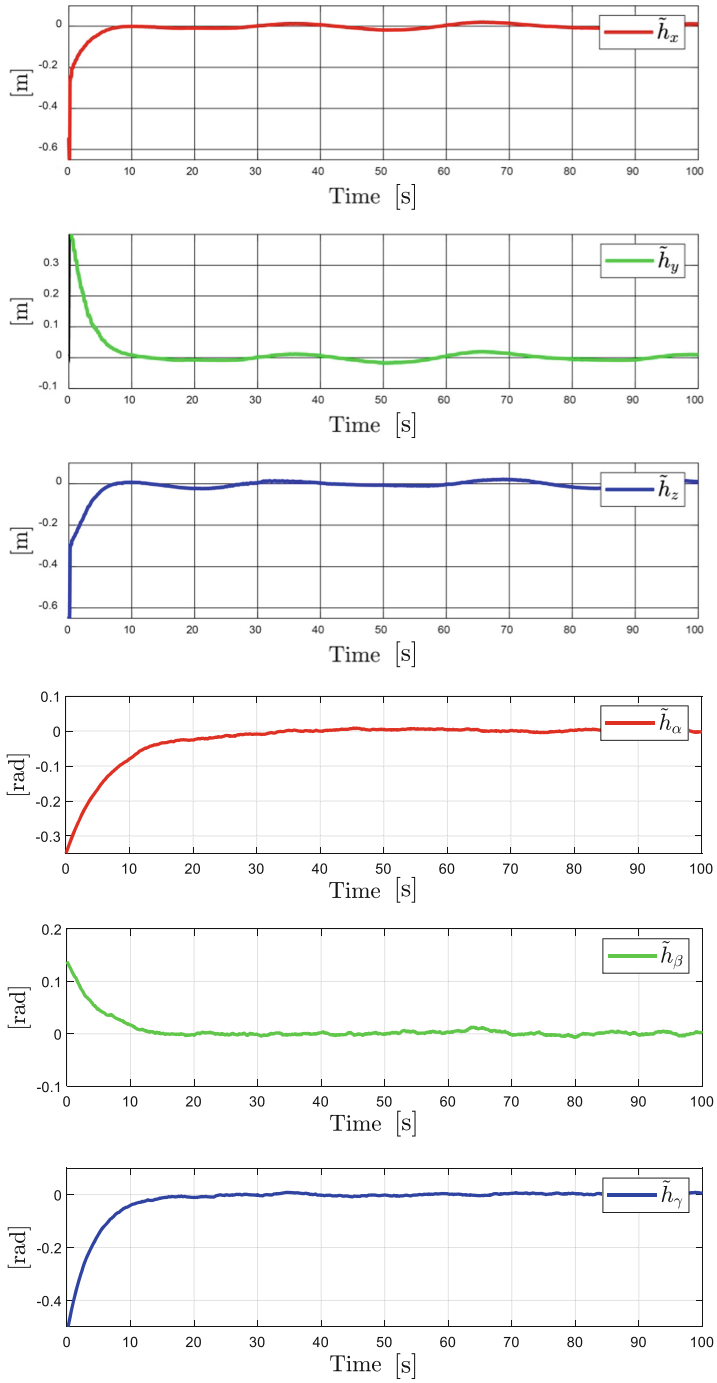


Fig. 9. Position and orientation errors of the trajectory task performed by the robot.

4.2 Virtual Laboratory Tests

This section contains the connection and functionality tests between the Virtual Laboratory Environment and the Robot Control Panel application. This environment is oriented to the teaching-learning processes, so, the user can observe the most important variables of the robot performance in a canvas in real time. Figure 10 shows the performance test of the Laboratory Environment connected with the Robot Control Panel Application.



Fig. 10. Virtual laboratory environment and robot control panel tests.

Figure 11 shows the performance of the controller when the user defines desired angles \mathbf{q}_d for the joint configuration of the manipulator robot by using the Robot Control Panel application.

The Virtual Laboratory Environment allows the user to understand the physical characteristics of the manipulator robot without buying the real one. With the Robot Control Panel application the user can define the desired values for the robot joints and observe the robot performance in real time. In order to send information from the Robot Control Panel to the Virtual Laboratory Environment the user needs to define the desired velocity and click simultaneously the dead man switch and the plus or minus button of the joint to move, so the robot performs the action.

4.3 System Usability

There was considered a group of 20 people for the usability tests in order to measure the experience in the use of the virtual environment. All of the participants installed the Virtual Training System before the tests and were capacitated in the use of the system. For the experiments were considered joint movement tasks defined by the users with the Robot Control Panel application and performed by the robot in the Virtual Laboratory environment. After finishing the tests, the experimental group carried out a usability test to measure the acceptance of the virtual system.

The degree of usability of the system is calculated by using one of the most popular questionnaires for usability tests in the scientific community, this is the System Usability Scale (SUS) [21, 22]. The questionnaire consists in a quiz that has positive and negative statements. The questions were adapted for the proposed Virtual Training System and

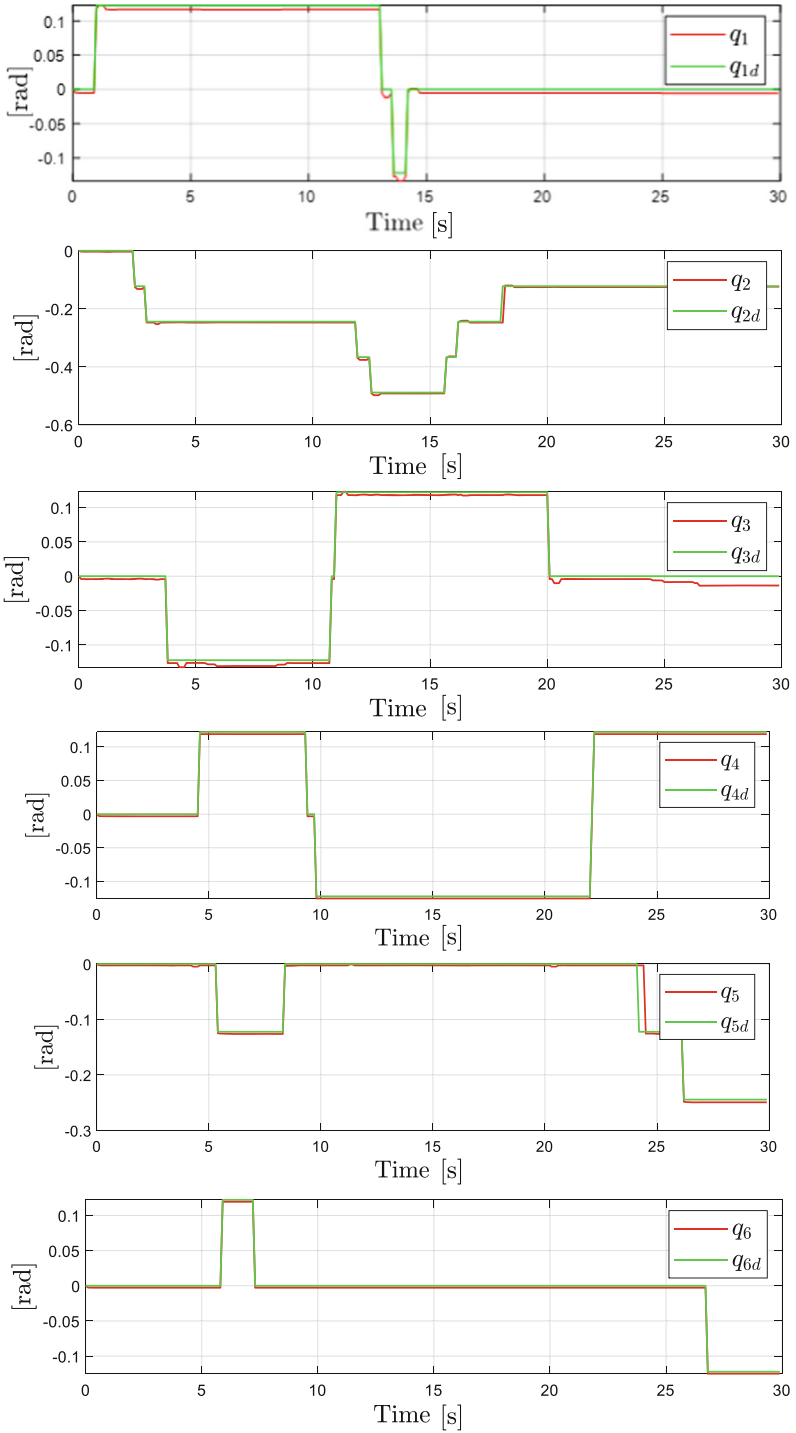


Fig. 11. Performance time graphics of the robot joint movements.

it's based on Likert scales, on which the response options varies from strongly agree (5) to strongly disagree (1). The questionnaire applied to the participants for the usability tests and the results can be observed in the Table 1, where the pair questions consider negative attitudes and odd questions express positive attitudes. To obtain the SUS score the odd question results should be subtracted by 1 and the pair questions should be subtracted from 5 by the obtained value. The resultant scores of each question should be added and that value should be multiplied by 2.5 in order to convert the value to 0–100 range [23].

Table 1. SUS survey results.

N°	Question	Score	Operation
Q1	I think that I would like to use this virtual training system frequently	5	$5 - 1 = 4$
Q2	I found the virtual training system unnecessarily complex	1	$5 - 1 = 4$
Q3	I thought the virtual training system was easy to use	4	$4 - 1 = 3$
Q4	I think that I would need the support of a technical person to be able to use this virtual training system	2	$5 - 2 = 3$
Q5	I found the various functions in this virtual training system were well integrated	4	$4 - 1 = 3$
Q6	I thought there was too much inconsistency in this virtual training system	1	$5 - 1 = 4$
Q7	I would imagine that most people would learn to use this virtual training system very quickly	4	$5 - 1 = 4$
Q8	I found the virtual training system very uncomfortable to use	1	$5 - 1 = 4$
Q9	I felt very confident using the virtual training system	4	$4 - 1 = 3$
Q10	I needed to learn a lot of things before I could get going with this virtual training system	3	$5 - 3 = 2$
		TOTAL	$34 \times 2.5 = 85$

The resultant value of the SUS test is 85%, this value shows an excellent degree of usability for the Virtual Training System; so, it allows that the virtual environment can be used in the teaching-learning processes.

5 Conclusion

VR environments serves as an alternative and effective method for the implementation of Industrial Robotics, using industrial robots in virtual environments to perform manipulation tasks is a viable way to allow users to understand the characteristics and restrictions of industrial robots without acquiring real robots. The interconnection of VR environments with mathematical software's using shared memory, allows implementing

and testing control algorithms for manipulator robots in real time. The implementation of numerical methods in closed loop controllers allows performing manipulation tasks with industrial robots in virtual environment with the possible less movements to perform the action. However, the synergy between VR environments and control algorithms in mathematical software's are a viable way to boost teaching-learning processes in the scientific community.

Acknowledgements. The authors would like to thank the Universidad de las Fuerzas Armadas ESPE and the ARSI research group for the support in the development of this work.

References

1. Stankovski, S., Ostojić, G., Baranovski, I., Babić, M., Stanojević, M.: The impact of edge computing on industrial automation. In: Proceedings of the 2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH), March 2020, pp. 1–4 (2020)
2. Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., Hoffmann, M.: Industry 4.0. *Bus. Inf. Syst. Eng.* **6**(4), 239–242 (2014). <https://doi.org/10.1007/s12599-014-0334-4>
3. Mirashe, S.P., Kalyankar, N.V.: Cloud Computing (2010). [arXiv:1003.4074](https://arxiv.org/abs/1003.4074) [cs]
4. Ranjan, J., Foropon, C.: Big data analytics in building the competitive intelligence of organizations. *Int. J. Inf. Manag.* **56**, 102231 (2021). <https://doi.org/10.1016/j.ijinfomgt.2020.102231>
5. Tresa, M., Francina, S., Jerlin Oviya, V., Lavanya, K.: A study on internet of things: overview, automation, wireless technology, robotics. *Ann. Roman. Soc. Cell Biol.* **2021**, 6546–6555 (2021)
6. Atici-Ulusu, H., Ikiz, Y.D., Taskapilioglu, O., Gunduz, T.: Effects of augmented reality glasses on the cognitive load of assembly operators in the automotive industry. *Int. J. Comput. Integr. Manuf.* **34**, 487–499 (2021). <https://doi.org/10.1080/0951192X.2021.1901314>
7. Araiza-Alba, P., et al.: The potential of 360-degree virtual reality videos to teach water-safety skills to children. *Comput. Educ.* **163**, 104096 (2021). <https://doi.org/10.1016/j.compedu.2020.104096>
8. Newbury, R., Cosgun, A., Crowley-Davis, T., Chan, W.P., Drummond, T., Croft, E.: Visualizing Robot Intent for Object Handovers with Augmented Reality (2021). [arXiv:2103.04055](https://arxiv.org/abs/2103.04055) [cs]
9. Radu, I., Hv, V., Schneider, B.: Unequal impacts of augmented reality on learning and collaboration during robot programming with peers. In: Proceedings of the ACM on Human-Computer Interaction, vol. 4, p. 1822. ACM Digital Library (2021). <https://doi.org/10.1145/3446568>
10. Gausemeier, J., Berssenbrügge, J., Grafe, M., Kahl, S., Wassmann, H.: Design and VR/AR-based testing of advanced mechatronic systems. In: Ma, D., Fan, X., Gausemeier, J., Grafe, M. (eds.) Proceedings of the Virtual Reality & Augmented Reality in Industry, pp. 1–37. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-17376-9_1
11. Arnarson, H., Solvang, B., Shu, B.: The application of virtual reality in programming of a manufacturing cell. In: Proceedings of the 2021 IEEE/SICE International Symposium on System Integration (SII), January 2021, pp. 213–218 (2021)
12. Ma, D., Zhen, X., Hu, Y., Wu, D., Fan, X., Zhu, H.: Collaborative virtual assembly operation simulation and its application. In: Ma, D., Fan, X., Gausemeier, J., Grafe, M. (eds.) Proceedings of the Virtual Reality & Augmented Reality in Industry, pp. 55–82. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-17376-9_3

13. Fan, X., Yang, R., Wu, D., Ma, D.: Virtual assembly environment for product design evaluation and workplace planning. In: Ma, D., Fan, X., Gausemeier, J., Grafe, M. (eds.) *Proceedings of the Virtual Reality & Augmented Reality in Industry*, pp. 147–161. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-17376-9_9
14. Almaged, M.: Forward and inverse kinematic analysis and validation of the ABB IRB 140 industrial robot. *J. Mech. Eng. Technol. (JMET)* **9**, 1–20 (2017)
15. Andaluz, V., Roberti, F., Toibero, J.M., Carelli, R.: Adaptive unified motion control of mobile manipulators. *Control. Eng. Pract.* **20**, 1337–1352 (2012). <https://doi.org/10.1016/j.conengprac.2012.07.008>
16. Corke, P.I.: A simple and systematic approach to assigning denavit-hartenberg parameters. *IEEE Trans. Rob.* **23**, 590–594 (2007). <https://doi.org/10.1109/TRO.2007.896765>
17. Siciliano, B., Sciacivco, L., Villani, L., Oriolo, G.: *Robotics: Modelling, Planning and Control* (2009). <https://doi.org/10.1007/978-1-84628-642-1>
18. Andaluz, V.H., et al.: Unity3D-MatLab simulator in real time for robotics applications. In: De Paolis, L.T., Mongelli, A. (eds.) *AVR 2016. LNCS*, vol. 9768, pp. 246–263. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40621-3_19
19. Carvajal, C., Méndez, M., Torres, D., Terán, C., Arteaga, O., Andaluz, Víctor. H.: Autonomous and tele-operated navigation of aerial manipulator robots in digitalized virtual environments. In: Paolis, L.T.D., Bourdot, P. (eds.) *AVR 2018. LNCS*, vol. 10851, pp. 496–515. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95282-6_36
20. Ortiz, J.S., Palacios-Navarro, G., Andaluz, V.H., Guevara, B.S.: Virtual reality-based framework to simulate control algorithms for robotic assistance and rehabilitation tasks through a standing wheelchair. *Sensors (Basel)* **21**, 5083 (2021). <https://doi.org/10.3390/s21155083>
21. Sauro, J., Lewis, J.: When designing usability questionnaires, does it hurt to be positive?. pp. 2215–2224 (2011)
22. Salvendy, G.: *Handbook of Human Factors and Ergonomics*. John Wiley & Sons, Hoboken (2012). ISBN 978–1–118–12908–1
23. Borsci, S., Federici, S., Lauriola, M.: On the dimensionality of the system usability scale: a test of alternative measurement models. *Cogn. Process.* **10**, 193–197 (2009). <https://doi.org/10.1007/s10339-009-0268-9>