A Teach Pendant to Control Virtual Robots in RoboAnalyzer

Ishaan Mehta, Keshav Bimbraw, Rajeevlochana G. Chittawadigi and Subir K. Saha

Abstract—**Teach programming is an interactive way to program industrial robots. It involves usage of a handheld control unit called teach pendant that can be used to control the motion of a robot. It provides a very convenient method to teach trajectories to the robot. One of the advantages of teach pendant programming is that an operator is not required to learn any special programming language. Hence, teach pendants can be used as an effective tool in educational robotics as well. These devices are generally proprietary and work for a specific robot. As few institutes have robots, the teach pendants are not available to a lot of students to use and control robots. In this paper, a generic 'teach pendant' has been reported that can be used to control the motion of virtual robots available in RoboAnalyzer, a 3D model based robotics learning software. The teach pendant can be used by students to program virtual robots.**

I. INTRODUCTION

Rapidly increasing demand for automation in various sectors has led to a surge of using robots. They are being used in various industries like automotive, healthcare, electronics, etc. As a result, a lot of research activities are going on in this field. Because of its relevance, various technical institutes around the world have introduced robotics to their curriculum at undergraduate and post-graduate levels. Robotics is a multi-disciplinary field and involves various aspects like mechanics, programming, control, electronics, motion planning, etc. One of the very important aspects of any robot operation is its programming. It is a method to give instructions to a robot regarding tasks to be performed. There are two basic modes of programming robots as given in [1], i.e., offline programming and online programming.

Offline programming is a technique in which a program for robot operation is developed on a computer with the models of robots, work cell, surrounding environment, etc. while the controller is not connected. The program can be simulated in virtual environment of the software using the CAD models of robots and the workspace. After simulation, if found appropriate, the program can be transferred to the robot's controller to run an actual robot. Various offline robot programming software are available such as RoboDK[2], RobCAD[3] and V-REP[4]. These software can also be used for effective teaching of robot programming to students.

Online programming is a technique in which a robot is operational while a program is developed and run on its controllers. Programming is done based on teaching the robot, points or trajectories it has to follow. Necessary data is collected by robot controllers during the teach process. Teaching is usually carried out using a handheld robot terminal called 'teach pendant'. Teach pendant provides convenient means to move the robot, teach locations and run robot programs, by making use of jogging of the robot in joint and Cartesian spaces. Most of the robots manufacturers provide their own teach pendants for their industrial robots.

One of the main advantage of programming using teach pendant is simplicity in programming, i.e. operator does not require any special programming skills. Hence teach pendants can be effectively used for educational purposes as well. The major setback is however that such setups are really expensive and are not accessible to many students. One of the proposed solutions to this problem is the use of a remote laboratory using the concepts of tele-robotics [5, 6]. These platforms provide an emulation of teach pendant in the form of a GUI that is used to remotely program real robots and the motion of the manipulator can be viewed using video streaming. RoboLab, a virtual and remote laboratory application for teaching robotics, has used joystick to define movement of its virtual robot [7, 8]. Besides this, a smartphone based teach pendant to control a virtual robot has been reported in [9]. Hammer, a tablet based interface for industrial robot programming reported in [10], comes with features like virtual teach pendant that can be used to execute motion of robot in simulated environment provided in the application.

The authors in the best of their knowledge, could not find in the literature, work related to hardware that control virtual robots. This combination lets students get to learn how teach pendants work. The other component, i.e., virtual robots in a software can be made available so that they can be integrated and controlled by the physical device.

In this paper, a generic teach pendant developed by the authors is reported. Commercially-of-the-shelves (COTS) available Arduino board and other components have been used for its development. It can be used to control serial robots in Virtual Robot Module of RoboAnalyzer software, also developed by the team of last two authors. The teach pendant can perform joint jogging and Cartesian level jogging. It can also be used to teach profiles to Virtual Serial robots. Besides this, it can also be used to control some actual robots in real time. Section II of this paper gives an overview of some of the basic robotics concepts and RoboAnalyzer software. Sections III and IV explain the working and applications of the developed teach pendant, respectively. The conclusions are given in Section V.

Ishaan Mehta and Keshav Bimbraw are pursuing B.E in Mechatronics Engineering in Thapar University, Patiala, India. The work reported is an outcome of their semester internship at IIT Delhi (mehtaishaan4@gmail.com, keshavbimbraw@yahoo.co.in).

Rajeevlochana G. Chittawadigi is an Asst Professor in Mechanical Engineering Department, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Amrita University, Bengaluru, India (rg_chittawadigi@blr.amrita.edu).

Subir K. Saha is a Professor and Head in Mechanical Engineering Department, Indian Institute of Technology Delhi, New Delhi, India (saha@mech.iitd.ac.in).

II. Overview of Robot Kinematics

Robot kinematics relates the joint motion and the end-effector (EE) motion in the Cartesian frame. A brief explanation of robot kinematics and RoboAnalyzer software is given below.

A. *Robot kinematics concepts*

- Serial robots are made of several links connected using single degree-of-freedom (DOF) joints. It is necessary to find the relation between coordinate frames of EE and base, in order to control EE with respect to the base. Hence, each robot link is attached with a coordinate frame as shown in Fig. 1. The relationship between the consecutive frames is given using Denavit-Hartenberg (DH) [11] parameters listed in Table 1 and are illustrated in Fig. 2.

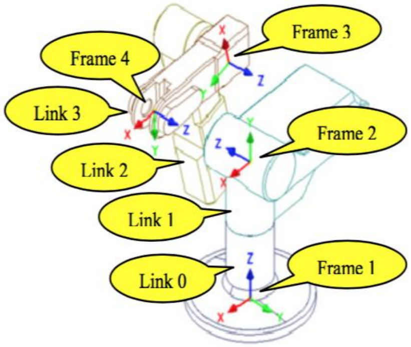


Figure 1. Coordinate frames attached to robot links

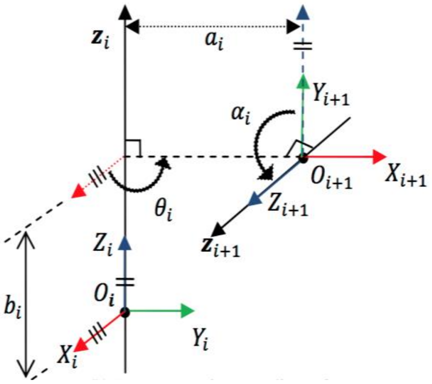


Figure 2. Denavit-Hartenberg (DH) parameters

TABLE I. DESCRIPITION OF DH PARAMETERS

Parameters	Description
Joint offset (b_i)	Distance between X_i and X_{i+1} along Z_i
Joint angle (θ_i)	Angle between X_i and X_{i+1} about Z_i
Link length (a_i)	Distance between Z_i and Z_{i+1} along X_{i+1}
Twist angle (α_i)	Angle between Z_i and Z_{i+1} about X_{i+1}

- Forward Kinematics: In this method, the position and orientation of EE is determined on the basis of given joint angles or values. For an n-DOF serial manipulator, each link is assigned a frame, Frame 1 being attached to the fixed link and Frame n+1 to the end of n^{th} link. Then, the DH parameters are defined as outlined in [1]. Accordingly, the homogenous transformation matrices (HTM), $\mathbf{T}_1$, $\mathbf{T}_2$, $\mathbf{T}_3$...$\mathbf{T}_n$, are defined. The transformation of Frame i+1 with respect to Frame i, i.e., $\mathbf{T}_i$, is given by

$$\mathbf{T}_i \equiv \begin{bmatrix} C\,\theta_i & -S\,\theta_i\,C\,\alpha_i & S\,\theta_i\,S\,\alpha_i & a_i\,C\,\theta_i \\ S\,\theta_i & C\,\theta_i\,C\,\alpha_i & -C\,\theta_i\,S\,\alpha_i & a_i\,S\,\theta_i \\ 0 & S\,\alpha_i & C\,\alpha_i & b_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

where $C\,\theta_i \equiv cos\theta_i$, $S\,\theta_i \equiv sin\theta_i$, and similarly for the terms containing α_i. On this basis, one can calculate the HTM of the EE with respect to the fixed base, i.e., $\mathbf{T}$, using (1) as

$$\mathbf{T} = \mathbf{T}_1\mathbf{T}_2\mathbf{T}_3 \ldots \mathbf{T}_n \qquad (2)$$

On substitution of (1) in (2), $\mathbf{T}$ can be expressed as

$$\mathbf{T} = \begin{bmatrix} \mathbf{Q} & \mathbf{p} \\ 0 & 1 \end{bmatrix} \qquad (3)$$

where $\mathbf{Q}$ represents 3×3 orientation matrix of the EE and $\mathbf{p}$ is the 3D position vector of the EE, both with respect to the base frame.

- Roll-Pitch-Yaw (RPY) is the set of Euler angles used for the description of orientation. These angles define the rotation of a frame about an axis, i.e., about X-axis (roll), about Y-axis (pitch), and about Z-axis (yaw).
- The manipulator Jacobian matrix relates the linear and angular velocities of the EE to joint velocities [1]. It is expressed as:

$$\mathbf{t}_e = \mathbf{J}\dot{\boldsymbol{\theta}} \qquad (4)$$

where $\mathbf{t}_e \equiv [\boldsymbol{\omega}_e^T\ \mathbf{v}_e^T]^\mathbf{T}$ represents the twist vector, $\mathbf{J}$ represents $6\times n$ Jacobian matrix and $\dot{\boldsymbol{\theta}}$ represents joint rate. For a serial manipulator with all revolute joints, $\mathbf{J}$ can be calculated using the expression

$$\mathbf{J} \equiv \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_n \\ \mathbf{e}_1 \times \boldsymbol{a}_{1e} & \mathbf{e}_2 \times \boldsymbol{a}_{2e} & \ldots & \mathbf{e}_1 \times \mathbf{a}_{ne} \end{bmatrix} \qquad (5)$$

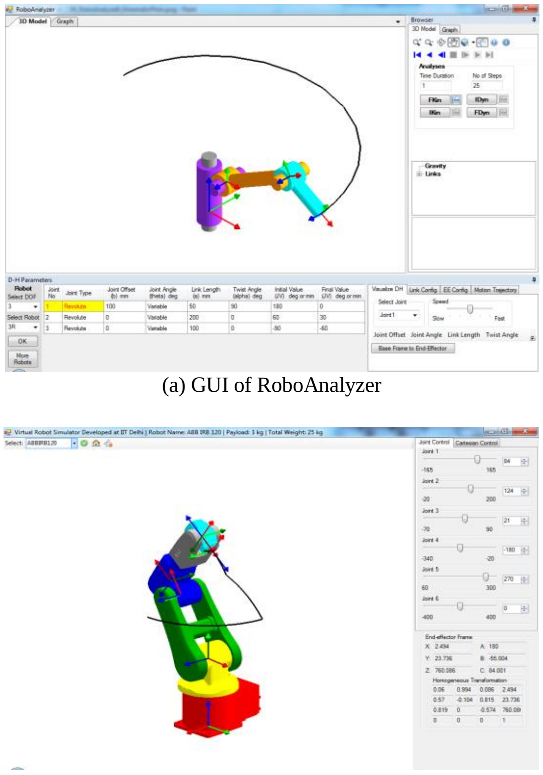
where $\mathbf{e}_i$ is the unit vector parallel to the axis of the i^{th} joint and $\boldsymbol{a}_{ie}$ is the vector drawn from the origin of Frame i to the origin of Frame n+1 attached to the EE. Vector $\boldsymbol{a}_{ie}$, for i = 1, 2...n, can be computed as

$$\boldsymbol{a}_{ie} = \sum_{j=1}^{n} \boldsymbol{a}_{je}. \qquad (6)$$

B. *RoboAnalyzer*

RoboAnalyzer is a 3D model based software having several modules that can be used effectively for visualization of various concepts of robotics. A screenshot is shown in Fig. 3(a). It can show the animation of DH parameters/frames and performs forward and inverse kinematics, and dynamic analyses of serial robots. Besides this, CAD models of different robots can be imported and viewed in 3D virtual environment [12].

Virtual Robot Module (VRM) [13], shown in Fig.3 (b), also a part of RoboAnalyzer, allows users to visualize 17 commercially available industrial robots using their CAD models. The VRM displays the selected robot model, its specification, DH frames attached to the various links, homogeneous transformation matrix of the EE, and the trace of the EE. The pose or the configuration of the EE is displayed in terms of the Cartesian coordinates and RPY angles with respect to the frame attached to the base link. VRM was primarily developed as robot motion planning module of RoboAnalyzer.



(a) GUI of RoboAnalyzer

(b) Virtual Robot Module

Figure 3. RoboAnalyzer software

VRM allows three types of robot motion planning:

- *Joint Jogging:* It allows joint-level motion study of the robots by the movement of one joint at a time between the specified joint limits. It can be accessed from Joint Control pane. Equations (1-3) are used for the simulation and animation.
- *Cartesian Jogging:* VRM allows changing the position and orientation of the EE in terms of Cartesian coordinates and RPY angles. Continuous motion of robot joints are achieved as the EE is jogged along the required direction or rotated by required angles. It can be accessed from Cartesian control pane. Equations (4,5) are used for determination and animation of desired robot motion.
- *Cartesian Motion:* In this mode EE can be made to move along a straight line by specifying initial and final positions in terms of Cartesian coordinates and RPY angles. It is an extension of Cartesian jogging.

III. Teach Pendant

The authors wanted to develop a device that can be integrated with the Virtual Robot Module of the RoboAnalyzer. It had to be a handheld control device that can be used to teach motion to various robots in VRM. It would primarily work as a teach programming unit for RoboAnalyzer, a software developed using Visual C# (.NET framework). The intent was also to build a device that can be used by students to teach motion to virtual robots and robots in real time as well. Hence, the experience of the use of an industrial teach pendant can be imparted.

With the given situation, there was a choice to use commercially available joysticks/gamepads as teach pendant. However, using such joysticks would diminish the scope of extending the scope of teach pendant for various applications. Alternatively, using commercially available microcontroller board would enable easy development of the pendant and also allow further modifications. A comparison between joystick and microcontroller board is given in Table 2.

TABLE 2. COMPARISON OF MICROCONTROLLER BOARD AND JOYSTICK

Microcontroller board based teach pendant	Joystick/Gamepads as teach pendants
Can be programmed according to requirements	No such provision is available.
A variety of peripheral devices like LCD can be interfaced similar to real-life teach pendant	No such provision is available
Different peripherals can be used to communicate with other devices over different channels like wireless, serial TTL, serial RS232, USB etc.	These devices generally use USB or Bluetooth (in some cases both) to communicate with PC.
It communicates with .NET applications using a built-in serial communication library	Specific API is required to communicate with .NET applications

While there are a wide variety of available microcontroller boards, Arduino was used for the development of intended teach pendant. Arduino is an open-source electronic prototyping platform which is cost effective and easily available. Another advantage of using it is the user-friendly programming environment, various libraries and a good online support community. Out of various Arduino boards, Arduino Mega 2560 was best suited for our requirements because of its good processing and a large number of ports. The teach pendant has been integrated with a 20×4 LCD to display user information and a 4×4 membrane keypad to give control inputs, as illustrated in Fig. 4. While there are various types of LCDs available, this specific LCD was selected because it provided a good amount of space to display information and had a reasonable size. The advantage of using membrane keypad over push buttons is that it uses lesser number of ports and is aesthetically and ergonomically better. Also, most of the real-life teach pendants for industrial robots and CNC machines use membrane switches.

The microcontroller in the teach pendant has been programmed to generate characters on the event of a key press for each key in the 4×4 membrane keypad. These characters are transmitted using serial communication to a computer or another controller, as explained next. A prototype of the pendant developed using acrylic sheet is shown in Fig. 5.

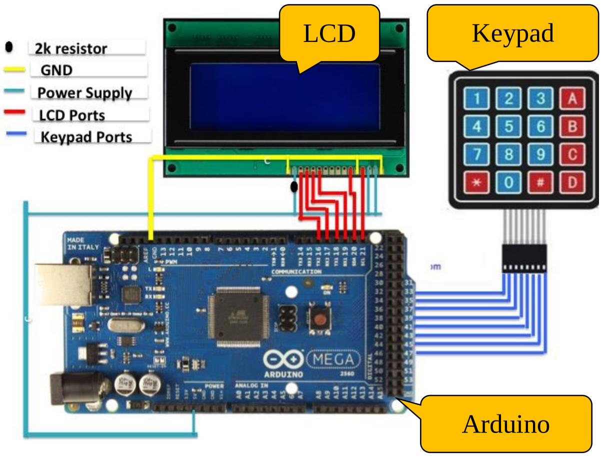


Figure 4. Circuit diagram of the teach pendant

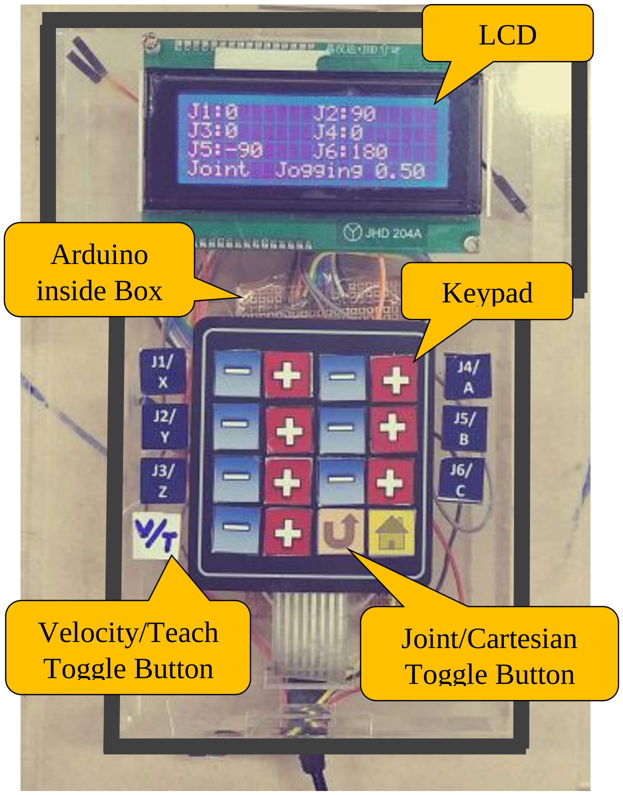


Figure 5. Prototype of the teach pendant

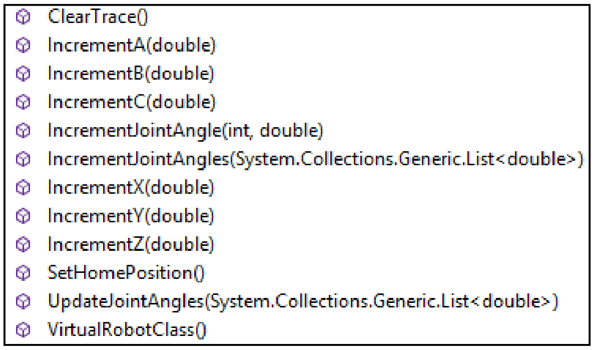
IV. Applications

The teach pendant developed by the authors was made to be generic such that it could be used to control virtual robots on a computer or other real robotic devices running from any microcontroller, whose integration has been reported below.

A. *Virtual Robots*

One of the main purposes of developing this teach pendant was to program robots in VRM. The standard VRM software application [13] was modified to act as a server in the form of a Dynamic Linked Library (DLL). The server has several functions/methods that can be called/invoked from other software applications, as shown in Fig. 6. Note that A, B and C are used for roll, pitch and yaw, respectively.

To integrate teach pendant with the server, a client software application was developed using C#, which acted as a wrapper between the two. The client software gets characters from the teach pendant and based on the mode selected (Joint jogging or Cartesian jogging, as explained next), the server is updated with the new values. The server then updates the configuration of the virtual robot in VRM. An overview of the integration of the teach pendant with VRM is shown in Fig. 7.



```
ClearTrace()
IncrementA(double)
IncrementB(double)
IncrementC(double)
IncrementJointAngle(int, double)
IncrementJointAngles(System.Collections.Generic.List<double>)
IncrementX(double)
IncrementY(double)
IncrementZ(double)
SetHomePosition()
UpdateJointAngles(System.Collections.Generic.List<double>)
VirtualRobotClass()
```

Figure 6. Methods/functions of VirtualRobotClass in server application

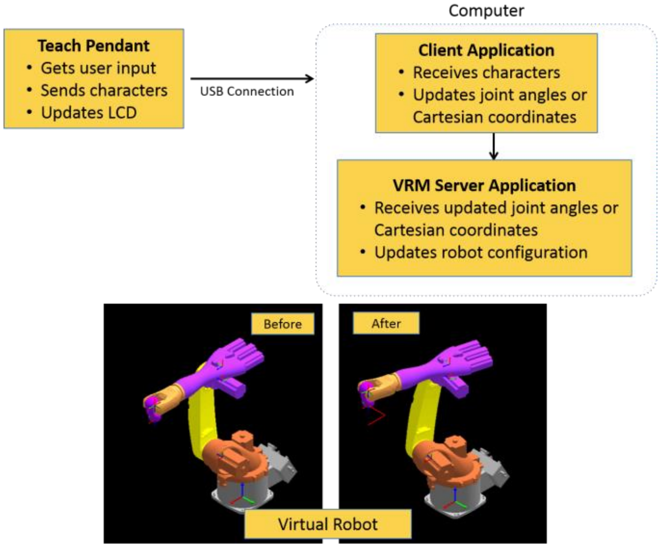


Figure 7. Integration of teach pendant with VRM

The teach pendant primarily operates in two modes:

- Joint Jogging Mode: The teach pendant has a set of two keys for incrementing and decrementing each joint angle. Based on the input received from the user, the joint angles are updated by the client. Subsequently, the robot configuration is changed in the server application. Joint information is constantly updated on the LCD. User can also change the speed of robot by pressing keys meant for the same, as shown in Fig. 5. Besides this, robot motion can be visualized when two joints are being simultaneously updated (i.e., when respective joint keys are pressed together).

- Cartesian Mode: This mode allows users to jog EE along XYZ axis and rotate about them. In this mode, three sets of keys update Cartesian coordinates and the other three update the RPY angles. On the event of increment or decrement of Cartesian coordinates or RPY angles, the teach pendant sends corresponding characters to the client, which in turn calls related function in the server by passing the changes in the coordinate or angle as a parameter. The implementation of the kinematics for the joint and Cartesian-level jogging are already discussed in Section II. In addition, a feature in the teach pendant allows any trajectory of the robot motion be recorded by the client and can playback the same motion in the VRM.

B. *Hardware integration*

Besides using Teach Pendant as control unit for virtual robots, it has also been used to control some real robots in real time:

- Stewart Platform is a 6-DOF parallel manipulator. The teach pendant has been used to control an Arduino based Stewart platform prototype developed at IIT Delhi. The controllers of teach pendant and the platform use TTL serial communication, as shown in Fig. 8. Teach pendant is used to update Cartesian coordinates and RPY angles by transmitting characters serially to the controller of the platform. Based on the received characters, controller of the platform updates the task space parameters. It then calculates the new leg lengths or translations in the linear actuators of the platform using inverse kinematics of Stewart platform already programmed in its controller. Thereafter, the robot controller updates the configuration of the platform.

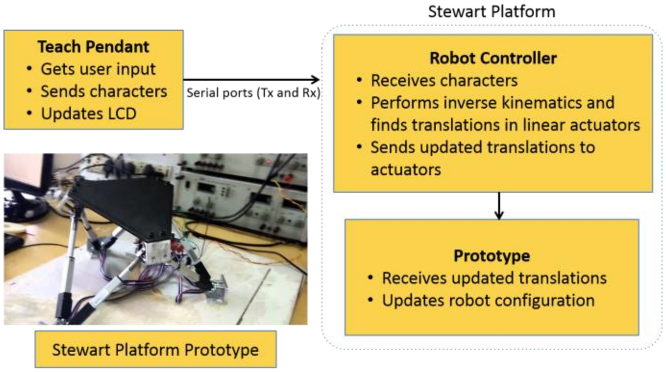

Stewart Platform
Teach Pendant
• Gets user input
• Sends characters
• Updates LCD
Serial ports (Tx and Rx)
Robot Controller
• Receives characters
• Performs inverse kinematics and finds translations in linear actuators
• Sends updated translations to actuators
Prototype
• Receives updated translations
• Updates robot configuration
Stewart Platform Prototype

Figure 8. Integration of teach pendant with Stewart platform prototype

V. CONCLUSIONS

A ‘Teach Pendant’ has been developed by the authors to control virtual robots and suitable hardware. It uses an Arduino based microcontroller board which is commonly available and has a good eco-system for its support and help. It is integrated with Virtual Robot Module (VRM) of RoboAnalyzer software to control virtual robots, where it can perform joint and Cartesian-level jogging. In addition, a real prototype of a Stewart Platform was also controlled using the developed pendant. The complete set of instructions, components list, circuit diagram, client and server applications for virtual robot simulation are available at http://www.roboanalyzer.com. Further it is planned to include features to teach position and motion to virtual robots such as linear, PTP (point to point), circular, etc., hence making it a full-fledged teach pendant.

ACKNOWLEDGMENT

The fabrication of the teach pendant body was done at Moonlit Industries, Sirhind, Punjab, India. Their timely support is deeply acknowledged.

REFERENCES

[1] S. K. Saha, *Introduction to Robotics*, 2nd Edition, New Delhi: McGraw-Hill Higher Education, 2014.

[2] http://www.robodk.com (accessed in May 2016).

[3] https://www.plm.automation.siemens.com/en_us/products/tecnomatix/manufacturing-simulation/robotics/robcad.shtml (accessed in May 2016).

[4] http://www.coppeliarobotics.com (accessed in May 2016).

[5] C. S. Tzafestas, N. Palaiologou, and M. Alifragis, “Virtual and remote robotic laboratory: comparative experimental evaluation,” *IEEE Trans. on Education*, vol. 49, no. 3, pp: 360-369, Aug. 2006.

[6] R. M. Parkin, “Control of robot arm with virtual environment via the internet,” *Proc. of the IEEE*, vol. 91, no. 3, pp: 422-429, March 2003.

[7] F. Torres, F. A. Candelas, S. T. Puente, J. Pomares, P. Gil, and F. G. Ortiz, “Experiences with virtual environment and remote laboratory for teaching and learning robotics at the university of Alicante,” *International Journal of Engineering Education*, vol. 22, no. 4, Aug. 2006.

[8] F. A. Candelas Herías, C. A. Jara Bravo, and F. Torres Medina, “Flexible virtual and remote laboratory for teaching Robotics,” *Current Developments in Technology-Assisted Education*, 2006.

[9] Y. Jan, S. Hassan, S. Pyo, and J. Yoon, “Smartphone Based Control Architecture of Teaching Pendant for Industrial Manipulators,” *in Proc. 4th Int. Conf. on Intelligent Systems Modelling & Simulation, IEEE*, 2013.

[10] C. Mateo, A. Brunete, E. Gambao, and M. Hernando, “Hammer: An Android based application for end-user industrial robot programming,” *in Proc. 10th Int. Conf. on Mechatronic and Embedded Systems and Applications, IEEE/ASME*, 2014.

[11] J. Denavit and R. S. Hartenberg, “A Kinematic Notation for Lower pair mechanisms based on Matrices,” *ASME Journal of Applied Mechanisms*, vol. 22, no. 2, pp. 215 - 221, 1955.

[12] J. Bahuguna, R. G. Chittawadigi and S. K. Saha, “Teaching and Learning of Robot Kinematics Using RoboAnalyzer Software,” *in Proc. Int. Conf. on Advances in Robotics*, 2013.

[13] R. O. M. Sadanand, R. G. Chittawadigi and S. K. Saha, “Virtual Robot Simulation in RoboAnalyzer,” *in Proc. 1st Int. and 16th National Conference on Machines and Mechanisms*, 2013.