

Virtual Robot Labs for Programming Industrial Robot Course

Dr. Maged Mikhail, Purdue University Northwest

Dr. Maged B.Mikhail, Assistant Professor, Mechatronics Engineering Technology Ph.D., Electrical Engineering, Tennessee State University, Nashville, Tennessee, August 2013. Dissertation title: "Development of Integrated Decision Fusion Software System For Aircraft Structural Health Monitoring" M.S., Electrical Engineering, Tennessee State University, Nashville, Tennessee, May 2009. Thesis title: "Development of Software System for Control and Coordination of Tasks among Mobile Robot and Robotic Arm." B.S., Electrical Engineering University of El Mina Cairo, Egypt, May 2001.

Virtual Robot Labs for Programming Industrial Robot Course

Abstract:

Implementing robotics systems requires mastery of software, and hardware components, which can only be obtained through extensive hands-on training using very expensive industrial robots, or simulation software. Traditionally this type of software was only available at high cost through robotics manufacturers, and was proprietary to their products. However recently, more open source options have become available for 3-D robotics simulation software. One of these applications called “RobotRun” was developed by Michigan Technical University in collaboration with Bay de Noc Community College. [1] This software was developed specifically for incorporation into an academic curriculum to give institutions an opportunity to offer training in situations where robots cannot be afforded, or to augment training in limited access scenarios.

This paper will explore the implementation of the “RobotRun” software into an existing robotics course as a “Virtual Lab” component. This “Virtual Lab” is then run in parallel with lecture, and a hands-on lab, reinforcing weekly concepts. The course “Programming Industrial Robots” is offered as an elective, and as course two of a four course robotics concentration for the Electrical Engineering Technology (EET), and Mechatronics Engineering Technology (MCET) Bachelor’s degrees offered by The College of Technology at [University].

Introduction:

The “Programming Industrial Robots” course offers students hands on experience with FANUC brand robots, an industry leader in robotics. The lab at [University] is composed of four FANUC LR Mate 200iD/4S compact six-axis robots. One unit is a FANUC training unit, which is fenced CERT Cart furnished by FANUC. The three others are fenceless, and are currently table mounted for use in the first course of the Industrial robotics concentration, “Programming Industrial Robots”. The three fenceless units will soon be incorporated into the second course, “Industrial Robotics Integrations”, where they will be used in conjunction with Allen-Bradley PLC’s to operate a scaled packaging line scenario, using conveyors, and various peripheral sensors. After completing these courses, students will then move on to a capstone course tailored to their specific bachelor’s degree. Along with the concentration, students will receive two FANUC certifications, one for material handling, and the other for vision systems.

With a typical class size being twenty students, it is necessary to divide the students into two lab sections to increase contact time with the robots. Even still, with only four robots, students must be divided into lab groups, which limits hands-on time with the robots to one half, or one-third of the lab time, depending on group size. With this limited contact, it is inherent to incorporate a simulation program as a familiarization to the teach pendant, and robot commands, and as concept reinforcement for lecture topics. The “RobotRun” software, and the labs developed for it are run in parallel with the hands-on FANUC training labs, so students are able to reinforce skills, and gain familiarity, but do so at their own convenience. For example, a hands-on lab is

performed to establish a tool frame with the FANUC robot, and that same week students are assigned a “RobotRun” lab in which they must create a tool frame, reinforcing the hands-on concepts, but also challenging them to repeat the task within a different system.

Software Overview:

RobotRun is open source emulation software written using Processing, a Java based programming language. OpenGL is used to facilitate the real-time 3D visualization of the industrial robot arm, which was modeled in a CAD environment. The software uses an inverse kinematics algorithm to calculate the proper orientation of each segment of the manipulator when given a user-specified position and orientation for the end effector when using the world frame coordinate system. This functionality is accomplished using what is called the Jacobian pseudoinverse approach, and implemented using a function within the Apache Commons Math Library [2].

The software screen features a teach pendant with controls consistent to that of FANUC brand pendant. Using the mouse, the user can manipulate the camera view 360° around the robot, by holding the right mouse button down, and panning the screen. The track wheel allows the user to zoom in and out as, well giving full access to the entirety of the work envelope.

The program offers a number of different end effectors, which can be interchanged by selecting Ctrl + E on the keyboard. The different end effectors which come pre-loaded in the program are suction cup, gripper, pointer, glue gun, and welder. This gives the student the ability to explore the most commonly used end tools found in industry.

Scenarios such as pick and place, gluing, and welding come pre-loaded in the software, so the student can practice. Once the student is comfortable, these existing scenarios can be modified, or a new scenario can be created from scratch. Fixtures, and objects can also be created, or imported into a program, with which the robot, and end effector can interact. The objects which are pre-existing such as tables, boxes, parts etc. can be imported to a blank scenario, or an object can be created from scratch within a scenario by the user defining its parameters.

Programming is identical to a physical industrial robot, and is accomplished by teaching the robotic arm through points which are assigned a movement type, such as linear, joint, or circular, and a speed. Once the desired points, and instructions, such as open or close gripper etc. are recorded, and tool and user frames are established, the program can be saved. This program can then be accessed later, and run automatically within the scenario, in the same way which an industrial robot would be used. These basic features, and their implementation are discussed in the lab description section, with detailed instruction included in the actual laboratory manual.

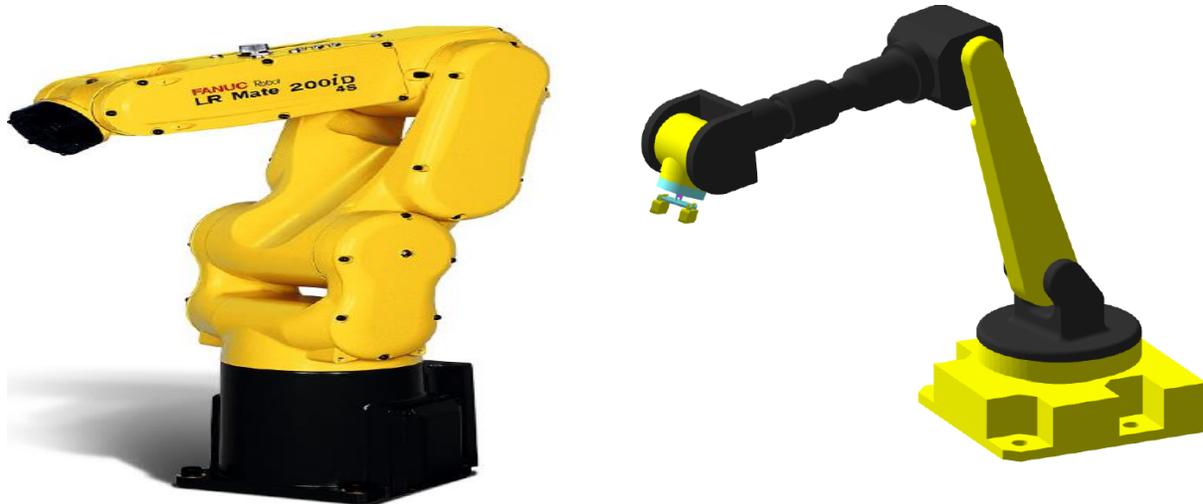


Figure 1. Comparison of FANUC LR mate robot (*left*) with RobotRun Simulation (*right*).

Software Simulation Advantages.

As previously discussed, the use of simulation can offer some distinct advantages, especially when the software comes at no cost to the University, or the students. The obvious main advantage is concept reinforcement, and overall exposure to the basic constructs, and commands of a typical industrial robot. Tasks such as jogging the robot, teaching frames, creating/modifying, and executing programs are essentially identical in the FANUC teach pendant, and the “RobotRun” teach pendant. This creates not only reinforcement, but a basal level of continuity which is necessary for an educational tool. Often the most challenging aspects of switching between a hands-on work environment, and a simulated environment, is the discontinuity between operations done by hand, and those done within a software program. Simulation software also has some notable disadvantages. First and foremost is that a simulation, even a really good one cannot completely emulate a real-world hands-on experience. This leads to the discontinuity mentioned previously. For instance, when using a FANUC controller, it is required that the operator hold a dead-man switch, and the shift button in order to jog the robot in teach mode. The software has the shift button, but unlike the FANUC controller it is not momentary, and once pressed, it stays pressed and you are free to jog the robot. The dead-man switch is absent from the “RobotRun” controller, and this is a fundamental feature of programming using a teach pendant, and takes time getting used to it.

Virtual Robot Labs Using “RobotRun”

The following section explores the first three virtual labs which cover the fundamental aspects of the software, and augment the physical robotics programming. The topics are learning to create a scenario, pick and place, tool frame. After completing these labs, students will have the fundamental tools with which to explore more advanced programming methods within the “RobotRun” software, and with physical robots. The labs are designed to augment topics covered in class, and are currently run in parallel with the hands-on portion of the lab. However, the software stands on its own, and the virtual labs are meant to be conceptual, and could certainly be used for an abstract approach without hands-on laboratory access.

Virtual Lab #1 – Creating a Scenario.

Objectives:

- *Students will learn to design a scenario, and implement objects.*
- *Students will learn to edit object dimensions.*

This lab is designed to give students the ability to create, and upload existing and newly created scenarios, which is the most fundamental aspect of the software. After creating a scenario student are tasked with creating an object, editing its dimensions and color, and positionality within the work space through the edit object function. The next task is to import a fixture such as a table, which already exists within the software, then place it at a position which is relevant to the desired operation of the robot. The lab also covers updating default parameters for objects, and fixtures. Once completed students will have the requisite skills necessary to complete the remainder of the labs, and complete programming tasks.

Creating a Scenario.

Step 1. Create a Scenario. *A default, or previously created scenario can also be modified again by selecting load, and the desired scenario.*

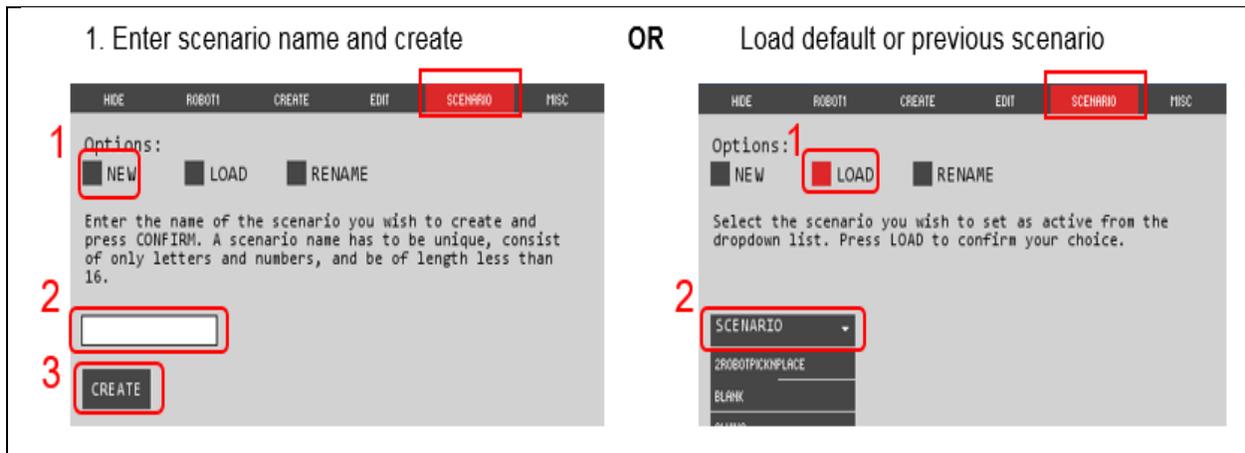


Figure 3. Creating a Scenario.

Step 2. Create an object and table within the scenario. *Existing objects and tables can also be imported from another scenario.*



Figure 4. Creating an Object.

Step 3. Edit the object position and rotation. *Ensure that update default is selected as indicated in number 4 of the figure. This allows the object to be restored to a default position if it is moved.*

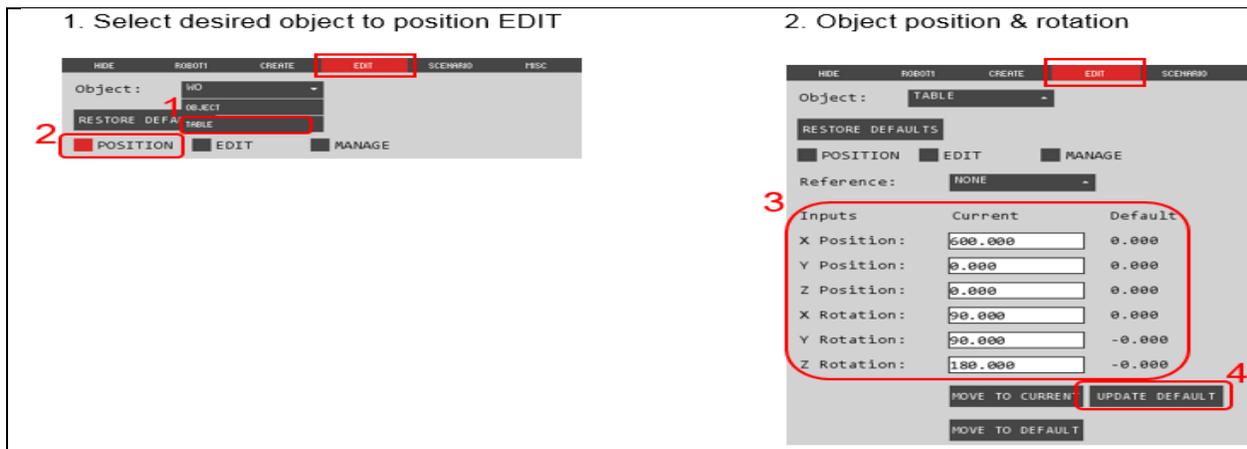


Figure 5. Editing Object Position and Rotation.

Step 4. Edit the object Dimension & color

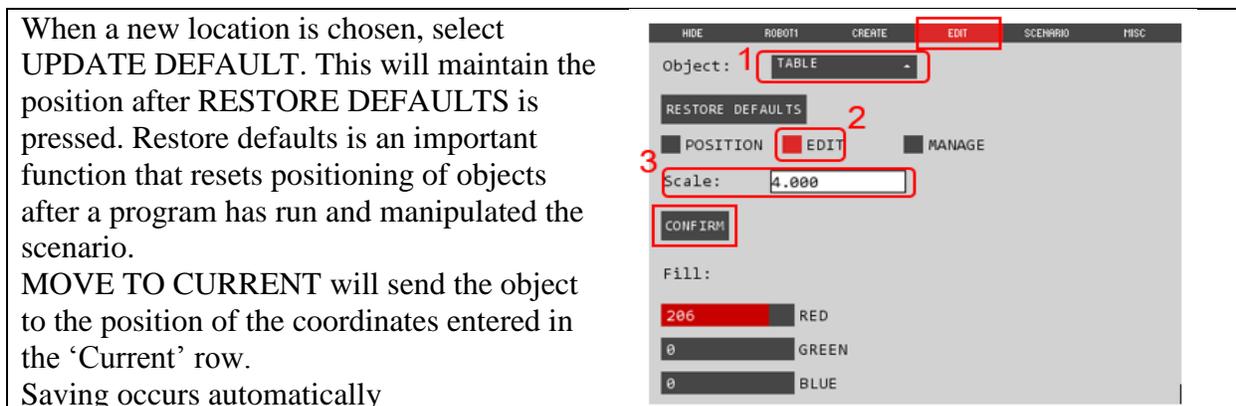


Figure 6. Edit Object Dimension and Color.

Shown below is an example created by a student for Lab 1. This is turned in as part of the weekly Lab report requirement. The program itself is written as part of Lab 2 Pick and Place program.

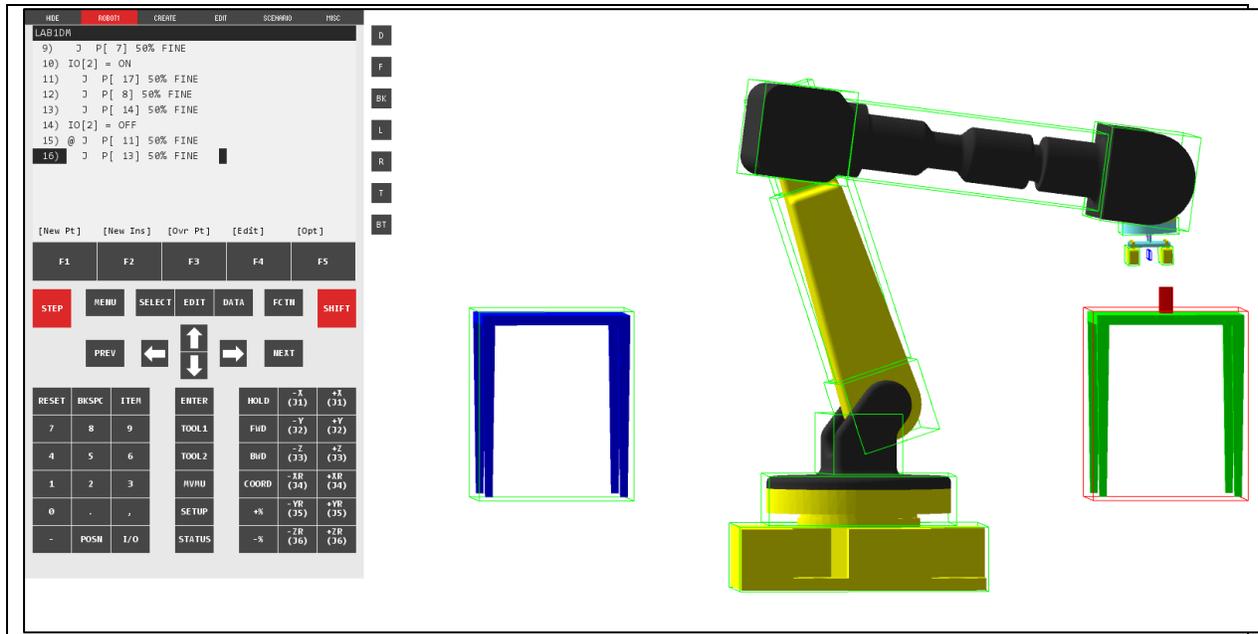


Figure 7. Finished Scenario. *Student Example*

Virtual Lab #2 – Basic Jogging with Pick and Place Program.

Objectives:

- *Students will learn to jog the robot in JOINT coordinate frame.*
- *Students will learn to write a pick and place program.*

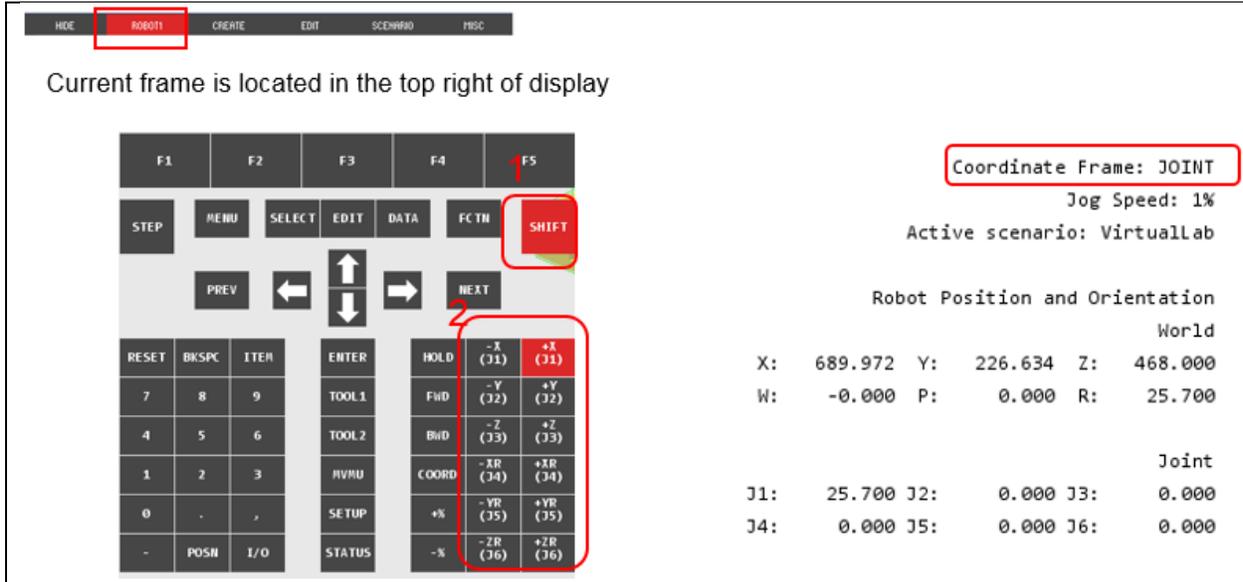
This lab explores jogging the robot, as well as key features of the teach pendant such as the SHIFT, COORD, and FWD buttons. Additionally, the student is shown how toggle through end effectors, add instructions, and record the robot position after completing a desired movement. These components are integrated into the task of completing the pick and place program which comes as a preloaded scenario in the software. This lab can also be used to reinforce topics from Virtual Lab 1, by requiring students to create a new scenario, import two tables, and create the object to be moved. Next the student must place the objects and fixtures in an appropriate position within the work envelope, in order to accomplish the required programming task.

Jogging the Robot.

SHIFT + any coordinate buttons circled in number 2 of the figure.

Changing the Coordinate Frame.

Without SHIFT enabled, press COORD to toggle.

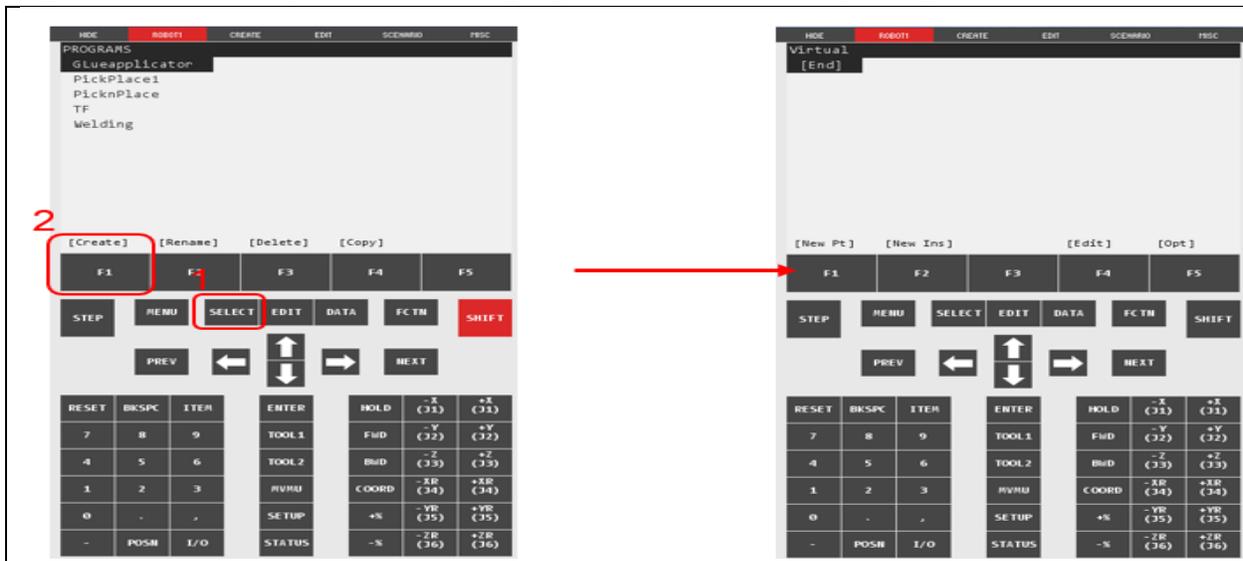


Change manipulator jog speed using +%, -% keys.

Figure 8. Teach Pendant Controls.

Writing a Pick and Place Program.

- Step 1. Load the appropriate scenario and restore defaults in EDIT if necessary
- Step 2. Toggle the end-effector by using CTRL + e. Select GRIPPER. This is displayed in the top right corner.
- Step 3. CTRL + T will reset the robot to its default position.
- Step 4. Use SELECT key to open a program window.
- Step 5. F1 [Create] key to create a new program. Once a name is given, press ENTER.



The screen above is where the program is written.

Figure 9. Programming Screen.

Step 6. Jog the robot to first desired point.

Step 7. SHIFT+ F1 [New Pt] to record the point in the program.

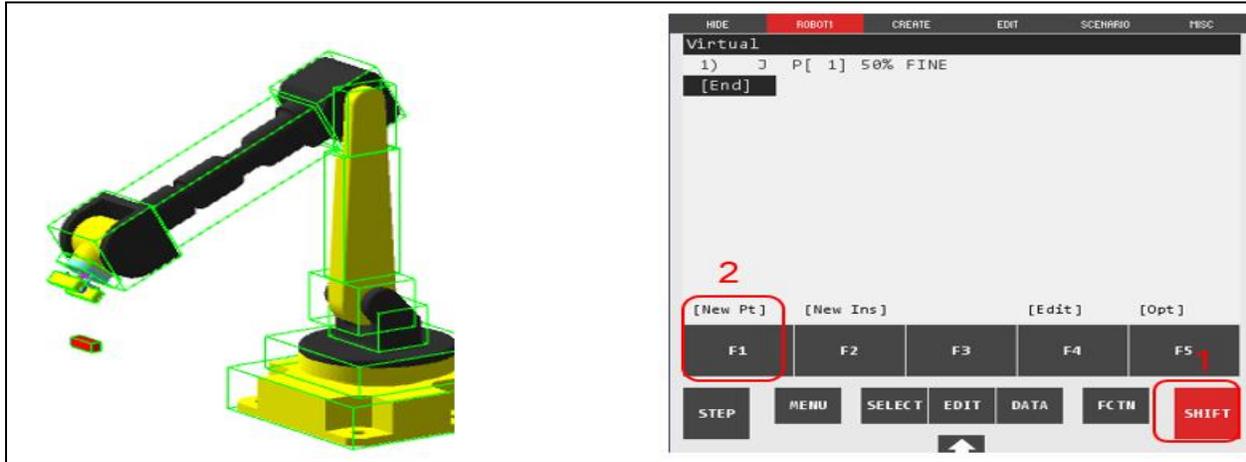


Figure 10. Recording a New Point.8. To secure an object in the gripper.

a. Jog until the object is between both jaws of gripper. The jaws will highlight GREEN and the object BLUE.

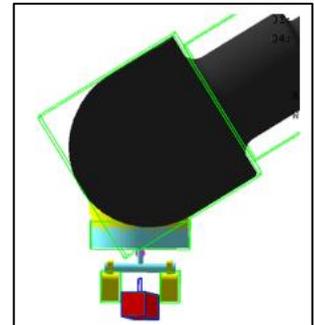
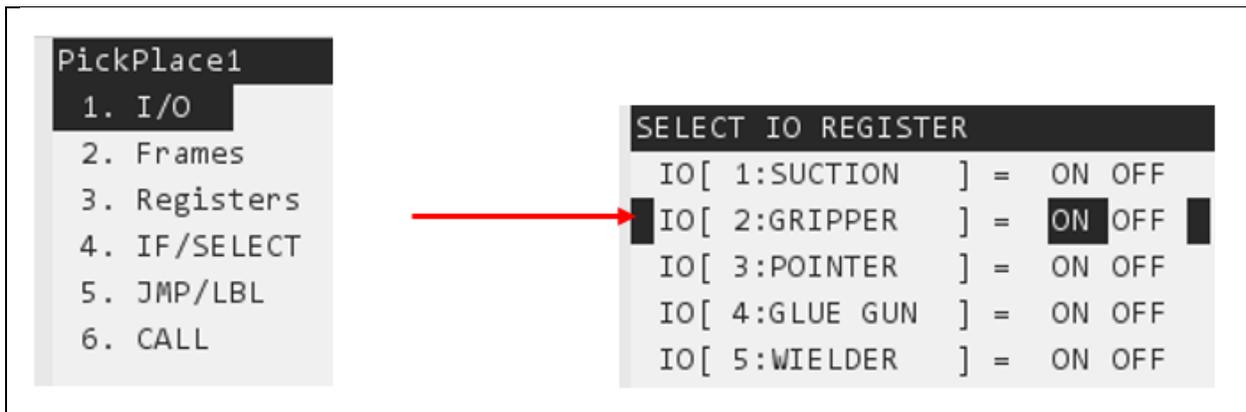


Figure 11. Successful Gripping of an Object.



Note: Gripper ON = jaws closed, Gripper OFF = jaws open.

Figure 12. Gripper I/O Instructions.

d. Select IO: GRIPPER [ON]. This will NOT close the jaws, this will only write the instruction.

e. There are two methods for closing the gripper

i. CTRL + F will toggle the jaws OPEN and CLOSED.

- ii. Highlight the line with the I/O instruction, press STEP, SHIFT, and FWD to run that single line of code.

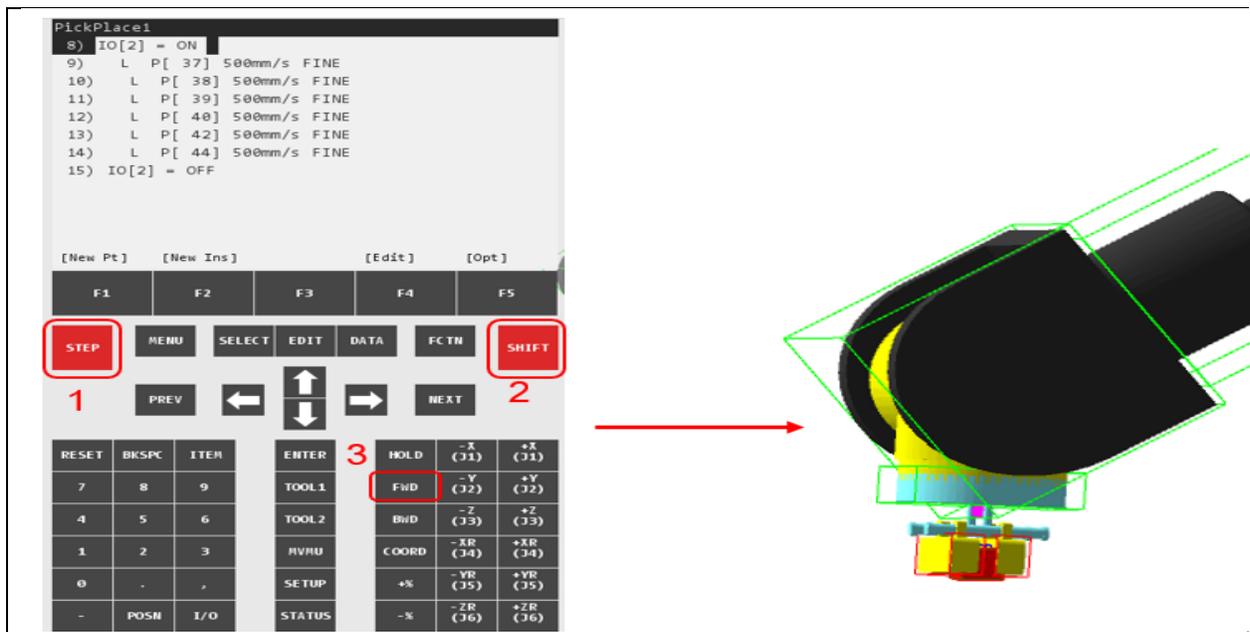


Figure 13. Closed Gripper.

- f. Repeat this method to write the instruction for release of the object.

Step 9. To test the program at any point, use SHIFT + FWD to run the entire program without pause.

- a. With the addition of STEP, the program will only run a single line at a time.

10. After the program has run, restore the scenario defaults under the EDIT tab & CTRL + T to reset the arm to the home position.

Virtual Lab #3– Tool Frame.

Objectives:

- *Students will learn to create a tool frame using the three-point method.*
- *Students will learn why tool frames are important to robot programming.*

Lab three is used to explore the very important element of establishing a tool frame for use within a program. The student will be shown how to access the frames menu, and then select the frame mode to be used, i.e. frame, or tool. Next the tool frame number will be selected, as well as the method used to establish the frame. The three-point method is shown in this lab, where three approach points are established relative to a fixed position above the work piece. The first approach point is established just above the object, with the second and third approaches being recorded at positions orthogonal to the first. The three points should converge on a single point, but singularity between points two and three need to be avoided. If these constraints are not met,

the program will not accept the frame as valid. After establishing the frame, students will select the frame as active, and jog the robot within the frame.

Creating Tool Frames.

Step 1. Create a new scenario that involves a cylinder and table with the characteristics below.

a. Create a table.

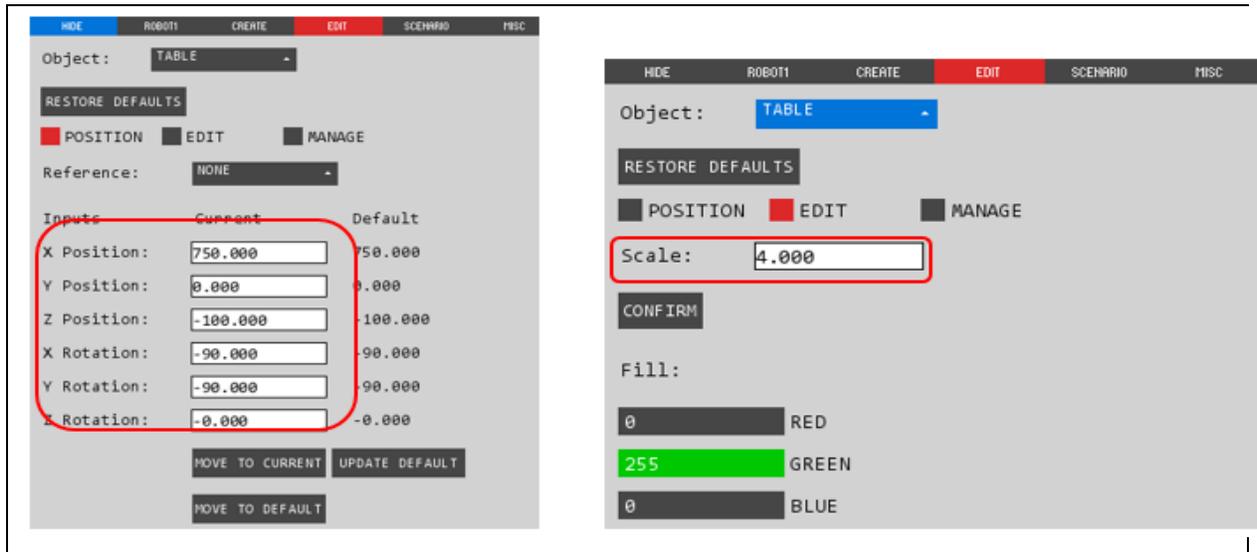


Figure 14. Creating the Table.

b. Create a cylindrical object.



Figure 15. Creating the Cylinder.

Tool Frame Setup.

Step 1. In Main Menu, select FRAMES.

Step 2. Select TOOL FRAME.

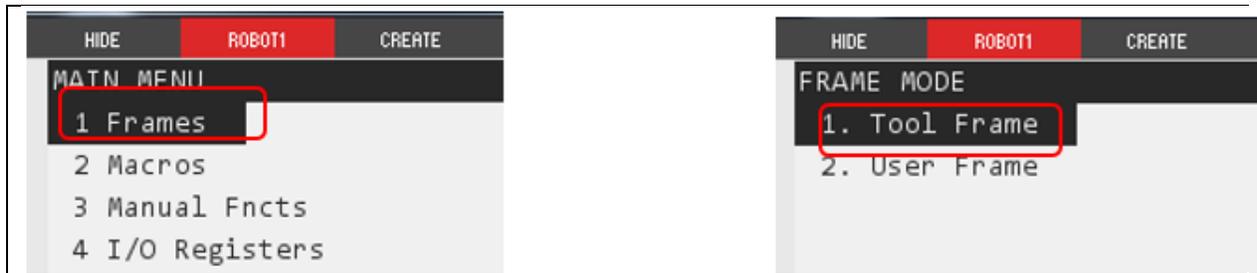


Figure 16. Selecting the Tool Frame.

Step 3. Use tool number 4 or greater. *Although it is not labeled tool frame 4 is used for the glue gun. Overwriting it will cause problems using this end effector.*

Step 4. Select THREE POINT METHOD.

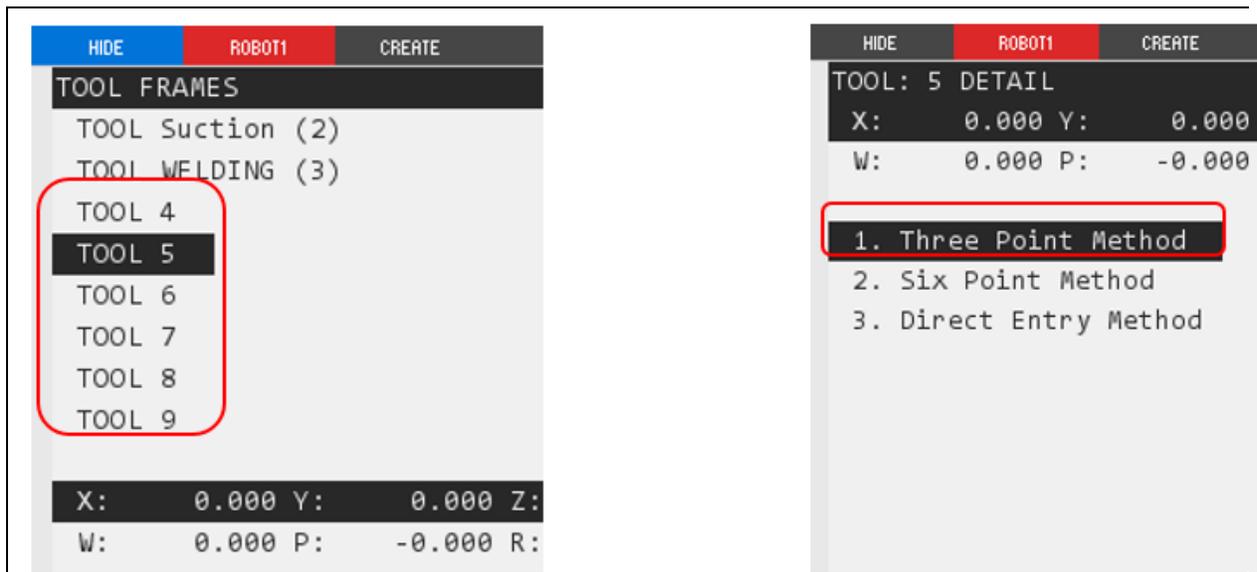


Figure 17. Tool Frame Method.

Approach Points.

Step1. With FIRST APPROACH POINT highlighted. Jog the arm and tool piece directly over the object

- a. Use the suction or pointer end effector, (CTRL + E) to cycle through the end effectors.
- b. Use SHIFT + F5 [Record] to save the point.

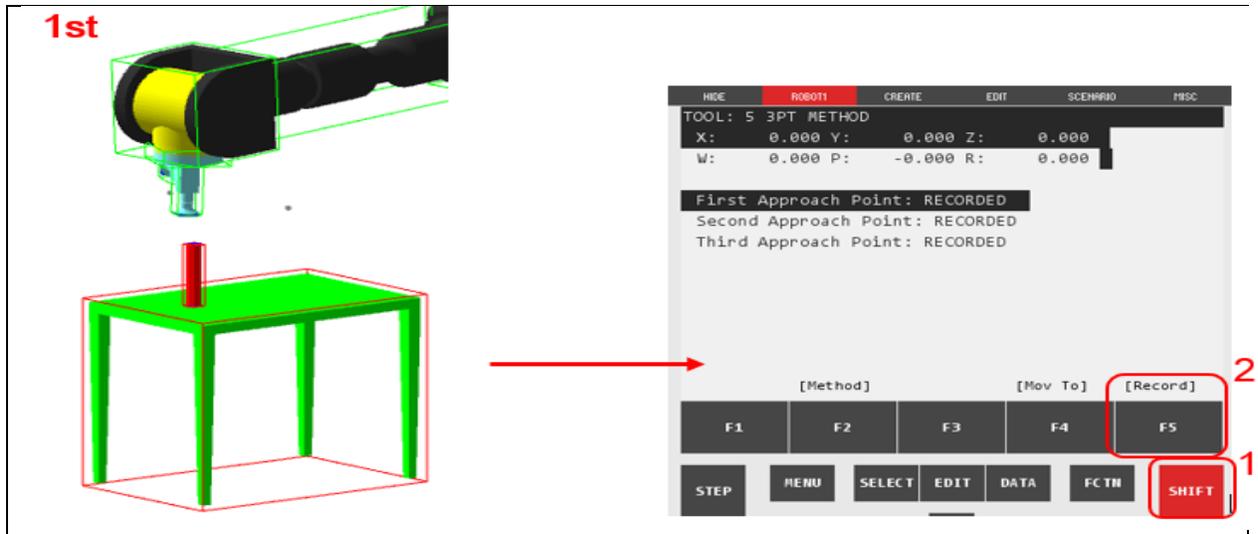


Figure 18. First Approach Point

Step 2. For the SECOND APPROACH POINT. Jog the arm and tool piece perpendicular & 90° in relation to first point.

Step 3. For the THIRD APPROACH POINT. Rotate arm and tool piece to a point at least 90° from second point.

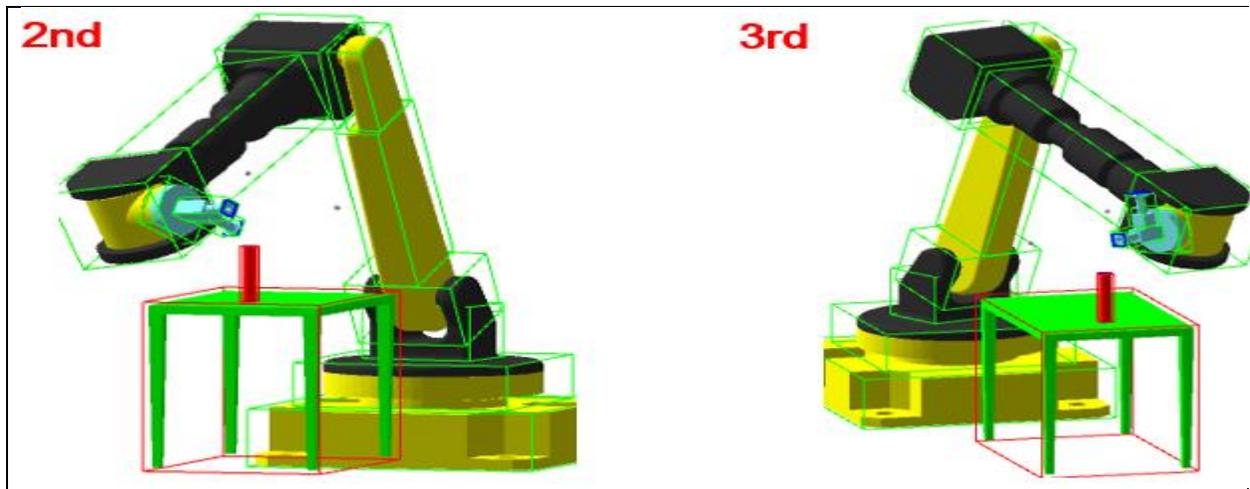


Figure 19. Second and Third Approach Points.

Step 4. To cycle, through the recorded points, highlight the desired approach point and press SHIFT + F4 [Mov To].

Note: To change the active frame at any time, use SHIFT + COORD.

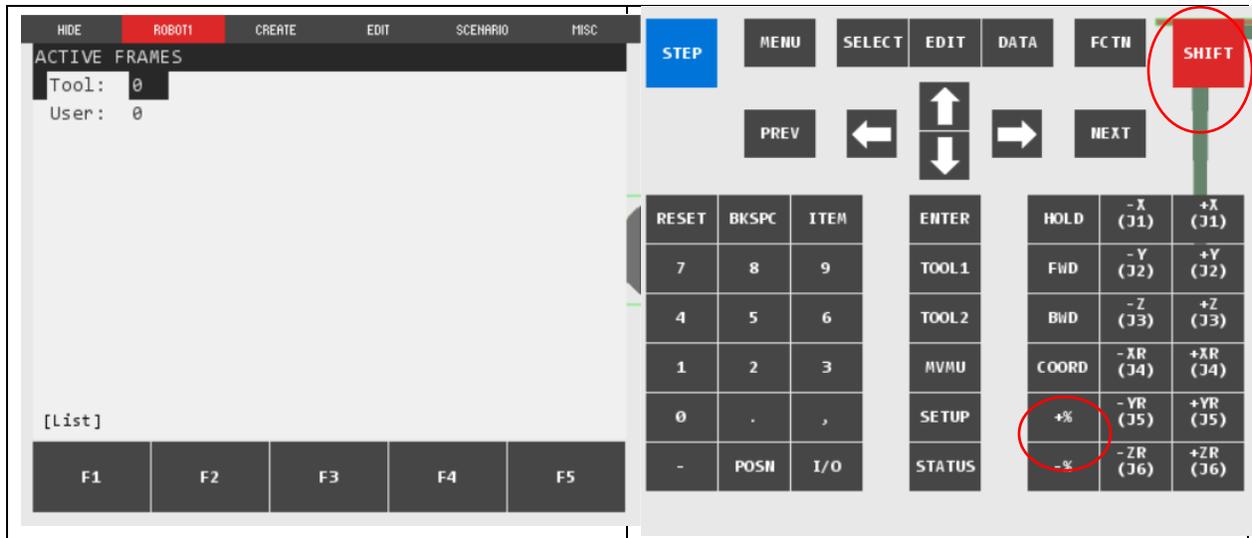


Figure 20. Active Frames.

Conclusions.

With increasing automation, and robotics use in an ever-expanding range of industries, it is imperative that educational institutions can facilitate technical training in these areas. As discussed earlier these tools are expensive, and almost always proprietary. For this reason, there exists a gap in the need for training, and the accessibility of the tools required to do so. Open source software simulation such as “RobotRun” has a number of distinct advantages which can be used as part of a hybrid program in conjunction with hands-on robot experience, or possibly even as stand-alone training vehicle in no-access situations. With access limitations, augmenting physical access to the robotics equipment is a crucial step in ensuring student educational outcomes. The expected outcomes of using this type on software in conjunction with a traditional hands-on environment are; increased concept reinforcement with better overall comprehension, increased class preparedness, and more efficient use of the physical equipment. The labs discussed in this paper give students a foundational tool kit which can be applied universally to any number of industrial robotics programming environments. The final two labs are discussed in detail in a subsequent paper.

Acknowledgements.

The Author would like to thank Dr. Aleksandr Sergeyev¹ and Dr, Scott for their effort to develop this software and make it free for use.

References.

- N. A. S. K. J. H. V. D. S. P. Aleksandr Sergeyev¹, "Development of the Open-Source “RobotRun” Robotic Simulation Software," American Society for Engineering Education, 2017.
- S. Y. Parmar, "Research and Development of Industrial," Open Access Master's Report, Michigan Technological University, 2017.
- Global Manufacturing Scorecard," 10 July 2018. [Online]. Available: <https://www.brookings.edu/research/global-manufacturing-scorecard-how-the-us-compares-to-18-other-nations/>