



Incorporating Visual Components Simulation Software with the Programming Industrial Robots 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.

Mr. Sandeep Bharti, Purdue University

Dr. Ali Alavizadeh, Purdue University Northwest

Dr. Ali Alavizadeh is an Assistant Professor of Industrial Engineering Technology at Purdue University Northwest (Hammond, IN). Previously, he taught at Purdue University Fort Wayne (Fort Wayne, Indiana), The George Washington University (Washington, DC), and Morehead State University (Morehead, KY) in the areas of Industrial Engineering Technology, and Engineering Management and Systems Engineering. His industrial experiences include software engineering, systems engineering and analysis, and production optimization for private, governmental, and nongovernmental organizations. His research interests include complex systems modeling and simulation and their application in healthcare and aerospace.

Incorporating Visual Components Simulation Software for Teaching programming industrial robotics course

Abstracts:

A robot programming simulation testing software system that is 'Visual Components' is utilized to put applications together for a real or physical robot without relying upon the genuine machine, subsequently decreasing expenses and time. For some situations, these applications can be moved onto the physical robot without adjustments. Integrating Visual Components (VC) simulation software for Robotic Programming has become essential for the past a few years as it serves comfort to engineers, designers, and customers to understand the actual vision of the robotic system. The Visual Components has more than 200 vendors, as these 200 vendors dispense a variety of robots to work on and to understand the fundamental design of these given robots. Robotics software simulation capabilities are expanding every day, becoming a key part of the integration process for robot integrator with the technical understanding to leverage this technology. Robot simulation software provides many advantages over traditional CAD systems and features much more functionality. The advantages of using simulation are huge such: proof of concept and proof of design, reduced integration costs, and shorter system delivery times. One of the most popular and powerful robotic simulation software with variety of tools in the market is Visual Components software. Integrating Visual Components simulation software for Robotic Programming courses has become essential for students in engineering and technology majors. In this paper, the software overview and capabilities will be discussed. The integration of Visual Components software with the programming industrial robotic course will be addressed. Furthermore, sample of the labs will be introduced. Finally, the student's feedback regarding incorporating visual components software with the programming industrial robots' course will be presented.

Keywords: *Robot simulation software, Visual Components software, industrial robot, 3D CAD data, virtual commissioning, Process Optimization, robot integrator*

Introduction:

The Visual Components (VC) simulation software has become an increasingly essential tool for manufacturing professionals having the responsibilities of layout, planning, and optimizing the work cell design for virtually commissioning the robot and accelerating the time to production [1]. It uses 3D CAD data to create a virtual model of the robot and work cell and simulate the processes and workflows inside and outside the cell. In other words, VC software has become a powerful tool for engineers and planners to evaluate trade-offs and make better decisions.

Integrating VC simulation software for Robotic Programming has become essential for the past a few years as it serves comfort to engineers, designers, customers to understand the actual operations of the robotic system. Robotics software simulation capabilities are expanding every day, becoming a key part of the integration process for robot integrator with

the technical understanding to leverage this technology. Robot simulation software provides many advantages over traditional CAD systems and features much more functionality [5].

Therefore, in a view of above-mentioned facts, the research work was undertaken to find the best probability and application of robotics that can be programmed with a simulation software referred as VC simulation software which could result in efficiently teaching the programming industrial robotics course virtually and also providing a meaningful return on investment for many types of automation projects by saving time, improving productivity, and helping manufacturers to identify opportunities for cost savings.

Accordingly, incorporating Visual Components simulation software for teaching the programming industrial robotics course is considered utmost important as the students can learn and understand all the manufacturing challenges and methods such as planning and optimizing layouts by programming robots and creating working cells/environment simulations for labs and projects as a proof of concept and proof of design.

Overview of Visual Components Software:

Programming a robot with a simulation software is referred to as Visual Components simulation software. Apart from achieving accuracy level, one of the benefits of offline programming, is that the robot is not stopped for programming purposes, but continues production while being programmed offline using the simulation and not the actual hardware. The offline programmed motions are smooth and collision-free, thereby increasing the robot's life and saving energy [5]. The VC software has come a long way over the years. What was once viewed as a technology that was too expensive and complex for most manufacturers is now more advanced, more affordable, and much easier to use. The VC Software provides a meaningful return on investment for many types of automation projects, by saving time, improving productivity, and helping manufacturers to identify opportunities for cost savings.

Robot programming is the process of programming a robot to perform a set of tasks. For industrial robots, programming can be done online, meaning in a production environment, or offline, meaning outside a production environment or using the virtual simulation [4], [5]. Two key methods for robot programming are robot teaching and offline programming as shown in Figure1, 2 & 3.

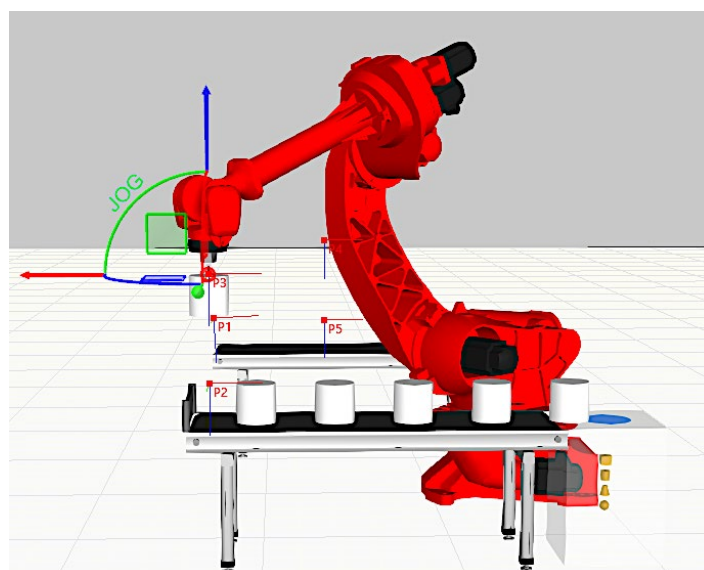


Figure 1. Teaching FANUC robot arm

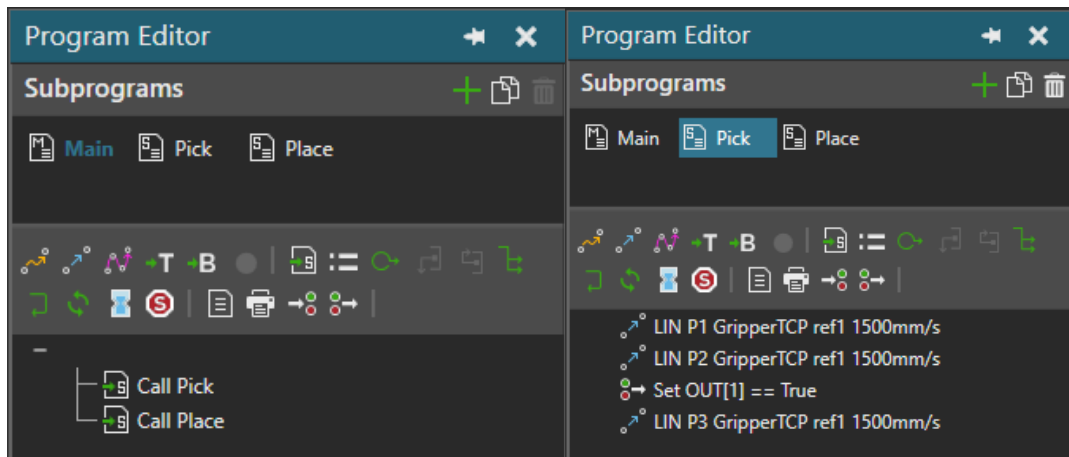


Figure 2 & 3. Offline programming the robot

In the teaching method, a teaching pendant is used for manually moving a robot end-effector to different positions and orientations at each stage of the robot task. Relevant robot configurations were recorded by the robot controller, and a robot program was written to command the robot to move through the recorded postures. It has also been reported that “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” [3, p. 1]. The Visual Components method uses 3D data to create a virtual model of the robot and work cell as shown in figure 4.

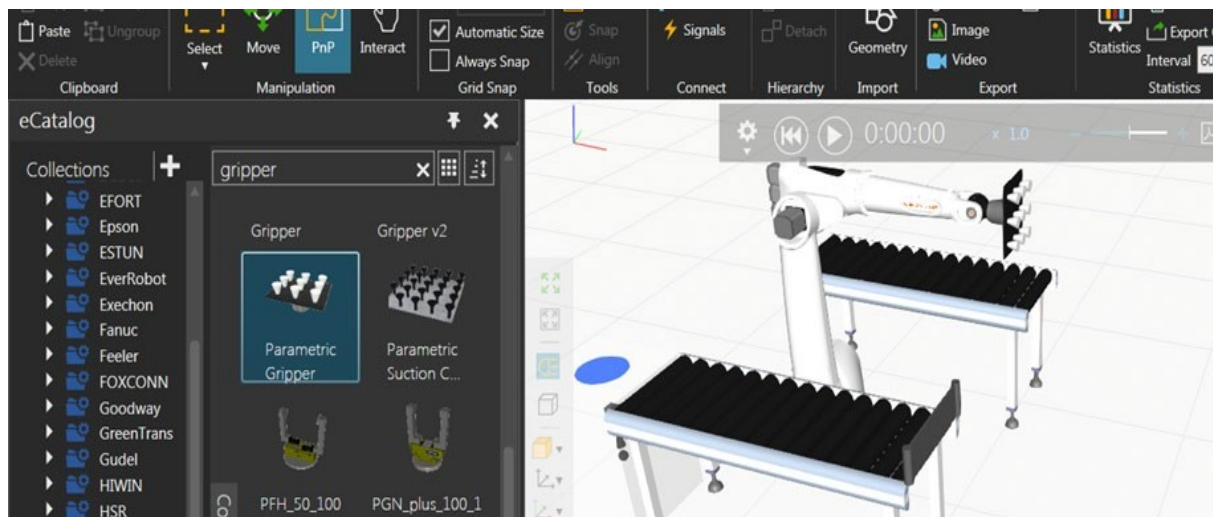


Figure 4. Building 3D Virtual Conveyor Model

Using simulation, it allows the user to virtually teach the robot. Because Visual Components simulation software is a computer-based approach that uses digital models and advanced simulation, it’s much faster and more accurate compared to online programming for many applications [4]. It was stated that “the 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” [3, p. 3]. The Visual Components simulation software provides “Recording Controls” [9], “Drawing Controls” and “Statistics Dashboard” that can be a catalyst for a meaningful return on investment for many types of automation projects, by saving time, improving productivity, and helping

manufactures to identify opportunities for cost savings by analysing several inputs and outputs of the components and providing with a statistics chart [9], an example is shown in figure 5 [9].

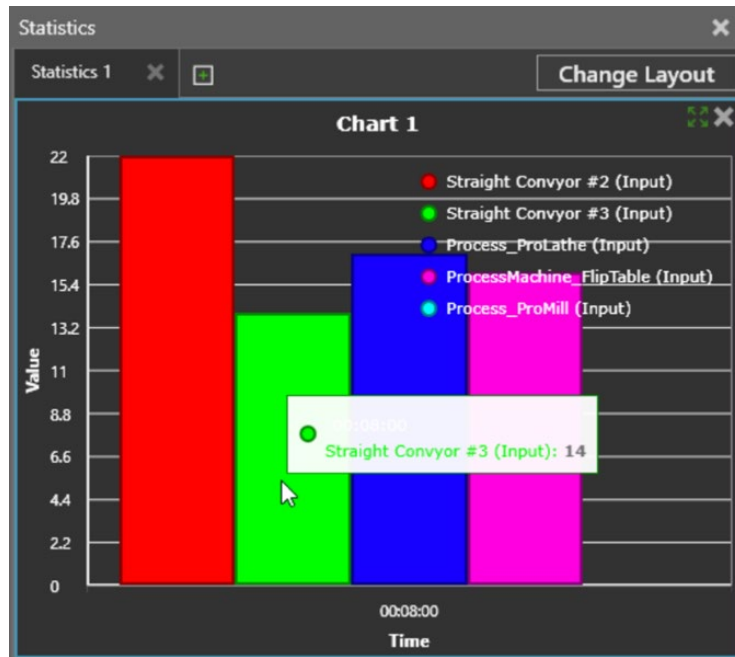


Figure 5. Statistics chart of value-time of different components

The Visual Components software also offers 32 pre-designed demo layouts that are extensive and complex factory floors designs such as car assembly line demo, pick and place in safety cell demo, automated warehouse demo, milling factory demo, etc., which can help student users to learn about the varied work environments, setups, and robotics programming easily [10], examples are shown in figure 6 [10].

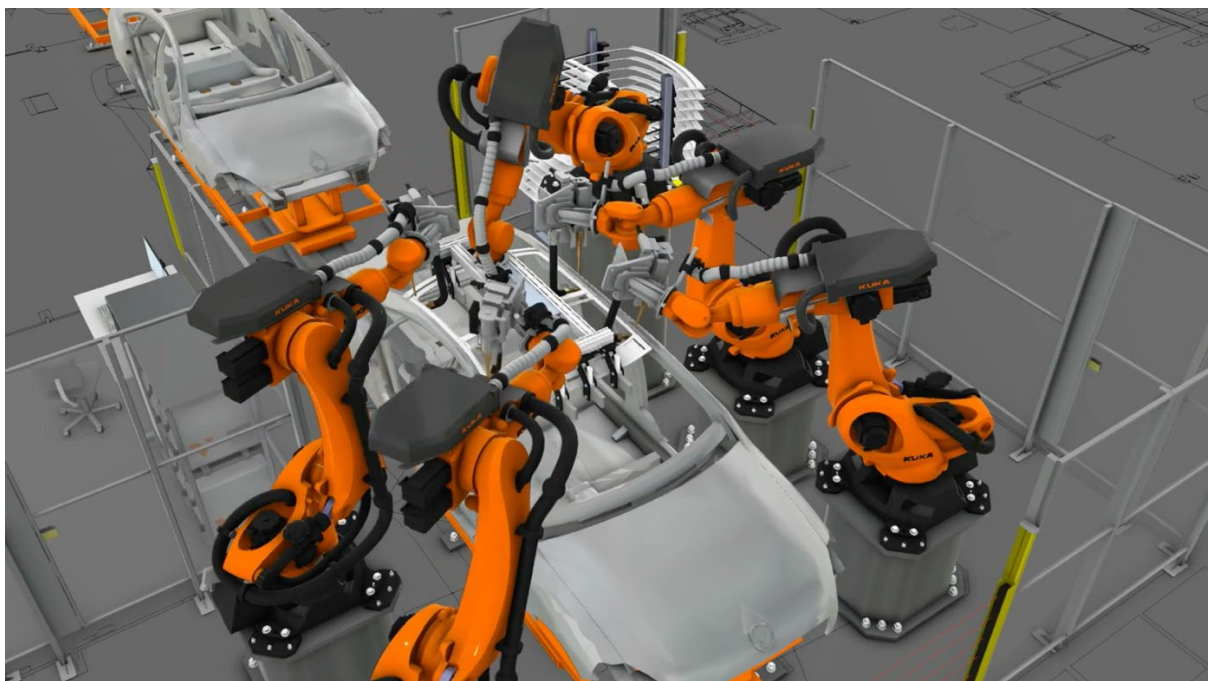


Figure 6. Car Assembly Line demo provided by the Visual Components

Importance of programming industrial robot:

If there is one technological advancement that would certainly make living easy and convenient, undoubtedly the robots would be the answer. Robots are human-like machines capable of doing tasks they are programmed to do. They had shown a significant decrease in human workload especially in industries and increasing safety by handling unsafe works such as welding the metal parts precisely. Designing and writing a simulation program on welding the different shapes on the Visual Components software is easy and resourceful. Therefore, industries can achieve precision in welding the objects and will generate less waste [8], an example is shown in figure 7.

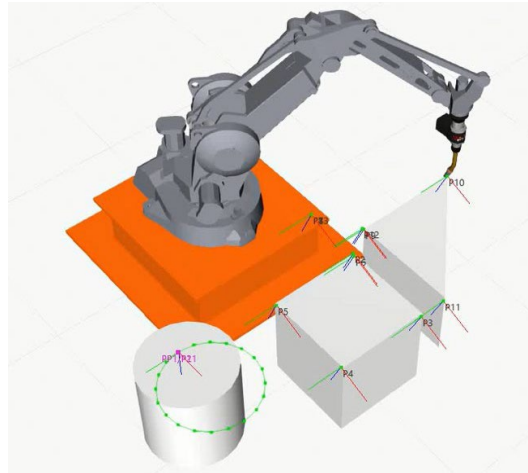


Figure 7. Welding objects with the Welding Torch on the Visual Components

As has been reviewed, the robots are mostly utilized in the manufacturing industry and the type of job that laborers encountered in this type of business is usually repetitive and monotonous. People who do the same thing repetitively for a long period tend to get bored and tired of what they are doing and might arrive to a position wherein they are unwillingly doing the job which could result in lowering in efficient and effectiveness level of work. Also, as human being, we get exhausted so the length of time that we can work is only limited. Considering all these facts, the importance of application of robots was realized. Robots can be set to function for a long time producing a consistent quality product throughout the production process [5]. This would result in an increased number of manufactured products and decrease in the production of defective goods. For example, programming a robot to advance on all sides of a tool frame connected to another component for deburring and polishing applications as shown in figure 8.

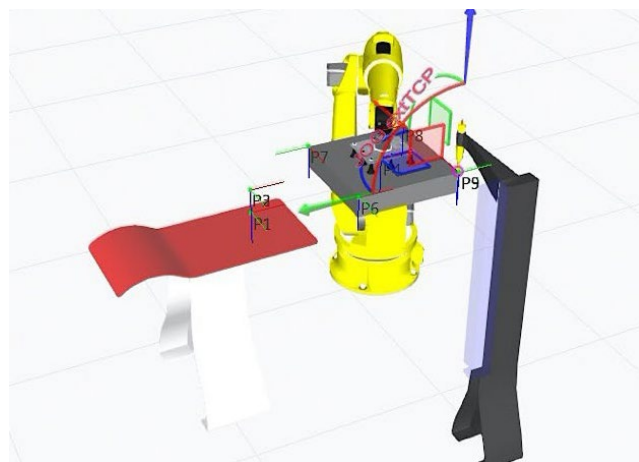


Figure 8. Polishing a component with the tool frame on the Visual Components

Capabilities of Visual Components:

The Visual Components provide us with no less than the 200 vendors, as these 200 vendors dispense us a variety of robots to work on and to understand the fundamental design of these given robots as shown in figure 9.

Robotics software simulation capabilities are expanding every day, becoming a key part of the integration process for robot integrator with the technical understanding to leverage this technology. Robot simulation software provides many advantages over traditional CAD systems and features much more functionality [6].

It is the industries that gained a lot of benefits out of robotics. With the application of Visual Components software, the productivity of the company risen to many folds and making businesses to achieve more profits. In other words, the losses of companies reduced because flawed products are trimmed down to almost none.

The brain of robots where they receive a set of instructions that make them perform tasks automatically is called artificial intelligence or AI [2]. There have been stories showing that these machines have become intelligent enough to think and act independently and overthrow humanity. At present, this is nowhere near to happen since robots nowadays are not capable enough to do tasks without being controlled.

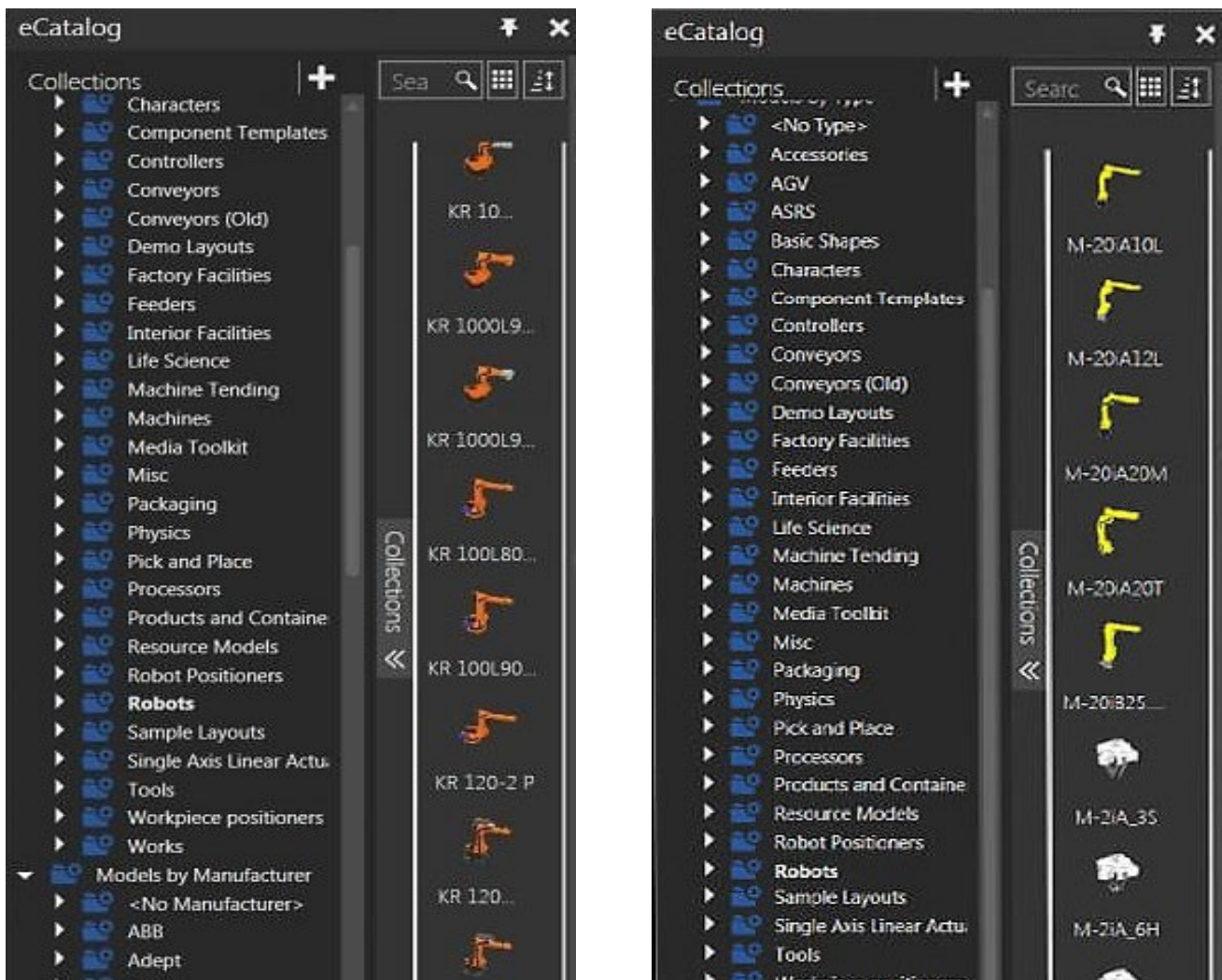


Figure 9. Variety of robots available, 200 vendors

Designing, implementation, and testing of Visual Components Software:

- a) *Generation of Models:* The first step to Visual Components software is to create a virtual model of the work cell. This involves creating or obtaining 3D CAD models of the equipment, workpieces, enclosure, tools, and other resources and fixtures that will be in the work cell; and importing them into our Visual Components software. There may be extra steps to simulate resources and processes, depending on the Visual Components software being used. Accuracy of the models and process-related information used is critical to generating a reliable simulation of the process and error-free offline program for the robots.
- b) *Tool Path Generation:* Tool path generation involves extracting robot positions from 3D CAD data with a specific tool center point – the point to which all robot positioning is defined. Many Visual Components software packages can do this automatically, and have built-in functions to automatically generate paths from features of the CAD models, such as corners, edges, or other geometry features.
- c) *Process Optimization:* Process optimization incorporates trajectory planning, process planning, and tooling design. It's an iterative design loop with several factors and trade-offs that need to be considered, so simulation helps significantly with this process.
 - **Trajectory planning** involves determining the best route for the robot to make. Robot work cells are typically designed in compact configurations, so motions and trajectories have to be planned carefully to avoid unintended interactions between the robot and other objects in the cell. Several factors need to be considered, such as motion type, joint configuration/speed/acceleration, reachability, and collision detection and avoidance.
 - **Process planning** involves planning the processes and workflow in the work cell. The major constraints for this step are budget, productivity, and quality, but there are several factors and trade-offs that need to be considered. This step includes layout design, resource selection (including robots and other equipment), maintenance considerations, and sequence optimization.
 - **Tooling design** involves the selection, modification, and placement of tools. This includes robotic end of arm tooling and other tools that touch the workpiece, such as positioner faceplates and clamps.
- d) *Post Processing:* After the robot program has been verified in the simulation environment, it needs to be implemented to the real robot. But first, the program must be converted into the language of the target robot. This step is called post-processing. Post-processors; the drivers that convert the robot programs; need to be created to perform this conversion. Post-processors are specific to the robot brand, application, and other customer-specific requirements for safety, usability, performance, etc. It's rare to develop post processors from scratch; it requires lots of work and many companies develop/sell them commercially; but they still require some customization for your specific application and setup. As a gross estimate, about 80% of the commands are the same for each post processor, but what changes is the local customization for each application, customer-specific requirements, and special macro commands that have to be called at the beginning or end of a program.
- e) *Initial Calibration:* Initial calibration involves calibrating errors between the work cell and virtual model, and updating the virtual model to match. The goal is to ensure the offline program is running at 100%, with no unplanned operator intervention. Calibration is performed using a tool centre point on the shop floor.

Depending on the application, it's also possible to calibrate without using a tool centre point; as an industrial robot can be used as its own coordinate measuring machine to determine the relative positions of critical components in the work cell.

It is not always necessary to perform initial calibration offline; it can also be done during the production environment. For example, in robotic spot welding, there are usually around 10- 20 points that need to be programmed in the robot. It would be much faster to have the operator to calibrate the robot versus doing it offline. Similarly, in arc welding, there is a lot of variation between workpieces, and robots use vision sensors to compensate for workpiece errors.

Virtual Labs in Visual Components:

Introducing a “Robot Safety Cell” virtual lab in Visual Components software in this section to give fundamental ideas about working robotic safety cells that includes topics such as selecting tool frame, jogging the robot, inserting safety components like generic fence, light curtain, grill door and safety area, and robots etc. Working on a “Robot Safety Cell” virtual lab using Visual Components simulation software will fulfil the objectives of programming, designing and operating robots and work cells without having a real physical robot and real working environment. The students will be able to program, design, implement and edit robots safety cell virtually after completing the above topics and understanding the real life scenario.

Virtual Lab sample – Pick & Place in Robot Safety cell

Aim:

Designing a simulation program on pick and place parts using while loops and sensor in a safety cell. A numerous small cylinders will be picked and placed on numerous blocks one-by-one, from one conveyor belt to another conveyor belt using sensors, done by a robot on while loop in a robot safety cell, so, the robot can easily detect the plate on one conveyor belt and hold it for a few second and in the mean while the robot will pick the cylinder and place it on the block. As robot has been programmed with while loop, the program will be continuous. Usage of safety cell will assure that there will be no intervention from human or any other thing can disturb the Industrial Robotic Process and no life will be placed under any danger.

Components Used:

Components used in this lab are Generic Fences, Light Curtain, Grill Door, Safety Area, Conveyors, Robot, Suction Gripper, Feeder, End Block.

Procedure:

1. Run “VisualComponents 4.1” software. In order to select components to the simulation screen, go to the **eCatalog** tab, click the **Models by Type** option.
2. Select any two **Conveyors** from **eCatalog** and place them side by side into the simulation.
3. Search and select two **End Block** from **eCatalog** and attach both to one ends of each conveyors.

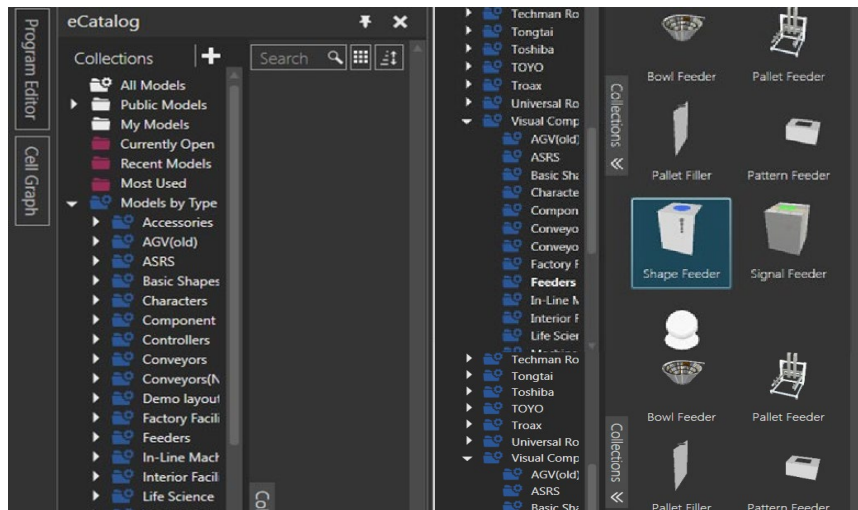


Figure 10. *eCatalog*

4. Search and select two **Feeder** from **eCatalog** and attach it to the both conveyors into the simulation, update the **Products** in both the **Feeders** such as **Block** and **Cylinder**.
5. Search and select any **Robot** from **eCatalog** and place it between the two conveyors.
6. Select a **Parametric Suction Cup Gripper** from **eCatalog** and attach it on the robot's head.
7. Select three **Conveyors** more from **eCatalog**, one conveyor is **Curved Conveyor** and other two conveyors are **Straight Conveyors** place them side by side into the simulation.
8. Start designing the **Safety Cell** and addition of the **Safety Signs** around your working area would be paramount in order to increase safety measures.
9. Search and select seven **Generic Fence** from **eCatalog** and attach it each other around the robot and conveyor belts.
10. Search and select one **Light Curtain** from **eCatalog** and attach it to the one **Generic Fence**.
11. Search and select one **Grill Door** from **eCatalog** and attach it to the one **Generic Fence**.
12. Select one **Safety Area** from **eCatalog** and place the robot and both the conveyors inside it.



Figure 11. *Generic Robot*

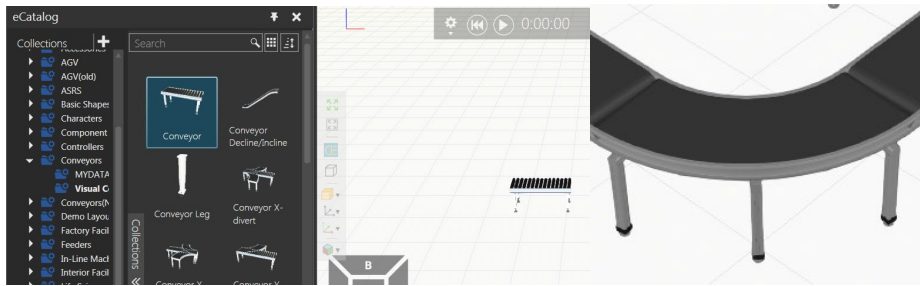


Figure 12. Conveyor

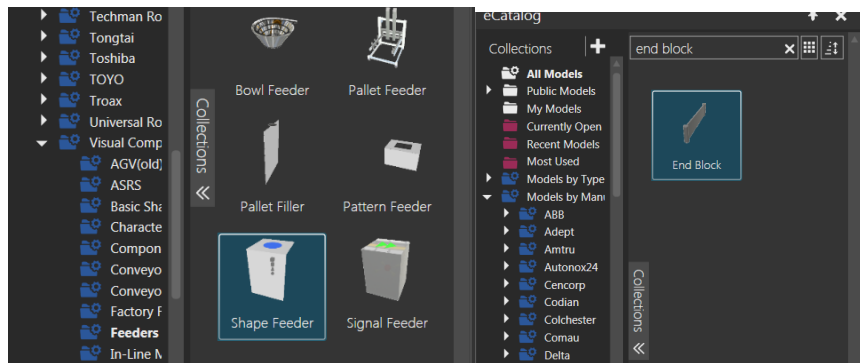


Figure 13. Shape Feeder Figure 14. End Block

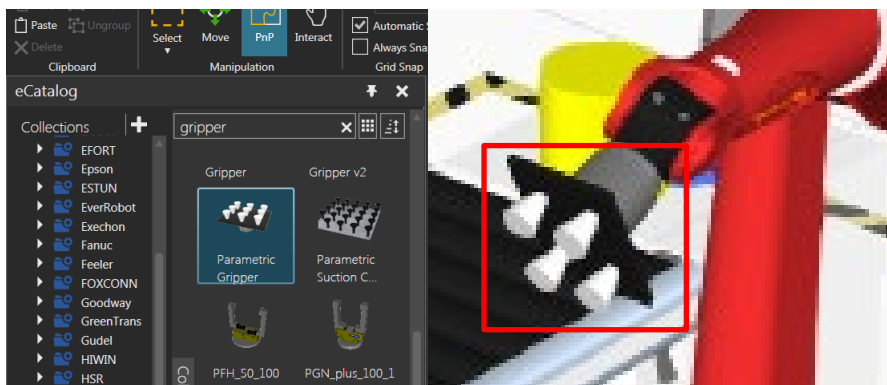


Figure 15. Parametric Gripper



Figure 16. Generic Fence

Figure 17. Grill Door

Figure 18. Light Curtain

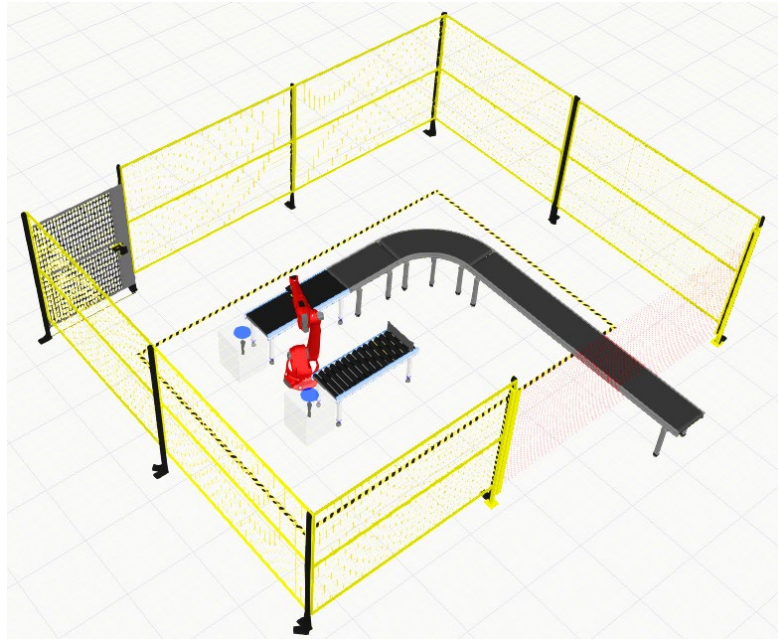


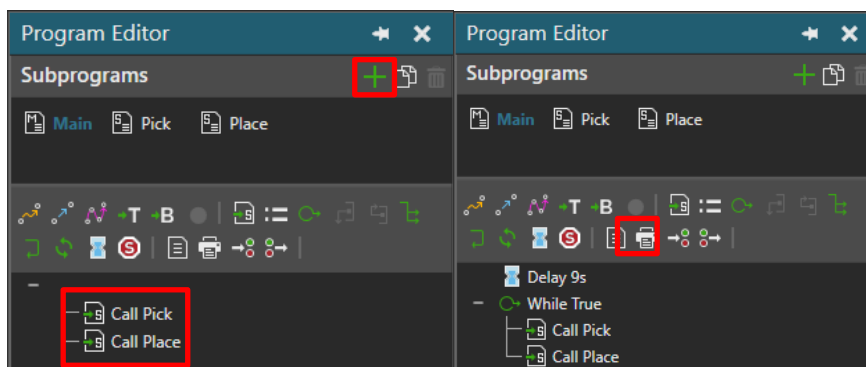
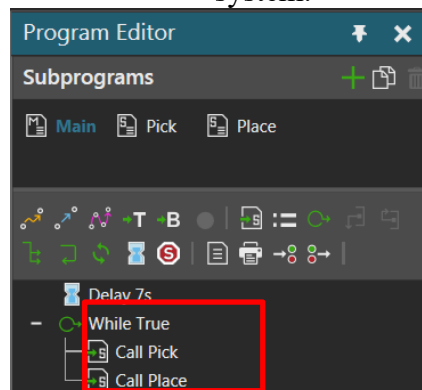




Figure 19. All the components placed at their position into the simulation screen

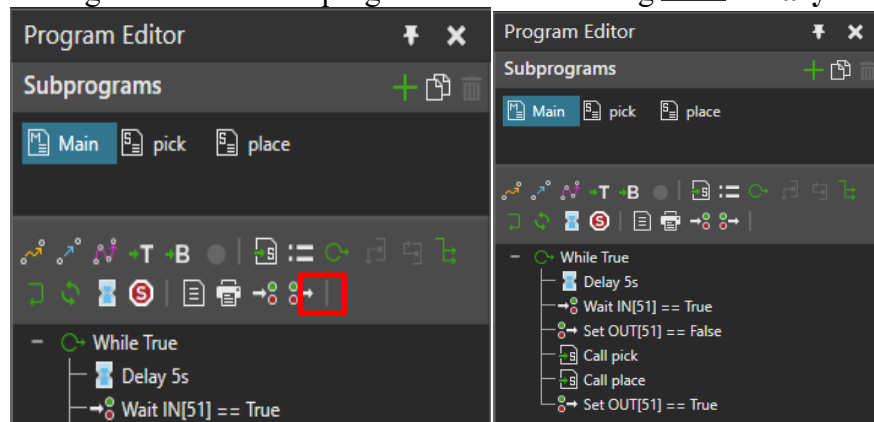
1. Create and name two subprograms by clicking the , that is 'Pick' and 'Place' under the **Main » Subprograms » Program Editor** Category.
2. Select **Main** and drag both the newly created subprograms into it. Add Delay of 5s by selecting this hour glass shape  in **Main » Subprograms » Program Editor** Category.



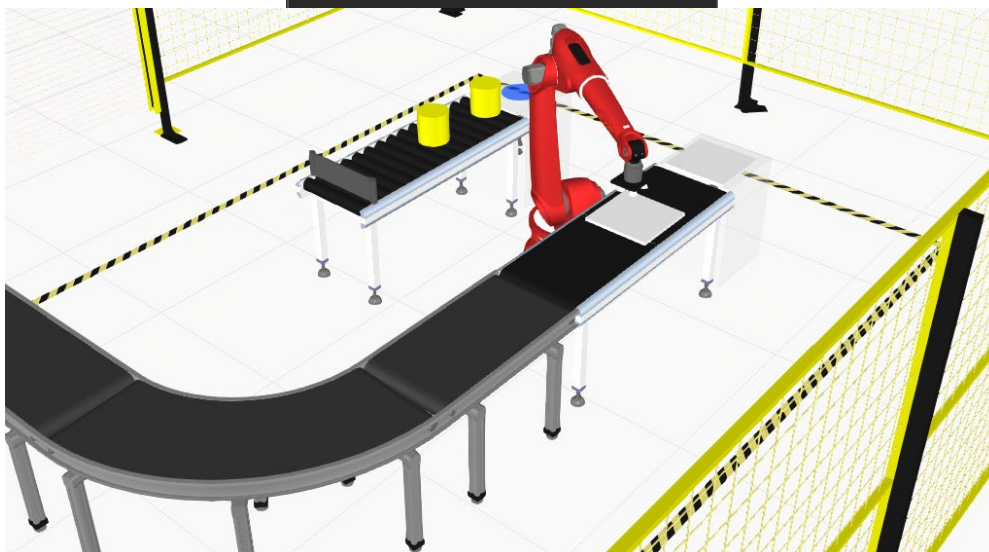
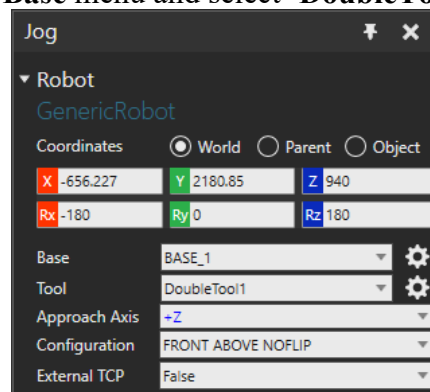
3. Put the both the subprograms that is 'Pick' and 'Place' under **While True** loop system.




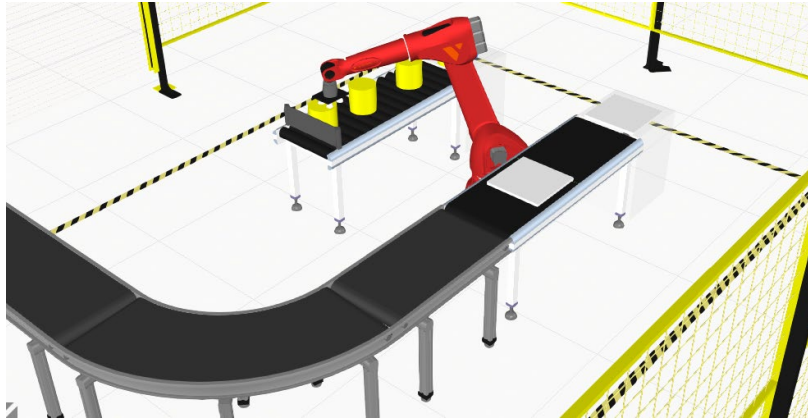
- Clicking the **Signal**  tool from the **Tool Menu**, the sensor of the **Conveyor** gets activated and programmed it with using  **Binary** tool.



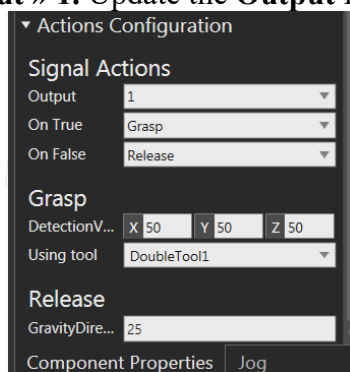
- Update the setting of the robot in the **Jog** or **Statement Properties** window. Select **'Base1'** in **Base** menu and select **'DoubleTool1'** in **Tool** menu.



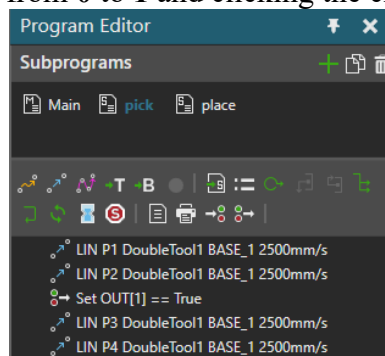
- The **Block** stopped because the **Sensor** sensed it.
- Start moving the robot and record every step to check the process.
- Select **Pick** » **Subprograms** » **Program Editor** Category and insert a point by clicking  the blue arrow.
- Jog the gripper to the top of the cylinder by dragging the purple circle in the center of the gripper to the center of the cylinder by using **Snap** tool.





10. Select the robot again and go to the **Component Properties** » **Action Configuration** » **Signal Actions** » **Output** » **1**. Update the **Output** from **0** to **1**.

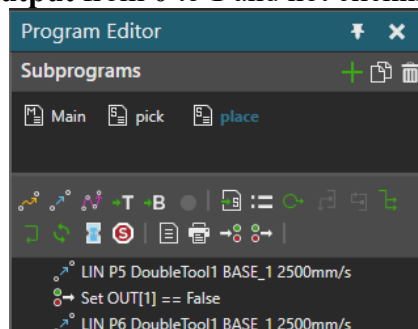


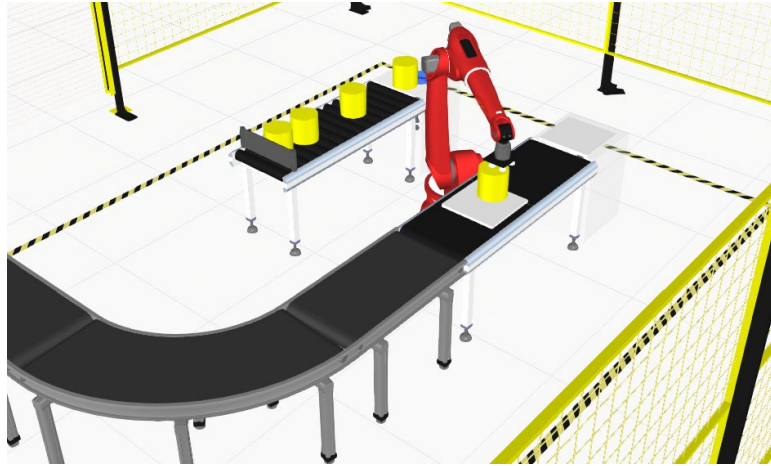
11. Move the robot and select this  **Binary** tool to pick the boxes. Make it **True** first by updating the **Output** from **0** to **1** and clicking the checking box.



12. Same processes for the **Place** subprogram as well instead of just one this  **Binary**

tool. In **Place** dropping the box would be done by making this  **Binary** tool **False** first by updating the **Output** from **0** to **1** and not clicking the checking box.





13. While Loop system starts, and the robot will do the same process continually.

Lab Conclusion:

The objective of this lab was to learn how to create a safety environment for Industrial Robotic Process. Therefore, there will be no intervention from human or any other thing can disturb the Industrial Robotic Process and no life will be placed under any danger. Adding safety components such as Light Curtains, Fences, Safety Area and Grill Doors around the robots is essential for overall safe environment, so that no human can pass through it without authorization. Safety Light Curtains are Opto-Electronic devices that are used to safeguard personnel in the vicinity of moving machinery with the potential to cause harm such as presses, winders and palletizers.

Discussion

It is discovered from this current COVID-19 situation that teaching classes online can be a good alternative and incorporating Visual Components to programming industrial robotics courses can help students to learn about robot programming and the robotics environment without having any physical robot. Overall feedback from the students is positive. Most of the students wanted to continue working with the Visual Components software for better understanding and want to work in the robot programming field. A few of the students feedbacks are mentioned below:

- a) Working with VC gave me great idea about real life robotics scenarios
- b) It is a good software for visualization and integration
- c) I want to pursue virtual robot programming field

Conclusion

This paper provides a quick and general overview of the steps followed in the Visual Components software process. As has been analyzed, Visual Components software is not just developing a robot program; it's a process for designing and planning an entire robot work cell. Visual Components software would help students to develop a practical approach towards automation industries and to design better production solutions by allowing them to evaluate and visualize the many requirements, constraints, and tradeoffs in the Visual Components process. This simulation software will act as a catalyst for manufactures as it can reduce errors, wastage and total manufacturing time.

References

- [1] D. Mihai, "Visual Components: An Advanced Production Line Simulation Software - Smashing Robotics", Smashing Robotics, 2020. [Online]. Available: <https://www.smashingrobotics.com/visual-components-production-line-simulation-software/>. [Accessed: 04- Feb- 2020].
- [2] [Dancun Kingori](#), The World of Robotics Revolution, Available: <https://www.newgenapps.com/blog/the-world-of-robotics-revolution>, Dec 21, 2018
- [3] Dr. Maged Mikhail, Virtual Robot Labs for Programming Industrial Robot Course, Purdue University Northwest, @American Society for Engineering Education, 2019
- [4] Robot programming, Using simulation to ensure accurate and smooth robot programming, Available: <https://www.visualcomponents.com/solutions/robot-programming/>. [Accessed: 04- Feb- 2020].
- [5] "Teaching Robots with Visual Components", Visual Components, 2020. [Online]. Available: <https://www.visualcomponents.com/insights/articles/teaching-robots-visual-components/>. [Accessed: 04- Feb- 2020].
- [6] "3D Simulation Software — ABCO Automation", ABCO Automation, 2020. [Online]. Available: <https://www.goabco.com/3d-simulation-software>. [Accessed: 04- Feb- 2020].
- [7] "Visual Components 4.2 - The next generation of 3D manufacturing simulation", Visual Components, 2020. [Online]. Available: <https://www.visualcomponents.com/products/visual-components/>. [Accessed: 04- Feb- 2020].
- [8] Visual Components Essentials Provides Powerful Production Simulation, Available: <https://www.roboticsbusinessreview.com/ai/visual-components-provides-powerful-june-21-2016>
- [9] [Academy.visualcomponents.com](https://academy.visualcomponents.com), 2020. [Online]. Available: <https://academy.visualcomponents.com/wp-content/uploads/2018/05/Module-1-Basics-of-Simulation.pdf>. [Accessed: 14- Apr- 2020].
- [10] "How to create project ready content with Visual Components", Visual Components, 2020. [Online]. Available: <https://www.visualcomponents.com/insights/blog/how-to-create-project-ready-content-with-visual-components/>. [Accessed: 14- Apr- 2020].