Applying Computer Science in Industrial Robotics Implementations

Demetrius E Johnson
CIS – 381 – Industrial Robotics
Dates of Experimentation: February 3 – 10, 2021
Date of Submission: February 11, 2021

II. Abstract

This is a compact review study of applying computer science to the American Fanuc Six-Axes Robot to demonstrate the practicality and implementation of modern industrial robotics. The materials used in the experiment include the six-axes American Fanuc robot arm and its respective robot controller; a basic, enclosed robot work cell, a small box fixture, a sheet of paper with shapes, and a robot vision system. The objective of the study is to allow a potential user to get a basic grasp on using the user interface in order to interact with the robot controller and control the robot. To achieve this, the experimentation involves learning how the robot uses frames of reference and understanding how the robot controller allows computer science and electrical and mechanical engineering to overlap to grant programmability of the industrial robotic machine to a user. The potential user will also learn more in depth of the various ways to program the robot arm and the associated challenges of mechanical speed and accuracy capabilities and how to account for these physical limitations using the robot controller software, and the robot vision system.

Table of Contents

IV. Introduction

This study consists of a cohesive report on three separate lab experiments conducted:

- Lab 1 – demonstrate how to jog the robot arm and creating a basic robot program by setting up a robot frame.
- Lab 2 – demonstrate how to create a position register and creating three basic programs to trace three shapes.
- Lab 3 – demonstrate how to use the robot vision system and some of its practical uses, including finding a defective part.

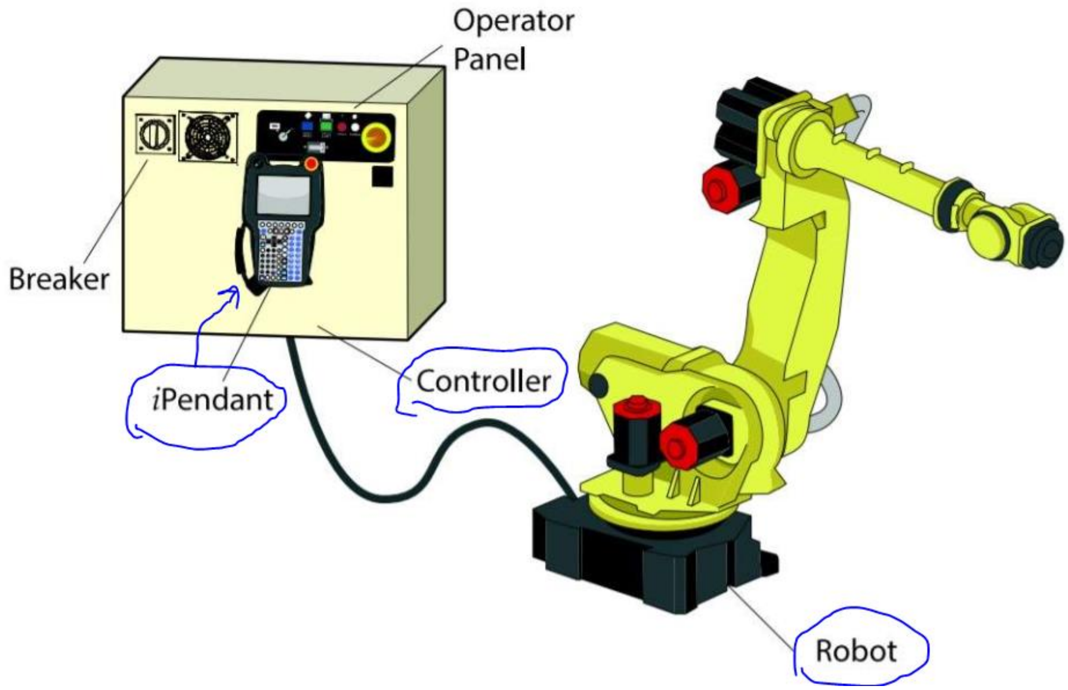
To provide some practical context, the aim of this study is to get the reader familiar with just one of many industrial robotics systems so that they can understand the primary components of any robot. For instance, if you enjoy and are knowledgeable on computer science principles or on mechanical or electrical engineering principles, this study provides an idea of how these subjects interact. Even if you do not have a strong background in these areas, it will still provide a general sense of understanding of robotic principles and basic components.

The reader will walk away with at least an idea of the level of sophistication and the wide application of robots, as well as the expandability and importance of the industrial robotics field. This study will also demonstrate (not by words from referencing various studies, but by showing even just a basic application of an industrial robot) why modern, advance robots have been able to successfully invade the production and development industry and continues to do so as the range of complexity of industrial robotics continue to develop – this will become quite obvious as this study progresses.

Before heading into the procedures and methods of this lab, let's talk about the basic components of an industrial robot; specifically, the industrial robot used for all lab experiments in this study is a 6-axes robot arm, built by the Fanuc Robot Corporation of America. A typically consist of three primary components:

- The computer: for robots this is often referred to as the "Controller".
- The user interface: this is basically like the keyboard or touch screen which allows a user to control ("interface with") the controller (computer).
- The robot: this is the actual mechanical device that performs the work sent to it by the controller.

So, you see, it is simple, and intuitive. For any electronic device to work, it needs something to compute and execute tasks, and there needs to be a way to tell that device to do a task, and finally in the case of robotics you translate the programmed work already done by the computer into some other form of work such as moving a robot arm, or turning a tire, or in the most basic sense changing the color being emitted by a light signal. In this experiment, you will see that the equipment used for the 6-axes robot constitutes of the three mentioned basic components: a controller, a user interface, and a robot. When a controller, which is essentially a computing device, a user interface, and some output mechanical device (robot) all work together, the whole system is referred to as a “robot system”. Below is the example robot used in this experiment:



For this experiment, the iPendant (or “teach pendant”) is the name used by the Fanuc Robot developers. It is merely their naming convention for the user interface they developed to send commands to the robot controller.

Next, the 6-axes robot arm gets its name from the number of axes it has. But what is meant by axis? A robot axes simply refers to any part of a robot (a robot arm for this experiment) that propagates a direction of movement; this can be a linear or rotational movement. Each location that can move the robot in some direction is known as a robot joint. So, a 6-axes robot arm is a robot arm that has six different locations on the robot that can move in some direction (joint location). Note that every joint can only move a robot along or about a single axis; if the joint can turn the robot on some axis and at that same location the robot moves outward or inward linearly, then at that location the robot is said to have two separate joints.

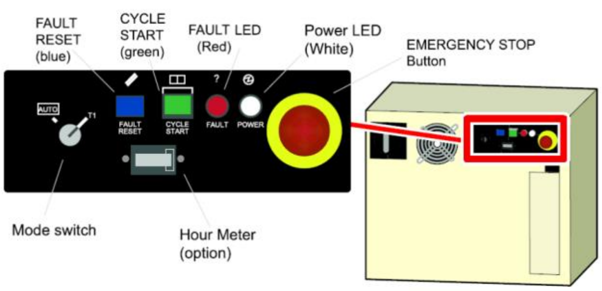
V. Methodology

Lab 1

Summary of Procedures

Here is the list of procedures for the first experiment:

1. The robot is equipped with emergency stop buttons; the first button is on the user interface mechanism, in this case it is called the “iPendant”, and the second is on the robot controller. Both emergency stop buttons were turned counter-clockwise to release them. They both had to be released, or else the robot would not power on. After this step, we turn the iPendat on using the on/off button located on the iPendant. See the image below to get an idea of the locations of the emergency stop buttons:

   Emergency stop button on the robot controller.

   Emergency stop button on iPendant (user interface). It is the red button.

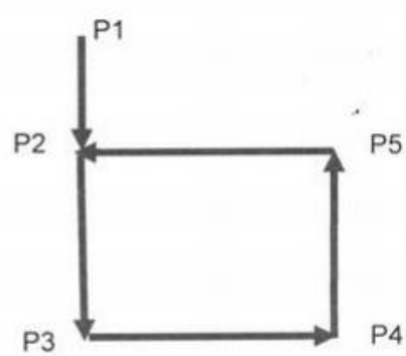
2. Next, the robot was jogged. The term “jog” is a term used in robotics which simply means to interact with the user interface and tell the controller to move the mechanical robot arm to a specified location. This can be a very dangerous process, and so for the experiment the robot is enclosed inside glass box; this box is what is known as a robot’s “work cell”. A work cell for a robot does not have to be in an enclosed case; it simply refers to the physical location where the robot is considered to be able to move and operate. There are four primary frames by which the robot can be jogged: jog frame, user

frame, and world frame. The world frame is the primary frame of reference that the robot sensers base all other frames off of, thus, the location of this frame remains constant at all times. The tool frame is the frame that considers the location of the origin of the end of the robot arm – this is often referred to as the Tool Center Point frame, since depending on the tool attached to the end of the arm, the center of the tool might change. The jog frame refers to the joint angles and positions; the robot sensors use the positional data of each joint to determine a point in space with respect to the world frame. Finally, the user frame is a frame that can be defined by the user; the only reason this frame works is because the robot always knows the location of the world frame origin and the tool center point origin.

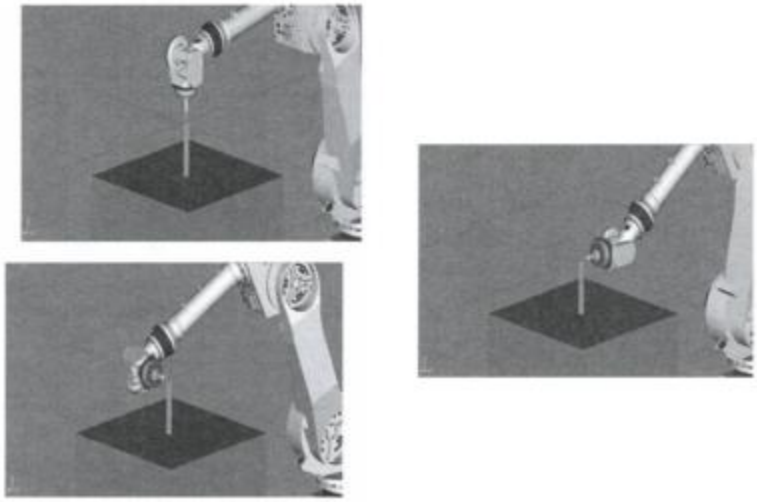
3. So, the robot was then jogged to a home location – a safe position from which we could start to write some instructions for the basic program for the experiment.

4. For this step, the robot is “taught” positions to move to by writing the following program instructions to execute a path that returns to its starting point:

```
BOX:
1: J P[1] 100% FINE
2: J P[2] 100% FINE
3: J P[3] 100% FINE
4: J P[4] 100% FINE
5: J P[5] 100% FINE
6: J P[2] 100% FINE
7: J P[1] 100% FINE
END
```



5. Next, after successfully executing the path shown above, the process of defining a Tool frame is completed. In order to define a Tool frame, the three-point approach methods was used. This method consists of approaching the same point in space from three separate joint angles so that the robot can sense and get an idea for what the origin of the tool frame will be. The point in space is approached by using the center point of the tool. See the image below for demonstration:



Above, the Tool Center Point (TCP) approaches the same point from three different joint angles (the orientation and joint positions of the robot are different).

6. Next, a user frame is defined, using the same three point method. The image below demonstrates the method screen shown on the teach pendant:

```
SETUP Frames
  User/RTCP Setup/ Three Point
  Frame Number: 2

    X: 0.0          Y:    0.0       Z:    0.0
    W: 0.0          P:    0.0       R:    0.0

  Comment: *********** ******
  Orient Origin Point:          UNINIT
  X Direction Point:            UNINIT
  Y Direction Point:            UNINIT

  Active UFRAME/RTCP $MNUFRAMNUM[1]=1
```

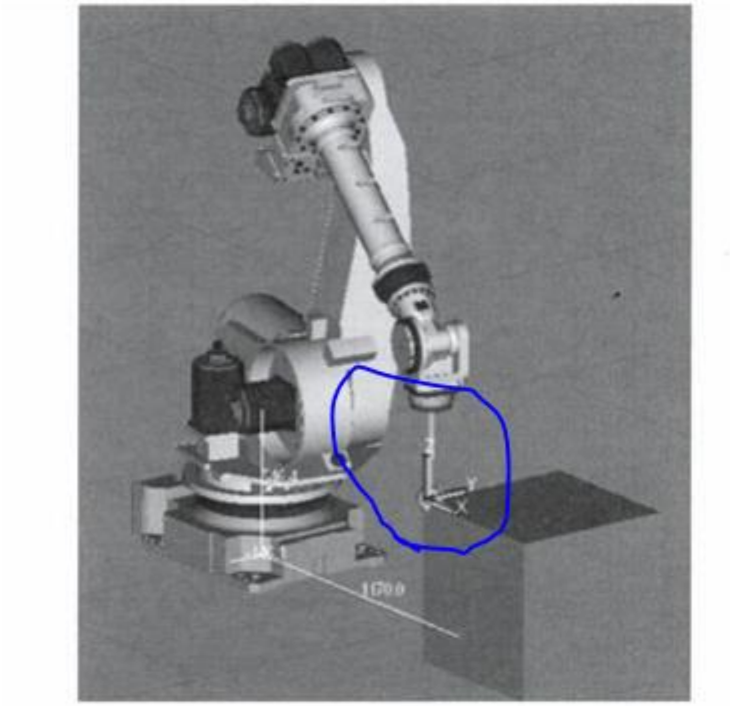
Program Discussion

In summary, the experiment was successful: a program was successfully written with the default start and stop methods for each recorded point, a tool frame was defined using the three point method, and a user frame was defined using the three point method to define the origin of the frame.

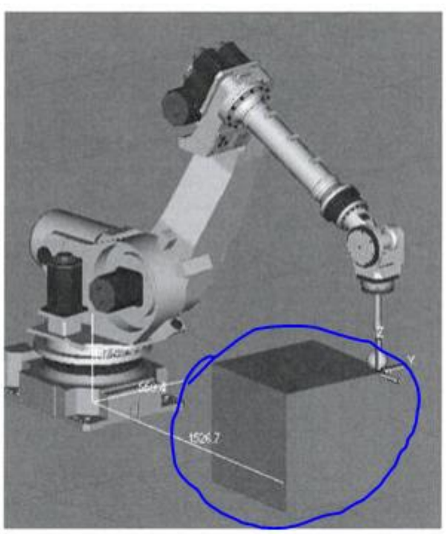
User frame and Tool Frame Pictures

Below are some photographs of the tool and user frames:

Tool Frame:



User Frame:

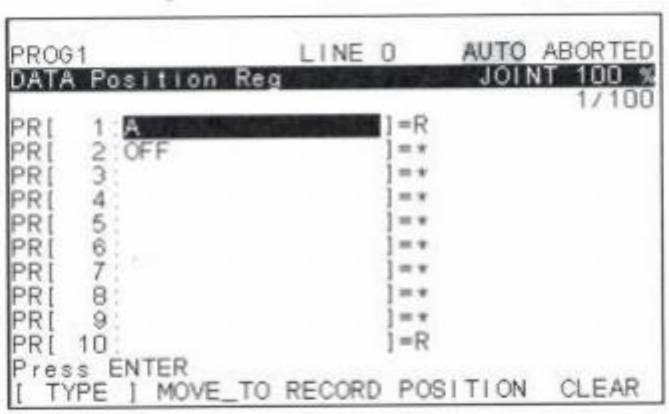


Lab 2

Summary of Procedures

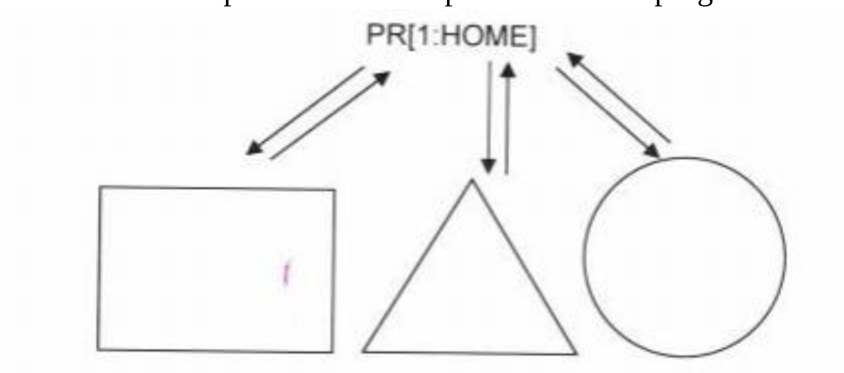
For this experiment, a Position Register was created and a Shape Program was created. A position register is a globally accessible position location that can be accessed by any program when recording a point. For example, when a position register is defined, two different programs can select the position register as a point for the robot to move to. Note, the default position register type is with respect to the user frame that is active when it is recorded; in order to have an absolute position register point that will always move the robot to the same location no matter what user frame is active, a joint position register must be recorded so that the position is based on joint orientation and position and not the tool center point with respect to user frame.

1. First, a position register was created in the next available PR slot on the iPendant.



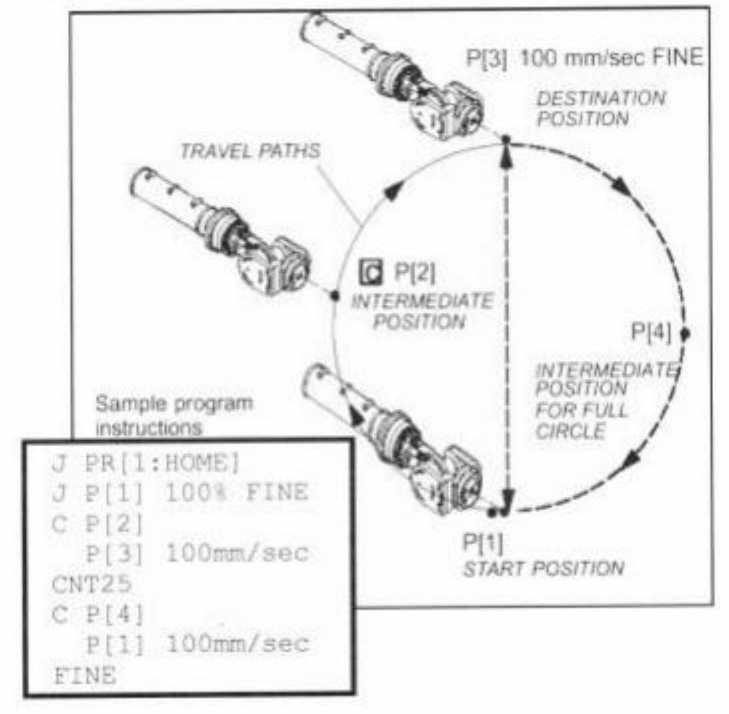
```
PROG1                    LINE 0      AUTO ABORTED
DATA Position Reg                     JOINT 100 %
                                           1/100
PR[   1:A                  ]=R
PR[   2:OFF                ]=*
PR[   3:                   ]=*
PR[   4:                   ]=*
PR[   5:                   ]=*
PR[   6:                   ]=*
PR[   7:                   ]=*
PR[   8:                   ]=*
PR[   9:                   ]=*
PR[  10:                   ]=R
Press ENTER
[ TYPE ] MOVE_TO RECORD POSITION  CLEAR
```

2. Next, a shape program was written to demonstrate the different methods necessary when writing a program needed for specific tasks. This part also demonstrates how a robot has many complex capabilities through various methods in order to complete tasks. Below is a picture of the shapes used for the program:



3. Before writing the program, a user frame was defined using the box on which the picture of shapes laid upon.
4. Then, each point on the triangle and rectangle are traced; the vertices of each shape were recorded as points.

5. As for tracing the circle, a special method had to be used. A three point arc method is used to define one half of the circle; the robot controller is then able to calculate the other half (corresponding arc) of the full circle. See the image below:

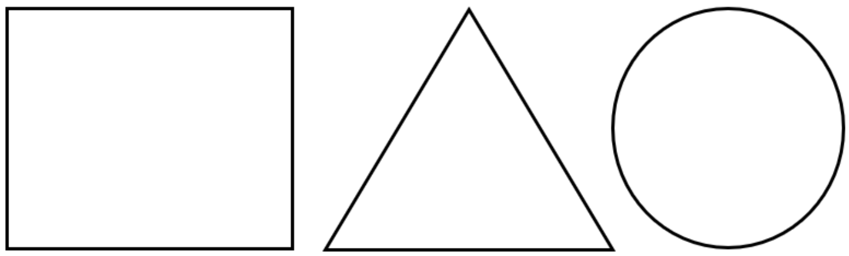


6. After all points were recorded, the program was ran successfully.
7. In order to test the usefulness of user frames, the work fixture on which the shapes laid upon is relocated in the work cell. To mitigate for this factor, the user frame used by the shape program is redefined with its origin at the correct location. Then, the program is executed again at the new location, and the shapes were successfully traced.

Program Discussion

The program ran successfully with both the original user frame, and the redefined user frame in order to match the new location of the shapes in the work cell. It is important to note that the Position Register created in this lab experiment was used as a home position; every time the program was executed, it began at this safe home position.

Pictures of the shapes drawn from the program:
Below is a picture of the official shapes used in the experiment:



Lab 3

Summary of Procedures

For the final experiment, which was conducted in Lab 3, we learned how to use the robot vision system. A robot vision system is essentially an extension of the robot function through the use of a camera, which is connected to the robot controller. The camera is able to use points of reference inside of the robot work cell in order to detect and move objects within the referenced workspace. Below, is a picture of the robot vision system camera used for this lab:

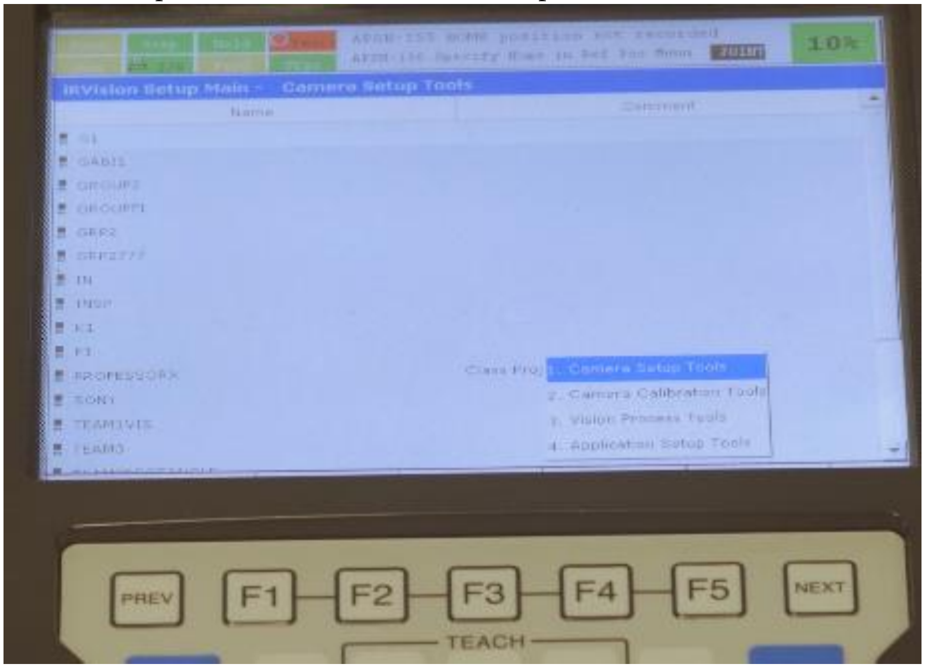
Above, the camera used in the robot vision system is circled in blue. It is a Sony XC-56 Camera.

Now, let us go through the procedural steps:

1. At the start of lab three, it was established that we use a grid made of circles for our camera's frame of reference. Each circle's center is spaced 30 millimeters apart. This information is crucial in order to make use of the robot vision system, which we had to input into the robot controller through the user interface (teach pendant):

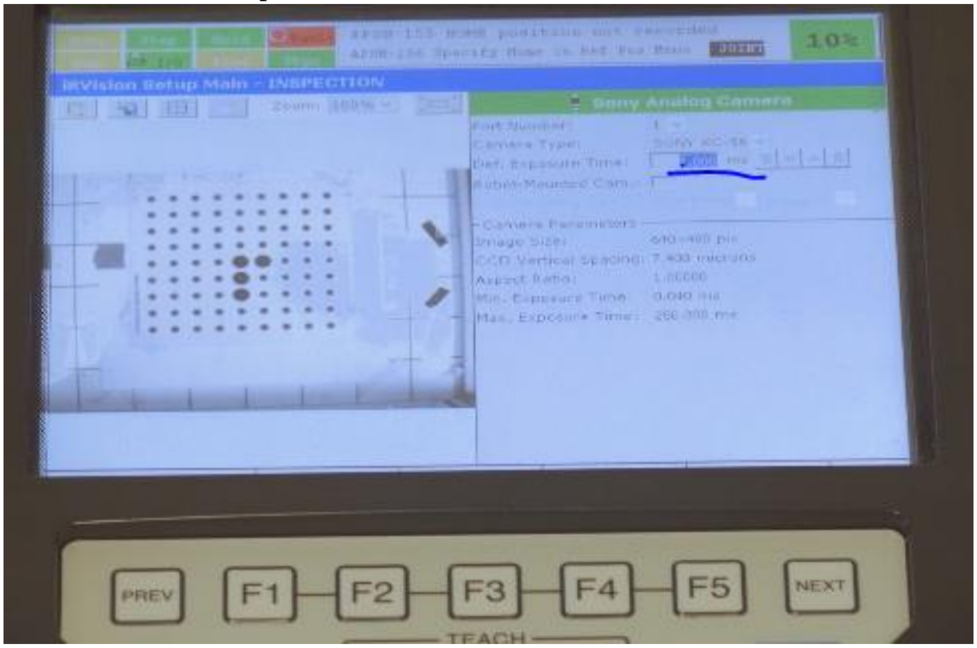
Above, the 30-millimeter spaced circle grid used for the experiment.

2. Next, on the teach pendant, the Camera Setup Tools option was selected, and a new camera setup was created and named "Inspection":



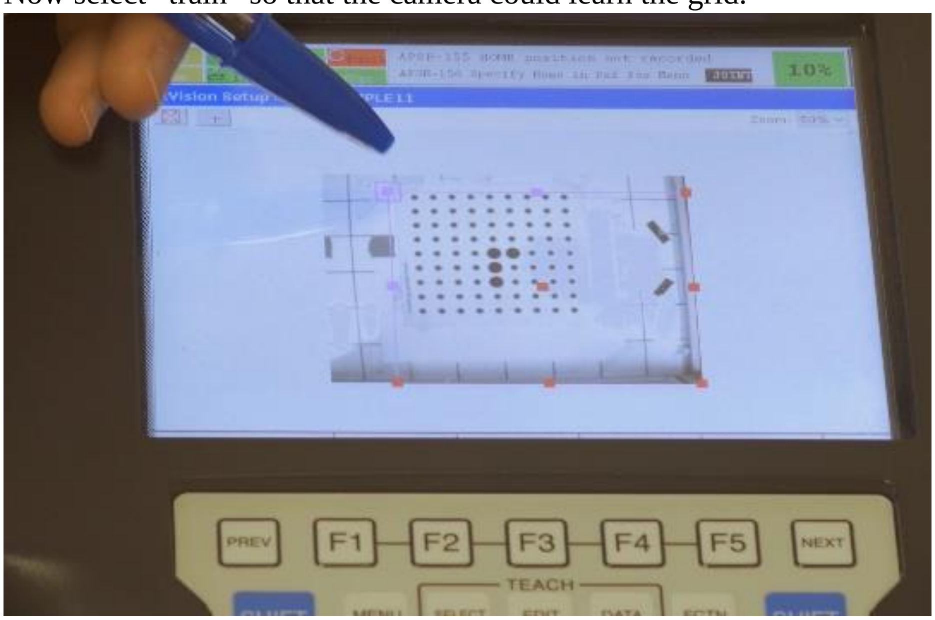
Above, the Camera Setup Tools option.

3. Then, the default exposure time is set to 1 ms:



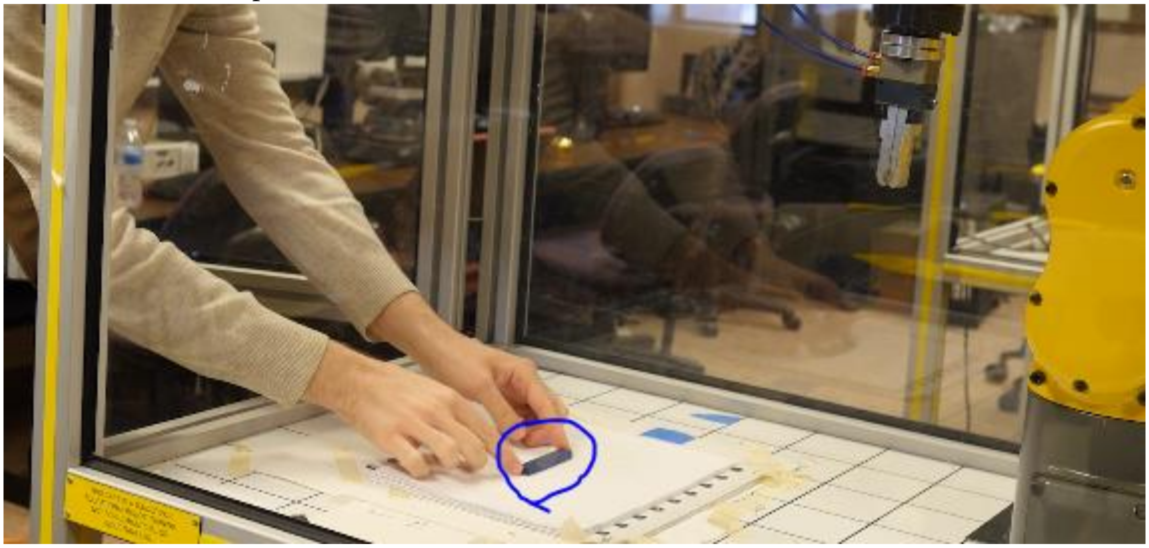
Above, setting exposure time.

4. Then, the vision process tool was selected.
5. Create new vision data named “sample”: single view inspection option.
6. Select the XC-56 Sony camera that was previously set up for the vision process.
7. Grid spacing set to 30mm.
8. Focal distances set to 6mm.
9. Now select “train” so that the camera could learn the grid.



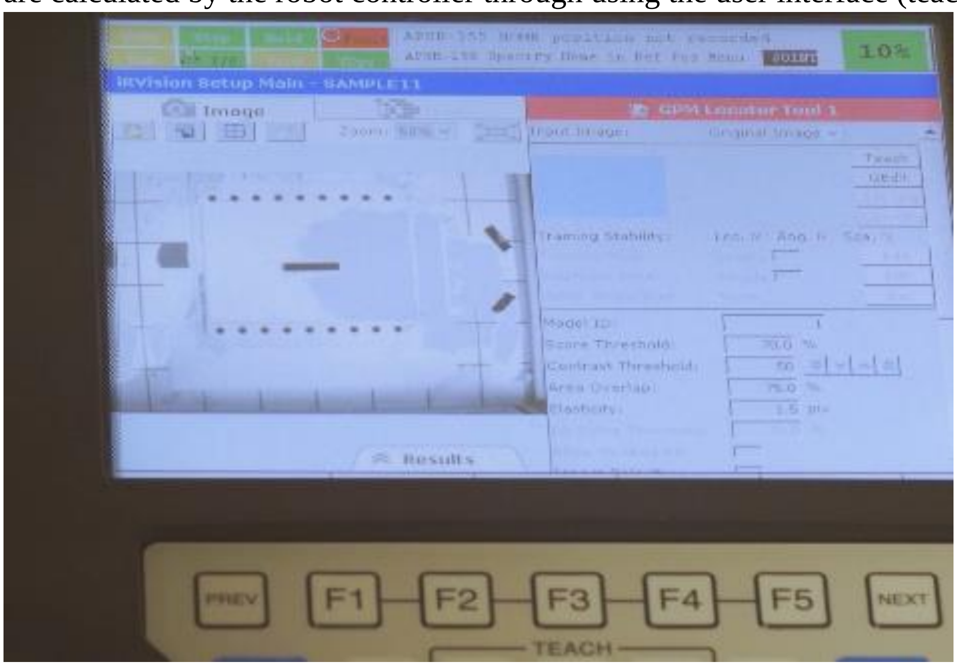
Above, the process of teaching the camera vision process the grid limits.

10. After setting the camera to learn the grids, the process was saved as a “Windows Shift Tool”.
11. Next, the tool that was created was selected as a tool to be used with the robot arm, and a test block was set up in the robot work cell:

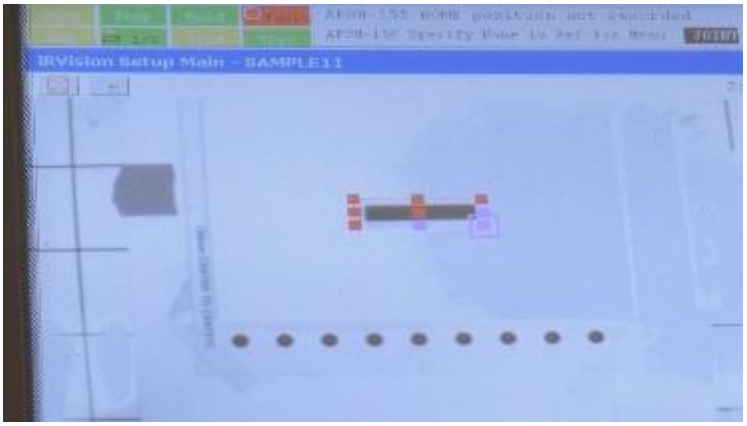


Above, the block set up for testing in the robot work cell.

12. Now the dimensions of the part placed in the vision system view in the robot work cell are calculated by the robot controller through using the user interface (teach pendant):

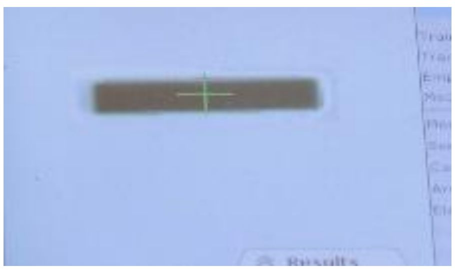


Zoomed-out view of the part and grid.



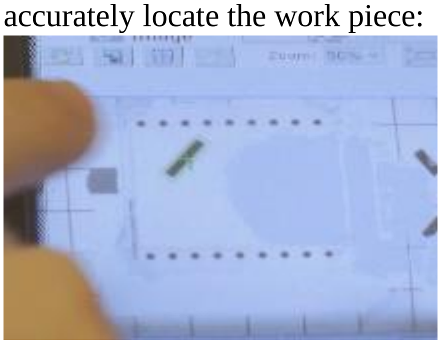
Zoomed-in view, teaching the camera to detect the dimensions of the part.

13. Then the robot did not accurately detect the dimensions, so the previous steps had to be retaken.



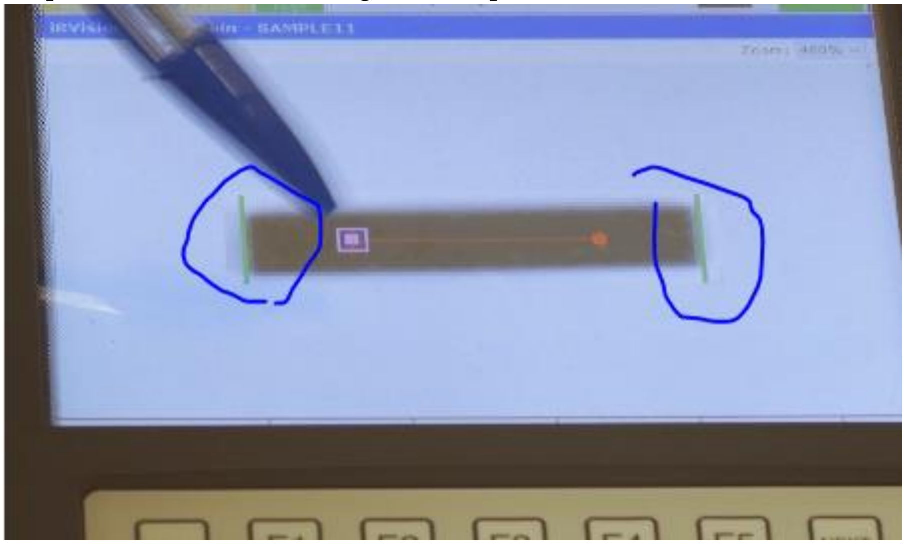
Above, the vision system, on the second try, accurately detects the part dimensions.

14. Now, a picture of the surface was taken by the camera through the user interface, this step is taken to test whether or not the vision system can accurately find the part.
15. Now the part is moved to another location to test if the robot vision system is still able to accurately locate the work piece:



Above, the vision system accurately locates the relocated work piece.

16. Next, the same steps as above were conducted in order to train the vision system the dimensions of the part, but this time the edges of the part were also trained:



Above, the edges of the part are trained.

17. The same processes were conducted to test the part at different locations in the work cell and see if the vision system could detect the part. The detection was a success.
18. For the final steps, the evaluation tool is selected on the teach pendant. The dimensions of the block include a length of about 76mm. Using the evaluation tool, a tolerance of +/- 3mm is tested for the detection of the block using the visions system. After the part is found, the part passes because it is within the tolerance length zone (79-73mm).

Discussion

From this successfully conducted experiment, it is easy to see how advanced and useful a robot vision system might be. Through the advanced optics detection systems, a robot arm such as the 6-axes Fanuc robot used throughout this report can harness the power of its reference frame capabilities using the vision system. Instead of only having frames of reference, the robot will essentially have optical “eyes” so that it knows where to move parts in its frame of reference automatically (without as much teaching of “dummy points” from invisible frames of reference) and can avoid and detect obstacles.

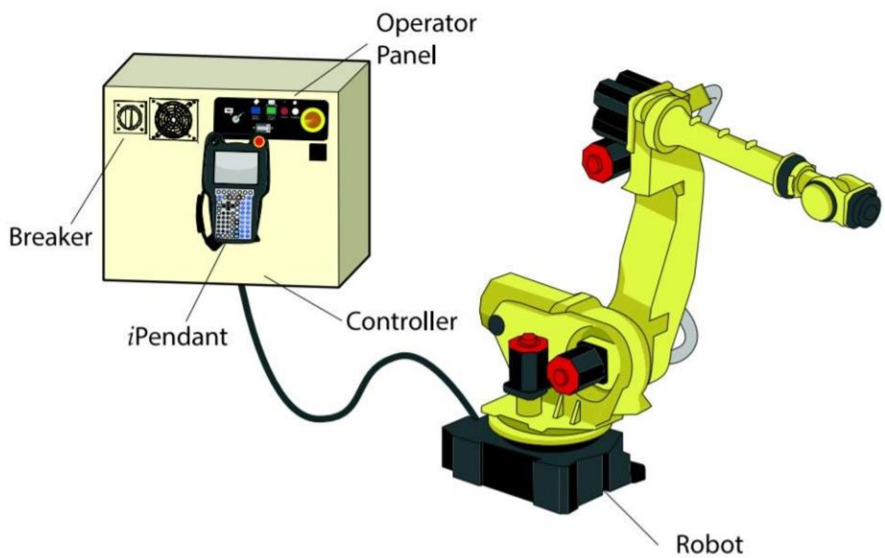
For example, instead of having to teach the robot arm a redundant movement to move a part, one could use the teaching and recording of points in conjunction with teaching the robot vision system the visual work cell so that together the robot can adjust and avoid obstacles, find obstacles, and accurately move obstacles. The robot arm is already advanced enough to do these things using its joint angles and world frame of reference, but the vision system is complimentary and greatly increases the teaching ability and reduces the time of robot arm program creations and “on the fly” adjustment capabilities.

VII. Apparatus (Equipment)

Teach pendant: the user interface for the 6-axes robot:

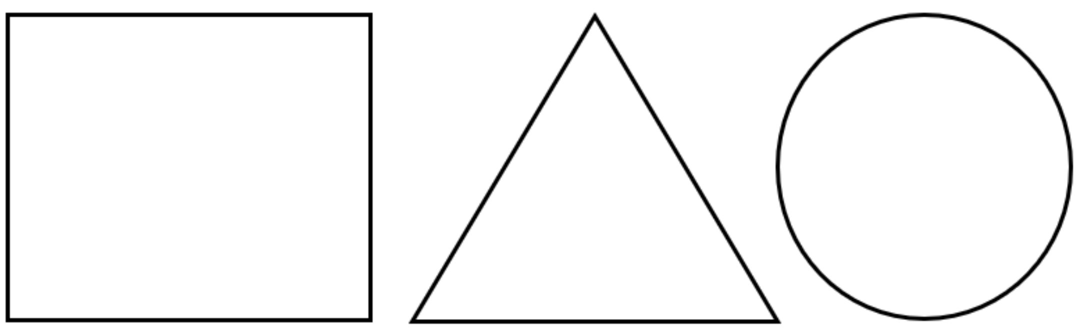
The Teach Pendant as mentioned earlier in this report is the user interface developed by Fanuc so that a user can interact with the robot controller and control the robot by jogging it or writing programs.

Robot Controller and 6-Axes Robot Arm:



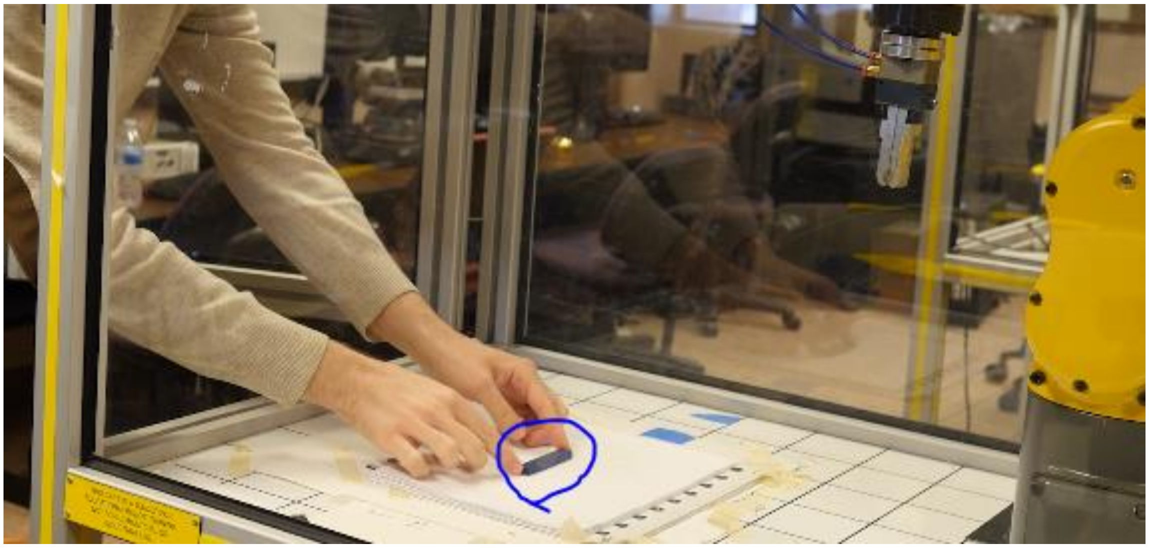
Also, as mentioned earlier, the robot arm is the actual mechanical output device that materializes movement commands sent to the robot controller. The robot controller takes all of the commands from the user and gets sensor information from the robot in order to make adjustments and tell the user its position. The controller is the intermediary, primary computing device. When a robot receives and sends information to the robot controller, the technical name for this type of robot system is known as a “closed circuit” system.

Shapes:



These were the various shapes used to show the process of using vertices and a special arc process when teaching the robot arm in a program how to trace each shape.

76mm Work Piece:



The small rectangular prism was used during the third experiment in order to use and test the robot vision system. It was moved to various locations to see if the camera was able to detect the part.

Circle Grid:

The circular grid was used in order that the camera system could have a point of reference in space. The spacing of the circles radius help determine a grid with a measure of accuracy by which to measure distances in the 2D space.

Robot Vision System Camera: Sony XC-56 Camera:

This is the advanced vision system camera used in the third lab experiment. It is connected to the robot controller so that the controller can receive the optical feedback from the camera sensors and make use of the spatial-visual information.

X. Discussion

A. Lab 1

For the first experiment, the basic functionality was demonstrated, including safety aspects. A basic program was written to show how a robot arm can be programmed. Right from the very first experiment conducted, one can easily obtain a measure of just how useful and complex just one of many robotics systems might be. Also, it is evident that learning and using a robot system can be very tedious and takes a very large amount of time, knowledge, and skills to fully harness complex, industrial capabilities. For instance, when jogging the robot or describing and understanding a frame of reference using the 3-point method shows that robot processes require thorough understanding.

B. Lab 2

In lab 2, the demonstrations from lab 1 are taken a step further; drawing shapes seems like a simple task, and it is. However, upon tracing the circle shape, the program required that another special kind of 3-point method be used to draw an arc and then be able to successfully draw the rest of the arc. These processes need calibration, and so it may take some time to give the robot controller the measure of accuracy required to get the desired results of the circle. Now, imagine applying this to advanced industrial processes, you can see it would take a lot of time and skill in order to safely, accurately, and efficiently develop a high-level (but extremely useful) robot application.

C. Lab 3

Finally, in the final experiment, the auxiliary robot vision system was tested. The application of the robot vision system can be just as sophisticated as the 6-axes robot arm. Now, putting the two complex systems together, and you have an even more complex system, which allows for even more complex tasks to be accomplished. The first two experiments proved to show the amount of knowledge and skills required to move the robot. The third demonstrated on its own the level of understanding to control the robot vision system alone. The advanced Sony camera allows the robot work cell to be detected not just logically through known robot joint angles and internal frames of reference, but by allowing a user to actually define a grid in space by which the vision system can find and detect parts. This would allow the two systems to work together to detect obstacles and move parts in the work cell to the appropriate, desired location. That level of sophistication requires yet another level of a highly skilled technician.

XI. Summary

In conclusion, this report accurately and sufficiently demonstrates the usefulness and complexity of an industrial robot. This study allows the reader to understand how and why (at least at a basic level) industrial robots can take over an industry. It allows for safety, precision, portability, and versatility in its ability to adapt and accomplish many kinds of tasks.

As the study progresses, it becomes more evident that such complexities can have a downside; the more complex a robot system becomes (such as adding auxiliary axes or systems like the robot vision system), the more time and skill that is required to develop and harness the capabilities of the robot system. So, if there is a ever an advanced robot somewhere and there is not enough skill and knowledge around, the advanced machine becomes useless; or, the full abilities may not be used. Also, if there is not enough time or if there is ever any damage, it may be difficult to reprogram or repair the robot in a timely manner.

Overall, though, the robot system is an advanced system that has developed into an unimaginably complex apparatus. There are endless possibilities to the things that one can do with an industrial robot. I predict that in order to make them more useful, developers need to make the robot advanced enough through more levels of abstraction, so that a basic user can still harness the advanced capabilities without having to have as much knowledge about the system behind the scenes. Such complexities should really be left to the actual technicians and computer science fields for reparations and development. There should be a greater measure of people who can take advantage of the robot without needing to know as much. This would make programming more difficult as in computer science, the more levels of abstraction the more difficult or time consuming it can be to develop, but the pay-off might be worth it.

XII. Acknowledgements

I would like to acknowledge the American Fanuc robot training program. Through their time consuming, but very thorough training, I was able to strongly put together how amazing robots are and understand how mechanics and computer science are brought together.

Also, I must thank the lab instructor Gabriele Galli for the lab demonstrations he conducted for the CIS-381 course at UM-Dearborn. It was unfortunate that we were unable to conduct the lab with the real robot, but we did at least get to see him demonstrate on campus through a video.

XIII. References

- https://training.fanucamerica.com
- University of Michigan-Dearborn, CIS-381 Winter 2021, Professor Cheol Lee, and lab instructor Gabriele Galli.