Engineering Graphics Concepts Used in Graphics-Based Off-line Robot Programming

Kevin L. Devine
Department of Technology
Illinois State University

Theodore J. Branoff
Department of Technology
Illinois State University

Abstract

Robotics industry sales statistics clearly document the increased use of industrial robots in manufacturing applications, and increased interest is being placed on the use of graphics-based industrial robot programming and simulation applications. This digest explores engineering graphics-based tools and concepts that are commonly used to develop industrial robot cells. Recommendations for engineering graphics course content are presented.

Introduction

Since the first industrial robot was installed in 1961, the number of industrial robots in use worldwide has grown to more than 1 ½ million (International Federation of Robotics, 2016). With annual sales of industrial robots increasing from 60,000 units in 2009 to 248,000 units in 2015, most experts agree that industrial robots will play an important role in the future of manufacturing (International Federation of Robotics, 2016). Along with the surge of robot installations, increased interest is being placed on the use of graphics-based industrial robot programming and simulation software. This digest will explore engineering graphics-based tools that are commonly used to develop (design and program) new industrial robot cells, and identify applicable concepts that can be taught in engineering graphics courses.

Background

In general terms, the International Organization for Standardization (ISO) in their ISO 8373 Robotics and Robotic Devices Standard defines an industrial robot as a reprogrammable, multipurpose manipulator used in industrial automation applications (International Organization for Standardization, n.d.). Industrial robots are general-purpose machines that must be configured to perform an industrial task. In concept, the general purpose industrial robot is much like a cordless drill in the sense that both the drill and the robot must be configured to perform a task. In the case of the cordless drill, the user must insert the proper drill bit and manually control the drill manually to create a hole at the desired location. In the case of an industrial robot, end of arm tooling must be installed and the robot must be programmed to perform a particular industrial task. Figure 1 shows an “off the shelf” industrial robot without customized tooling.

Figure 1. Industrial Robot As Delivered by Manufacturer

Industrial robots are almost always integrated as part of a group of machines and devices to form an industrial robot cell. ISO 8373 defines an industrial robot cell as one or more industrial robot systems, including associated machinery and equipment and the associated safeguarded space and protective measures (International Organization for Standardization, n.d.). Figure 2 shows an industrial robot cell that has been configured to drill holes. In this example a standard industrial robot is mounted on a pedestal, has end of arm tooling and related peripheral devices, has been safeguarded and has been programmed for the task of drilling holes in a part mounted in a fixture.

Figure 2. Industrial Robot Cell.

Most industrial robots are equipped with a hand-held teach pendant that is used by the robot operator to interact with the device. Many robot programs are developed interactively using the teach pendant. In this on-line programming method, the programmer manually positions the robot to desired locations which are recorded in a series of program steps. Although still widely used, the on-line teach method has many shortcomings, and off-line graphics-based robot programming and simulation tools are gaining in popularity (Devine, 2009). Figure 3 shows an off-line programming system being used to program the industrial robot cell previously illustrated.

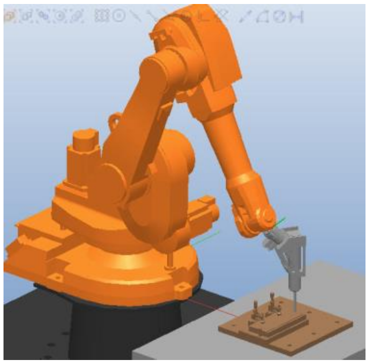

Figure 3. Graphics-Based Offline Programming Software Being Used to Program an Industrial Robot Cell.

Graphics-Based Programming Tools

There are many off-line graphics-based programming systems in use today. These systems are generally developed either by robot manufacturers such as Fanuc (Roboguide), ABB (RobotStudio), and Kuka (Kuka.sim), or by CAD/CAM vendors such as Siemens PLM (Tecnomatix) and Dassault Systems (Delmia). The systems developed by the robot manufacturers are tailored to create programs for their brand of robots. RobotStudio, for example, can only create programs for ABB robots. Systems such as Tecnomatix and Delmia, on the other hand, are conceptually similar to CAM systems in that they develop generic robot programs that are then postprocessed to create programs for a particular robot. Regardless of the system used, there are some common engineering graphics tools and concepts that robot programmers would benefit from understanding.

Solids Import and Geometry Transformations

Most modern off-line robot programming applications have a library of robots that can be imported to create the nucleus of a virtual robot cell. The robot models have solids-based geometry and robot kinematic properties already modeled by the software vendor. In addition to the robot geometry, user-defined solid geometry is usually imported into the virtual cell and positioned relative to the virtual robot. Many robot programming systems have only bare-bones solid modeling tools, making it common practice for complex geometry to be modeled in a separate CAD application. Figure 4 shows a complex end of arm tool that would not be practical to model inside a robot programming application.

Figure 4. Complex End of Arm Tooling Used to Place a Windshield on a Car

In some cases the solid geometry is imported in the native CAD file format, but often a neutral file format such as STEP is used. Models are initially inserted into the virtual robot cell based on the model origin of the imported geometry. Geometry that must move with the robot, such as end of arm tools, are repositioned and "attached" to the robot (see figure 5), while other geometry representing stationary objects such as workpieces and fixtures are repositioned using "geometry snaps" and various geometry translation and rotation functions (see figure 6).

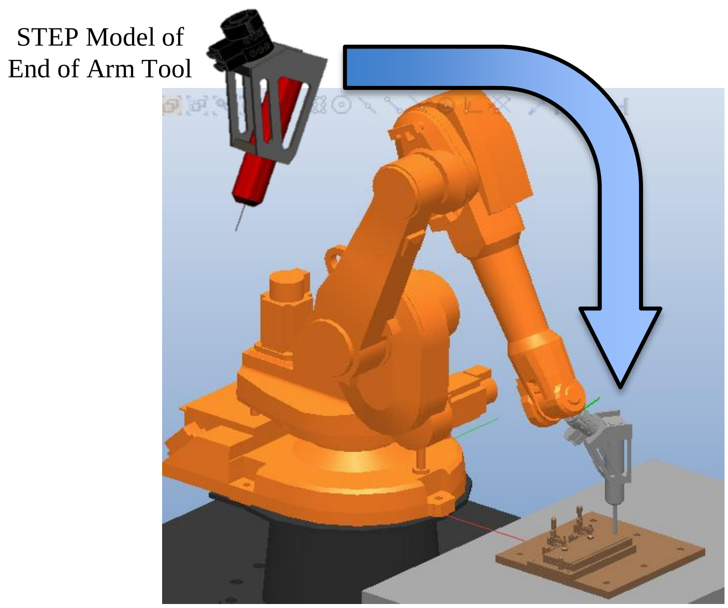

Figure 5. STEP Geometry is Imported and Attached To Virtual Robot.

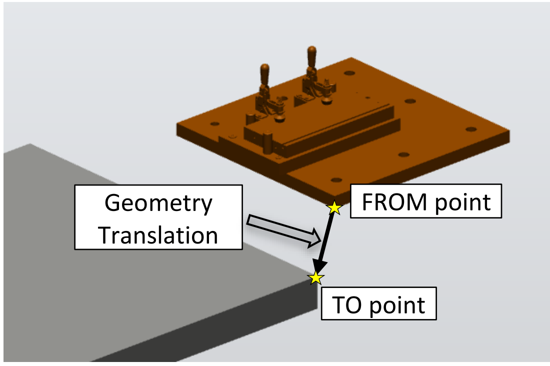

Figure 6. STEP model of Fixture Geometry is Imported and Positioned Using Object Snaps and FROM and TO Points.

Right-Hand Cartesian Coordinate System

Robots make extensive use of the right-hand Cartesian coordinate system as represented by the RGB (XYZ) triad symbol common to many CAD systems. Figure 7 illustrates three different examples of right hand coordinate systems in use in RobotStudio. The Tool Center Point (TCP) provides the means to position and orient the end of arm tool during program movements. The target frames provide locations and orientations for the TCP to be positioned, and the Work Object is a floating coordinate system that allows the program created in virtual space to be calibrated to the physical robot cell after the program has been deployed to the real robot. Figure 8 shows the TCP oriented at one of the target frames.

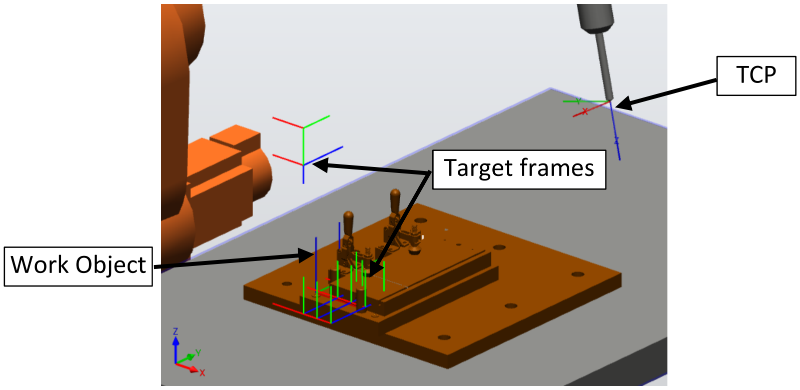

Figure 7. Right Hand Coordinate Systems

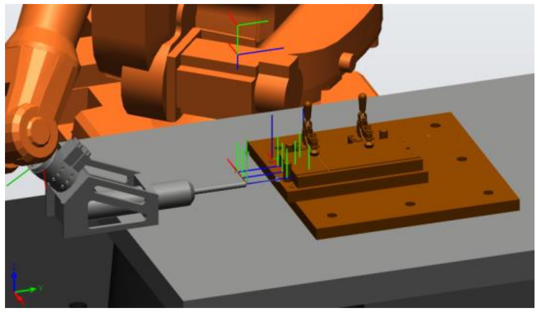

Figure 8. Robot Tool is Located and Oriented by Positioning the TCP at a Target Frame

Surface Normals

Solid geometry imported in to modern off-line robot programming applications often use the surface normals inherent in solid geometry to orient the robot's tool relative to the surface of the work. This is critical for tasks such as painting, welding and sealant application where precise control of the tool orientation relative to the part surface is important. A basic understanding of the principle of surface normals and IJK vectors is therefore helpful to the robot programmer. Figure 9 illustrates a robotic welding application where targets were created relative to the surface of a workpiece. The welding torch is positioned to these targets to control the orientation of the tool during program development.

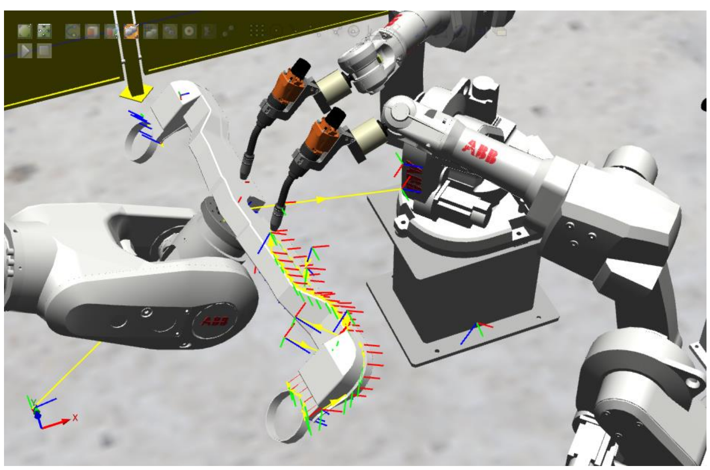

Figure 9. A Virtual Robot Welding Cell Surface Normals are Very Important in Welding Applications

Conclusions

Most experts agree that industrial robots will play an important role in the future of manufacturing, and increased interest is being placed on the use of graphics-based industrial robot programming and simulation applications. This digest explored engineering graphics-based tools and concepts that are commonly used to develop industrial robot cells. Engineering graphics educators can help prepare students for careers in robot programming by including content coverage in the areas of solid model formats, geometry transformation methods, geometry snap principles, the right-hand Cartesian coordinate system, and the principles of surface normals.

References

Devine, K. (2009). *Integrating Robot Simulation and Off-Line Programming Into an Industrial Robotics Course*. Austin, TX: Proceedings from the 116th American Society for Engineering Education (ASEE) Conference and Exposition.

Devine, K. (2012). *Using NX in Project-Based Design & Build Courses*. Siemens PLM Connection Annual Users Conference, PLM World, Las Vegas, NV.

Devine, K., Reifschneider, L. (2009). *Agile Robotic Work Cells for Teaching Manufacturing Engineering*. Valparaiso, IN: Proceedings from the 2009 American Society for Engineering Education Illinois/Indiana Section Conference.

International Federation of Robotics (2012). *History of Industrial Robots*. Retrieved from http://www.ifr.org/downloads/.

International Federation of Robotics (2016, June). *International Federation of Robotics CEO Round Table*. Symposium conducted at Automatica, Munich, Germany. Retrieved from http://www.ifr.org/news/ifr-press-release/world-record-816/.

International Organization for Standardization (n.d.) *ISO 8373 Robotics and robotic devices*. Retrieved from https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en.