Robotic Inspection and Automated Analysis System

Maged Mikhail, Purdue University Northwest; Karen Abad, Purdue University Northwest

Abstract

Meeting customer requirements has been the main priority in industry. With time, though, these requirements have changed and customers are expecting more quality while spending the same amount of time and money. Thus, companies have developed various systems to improve their processes. The process focus has been to produce more quantity in less time. To accomplish this task, it is important to also consider defects. Defective products can cause delays in the production line and rework, as well as loss of money, downtime, and resources. In this current project, the authors focused on developing an integrated inspection system in which an integrated system, consisting of a conveyor belt, PLC, HMI, industrial robot, and motor drive, was used to perform the automated system. The vision system used a high-speed camera to perform the inspection. The detailed system design and a sample of the code and results are presented here.

Introduction

Inspection is a required process in many industries. Among the industries in which inspection is a required process, efficiency in power generation was (23.5%), followed by aerospace, military, and defense at 23.1% (Ahad, 2017). Results from previous research showed that visual inspection errors ranged from 20% to 30% being caused by human error or limited space. In another study, the author found that manual inspection was only 85% efficient (Andersen, 1985). Human inspection is not one of the most reliable methods, since it can lead to less consistent results due to human error and different levels of expertise (El-Agamy, Awad & Sonbo, 2016). Using a vision system can provide more reliability with better inspection results. Automating vision systems also provides cost savings to industries. Inspection by humans is estimated to be 10% of the workforce, or around 400,000, meaning an estimated $8 billion is dedicated to industrial inspection (Fathi, 1989). Investing in vision systems can improve the quality of the products offered to the customer, while reducing costs.

According to a survey conducted in 2016, only seven percent of the participants mentioned that 76-100 % of all the quality inspection processes are automated. Participants that automate some portion of the quality inspection process mentioned some benefits. The benefits mentioned by the participants included higher quality products, lower costs, reduced recalls, regulatory compliance, improved production throughput, less rework, and less waste (Ye, Chang, Pan, Chiang & Gabayno, 2018). The quality inspection can be performed before, during, or after production. Production occurs after a process such as labeling or capping has been completed. Detecting defects before a product is manufactured is an important step. It can help employees to implement corrective actions before delivering to the customer. It can also help to reduce rework, customer dissatisfaction, disruption in the manufacturing line, and delay of products to customers (Ye et al., 2018).

Literature Review

In recent years, there has been a growing interest in the development of automated inspection systems across various industries. Universities have been actively involved in educating students about these systems through coursework and practical projects. Notable examples include projects conducted at Michigan Technological University, the Oregon Institute of Technology, and the University of Alba Iulia. At Michigan Technological University, one project focused on incorporating robotics into the academic curriculum. This initiative involved the utilization of robots, a conveyor belt, and a vision system. The outcomes of this project encompassed the development of a comprehensive curriculum and hands-on experience for students in the field of robotics (Michigan Technological University, 2017).

Similarly, the aim of the project at the Oregon Institute of Technology was to enhance productivity and product quality by integrating a robot, a conveyor system, and a 2D-vision system. The primary focus of that project was to identify crucial factors for inspection and relevant defects that could impact overall product quality (El-Agamy et al., 2016). The project at the University of Alba Iulia centered around analyzing defects in porcelain production using computer-vision techniques. Through the implementation of a vision system and various tools, this project aimed to detect defects on porcelain plates and improve surface flaw inspection. The project concluded that employing this approach to quality control could have a positive impact on the porcelain industry by identifying defects earlier in the production process (Onita, Vartan, Kadar & Birlutiu, 2018).

In this current study, the authors used various pieces of hardware, including a programmable logic controller (PLC), an articulated industrial robot, a high-speed industrial vision system, a conveyor belt, and a variable-speed drive. These hardware components were interconnected using multiple software systems. What makes this work distinctive is that it successfully integrated all of these hardware components into a single system. This level of integration sets it apart from similar projects that have been carried out at other institutions.

Statement of the Problem

In industry, quality control is important in meeting customer expectation. Human operators perform most inspections. This method is not one of the most accurate, given that human operators can face fatigue while performing inspections all day long. The development of an automated analysis inspection system that collects data could allow for a testing system capable of detecting defects in bottles, acting, and sending feedback of the inspection results to the human machine interface (HMI) screen. This could provide insight into how data collection can help to improve a process or machines by getting real-time feedback from different production lines.

Purpose of the Project

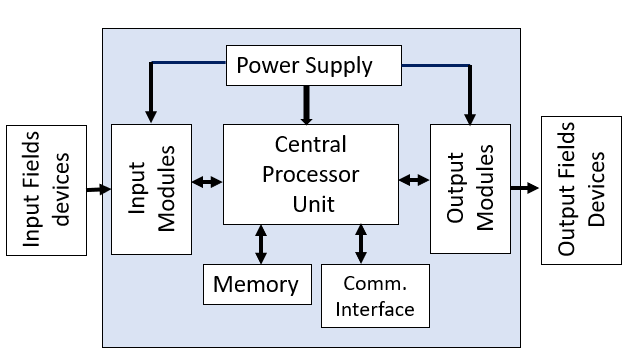
The goal of this current project was to design and program a working prototype of an automated inspection system that could collect and analyze data from an integrated inspection process. Results could provide insight into how this could be applied in industry to improve quality. The developed system was a comprehensive academic prototype intended for laboratory use, allowing students to gain hands-on experience with the concept of automated inspections.

System Components

The main components were:

* PLC
* Robot
* Conveyer belt
* Vision system

A PLC is a specially designed personal computer meant to work in a harsh environment, use programmable memory to store different instructions, and implement functions such as timing, logic, counting, sequencing, and arithmetic. These functions are implemented to control processes and different machines. PLCs are similar to computers. However, they are designed for industrial environments and control tasks. The PLC consists of a power supply, processor unit, input and output interfaces, processor unit, communications interface, processor unit, and programming device. Figure 1 shows the main components of PLC architecture.

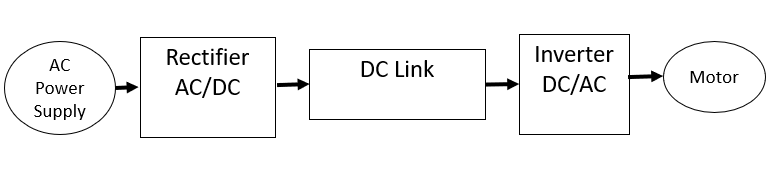


*Figure 1*. The PLC’s main components.

Devices that communicate with the PLC can be wired in two different ways: sinking and sourcing. During sourcing, the input module becomes the source of the current, which means that the device receives current from the input from positive to negative. During sinking, the input module becomes the sink of the current, which means that the device supplies current to the input.

The HMI is used in an industrial environment. HMI allows a user to interact with devices, machines, and systems. The functions of HMI are monitoring outputs and inputs, tracking production time, tags, and trends, and displaying data. An HMI can be present in different forms, such as computer monitors and built-in screens on machines and tablets.

A variable frequency drive (VFD) can control the rotational speed of an AC motor. This control can be achieved by controlling the frequency of the electrical power that is supplied to the AC motor. The system used to operate a VFD consists of a controller, AC motor, and user interface. Figure 2 shows the operational principle of a VFD.



*Figure 2*. VFD stages.

Assumptions

The scenario created in this project by the authors evaluated when a bottle passed or failed inspection, and real-time data were sent to the HMI screen. This concept could be applied in an industrial setting. For instance, on a production line, each product can be inspected once the process has been completed at the end of the line. For example, during the labeling process, an inspection system can be installed to see if the labels meet the specifications. During the filling process, another inspection system can evaluate if the liquid level is as specified by the operators. Collecting and analyzing these data for these processes can show if something is out of range, which can indicate a potential problem with the machine or the process itself.

Limitations

The project was not designed in an industrial environment, which could have helped to demonstrate this concept with machines on a real production line. However, integrating the robot, the PLC, a camera, and a conveyor belt helped to develop this concept using different machines. In addition, inspecting bottles helped to simulate a production line, where the caps and labels were inspected. This idea could be further demonstrated by connecting more PLCs and machines to the network and having all the data from different inspection systems talk to one another. The historical data could be shown in more detail; for example, the specific process that is facing problems and an acceptable range of failed items that are not related to a malfunction in a machine.

Hardware

The robot used for this current application was an LR Mate 200iD Fanuc industrial articulated 6-axis robot. The robot had six controlled axes and a maximum load capacity at the wrist of 4 kg, along with a mechanical weight of 20 kg, a repeatability of 0.01 mm, and a reach of 550 mm. A Cognex camera was used for this application. The camera was an IS2000M-130-40-125. Tables 1 and 2 show the main specifications for the camera model and the conveyor belt motor model (VWDM3538), respectively.

*Table 1*. In-Sight 2000 technical specifications.

|  |  |
| --- | --- |
| Attribute | Insight 2000 Cognex camera |
| Power | 24VDC, +-10%, 48W |
| I/O | One Acquisition trigger, general purpose input, and four general purpose outputs |
| Connectors | An industrial M12 Ethernet, a M12 Power &I/O |
| Speed | 75 fps for monochrome and 55 fps for color |

*Table 2.* Conveyor belt motor specifications.

|  |  |
| --- | --- |
| Attribute | WashdownDuty Motor |
| HP | 0.5 |
| Voltage range | 230/460 |
| Amps | 1.6/1.8 |
| RPM | 1725 |

Network Communication Design

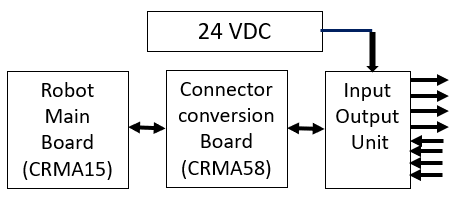
The communication protocol used for this project was Ethernet/IP. Devices were connected to a common switch. Table 3 shows the communication design and IP addresses used for each device. The subnet mask was 255.255.255.0 and the default gateway was 192.168.1.1.

*Table* 3. Assigned IP addresses.

|  |  |
| --- | --- |
| Device | IP address |
| PanelView Plus 600 | 192.168.1.108 |
| In-Sight 2000-130 | 192.168.1.20 |
| Powerflex 525 | 192.168.1.21 |
| Micrologix 1100 | 192.168.1.45 |
| PC | 192.168.1.11 |

Integration of the PLC with the Robot

Integrating the PLC with the robot allows better control of the operation, as everything can be controlled by the PLC program. Integrating everything into a common program also helps to have synchronization, so that when machines stop, the next one can start operating. Fanuc robots can be connected to peripheral devices by installing a connector converter board. The converter is compatible with the peripheral device control interface. The drawing number for this was A20B-2004-0410. Figure 3 shows how the connector converter board connected from the main board (CRM15 and CRMA16) and converted to the peripheral unit (CRMA58 and CRMA59). Table 4 refers to the Honda Tsushing Kogyo control interface, and has the peripheral digital inputs and outputs that go from the robot to the PLC.



*Figure 3.* Fanuc Honda Tsushin Kogyo MR-50RFD.

*Table 4*. PLC and robot I/O mapping.

|  |  |  |  |
| --- | --- | --- | --- |
| PLC OUT | Robot Out | UOP | UOP Signal |
| O:1.O/O | DO101 | UI 1 | IMSTP |
| O:1.O/1 | DO102 | UI 2 | HOLD |
| O:1.O/2 | DO103 | UI 3 | SFSPD |
| O:1.O/3 | DO104 | UI 4 | CSTOPI |
| O:1.O/4 | DO105 | UI 5 | RESET |
| O:1.O/5 | DO106 | UI 6 | START |
| O:1.O/6 | DO107 | UI 7 | HOME |
| O:1.O/7 | DO108 | UI 8 | ENBL |
| O:1.O/16 | DO117 | UI 17 | PNSTROBE |
| O:1.O/17 | DO118 | UI 18 | PROD\_START |

The following steps were performed to assign the peripheral I/O:

* Press the menu key
* Select I/O
* Press type which is the F1 key
* Select UOP
* Allocate I/O by pressing F2 config
* Change I/O assignment

Enter information for the rack, slot, and start point then determine if the information provided is accurate by looking at the STAT. If the values are valid, PEND will be displayed. On the other hand, if the value is not valid, it will show INVAL, and a power cycle allows validation of the new values. Table 5 shows the information that was used to assign the UOP.

*Table 5*. UOP configuration.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | RANGE | RACK | SLOT | START |
| 1 | UI[ 1- 1] | 48 | 1 | 1 |
| 2 | UI[ 2- 2] | 48 | 1 | 2 |
| 3 | UI[ 3- 3] | 48 | 1 | 3 |
| 4 | UI[ 4- 4] | 48 | 1 | 4 |
| 5 | UI[ 5- 5] | 48 | 1 | 5 |
| 6 | UI[ 6- 6] | 48 | 1 | 6 |
| 7 | UI[ 7- 7] | 48 | 1 | 7 |
| 8 | UI[ 8- 8] | 48 | 1 | 8 |

Figure 4 shows how the system required a camera so that it could recognize when a bottle failed or passed inspection. The Insight 2000 camera was selected, as it can send feedback to the system so that the total number of passed and failed items could be calculated. The Insight 2000 camera was connected to the PLC I/O so that the variables could be seen in the program.



*Figure 4*. Vision system setup.

The trigger connected to the digital output in the PLC. The connection of the digital output from the PLC was sourcing, so the camera connection was sinking. Table 6 shows the physical connection to trigger the camera. The output from the camera was sent as an input from the PLC. If the object passed or failed, these signals were sent as PLC inputs. Table 7 shows the physical connections of the camera outputs.

*Table 6*. Insight camera addresses.

|  |  |
| --- | --- |
| Insight Camera Trigger | PLC Embedded Output |
| Trigger (orange) | O:0.0/0 |
| Input Common (white violet) | +24 DC |

*Table 7.* Insight camera addresses the wiring connection with the PLC inputs.

|  |  |
| --- | --- |
| Camera Output | PLC Embedded Input |
| HS OUT 0 (blue) | I:0.0/0 |
| HS OUT 1 (grey) | I:0.0/1 |
| Output common (green) | +24 DC |

Insight Explorer was the program used to teach the object parameters to pass or fail the inspection. In this project, Insight Explorer 5.4.0 was used, since it needed to match the camera firmware version.

A PanelView Plus 600 HMI was used to display real-time information to the user about the inspection results. In this case, indicator lights indicated when the system passed or failed. And, pushbuttons allowed the system to be started and stopped. The HMI was connected to a 24 VDC power supply and showed the total number of items that passed and failed. The Ethernet port allowed communication with the Ethernet switch so that the program could be transferred to the HMI.

A PowerFlex VFD was connected to the conveyor belt so that all the functions could be controlled from the PLC. The communication used for this purpose was Ethernet/IP. The VFD had an output voltage of 120V, rated for the motor being used, and was wired to the motor on the conveyor belt. Table 8 explains the specifications for each port.

*Table 8.* PowerFlex terminal block specifications.

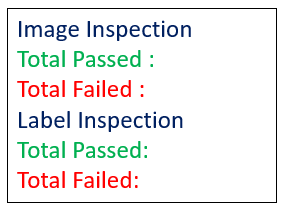
|  |  |
| --- | --- |
| R/L1, S/L2 | 1-Phase Input Line Voltage |
| R/L1, S/L2, T/L3 | 3-Phase Input Line Voltage |
| U/T1, V/T2, W/T3 | Motor Phase Connection |
| DC+, DC - | DC Bus connection |

After all the inputs and outputs were wired and assigned to perform different functions, a program was created considering the address for each device. The complete program can be requested from the authors. The first lines consist of starting the conveyor belt. Since the stop input from the VFD is normally closed, it was activated with the start input and with the frequency input that ran at 10 Hz. The start and stop were latched until the bottle reached a certain position, when the conveyor stopped. After the inspection time had been completed, the conveyor was started again.

In the next few rungs, a timer was used to track the position time of the bottles. The N11:18 address was used to send signals to other devices so that they could start operating. The position time would stay on hold when the inspection process was occurring and during the fail-remove bottle process. These rungs were used to stop the drive under different conditions. To stop the conveyor, the start drives and the stop inputs of the VDF had to be unlatched. The drive stopped when the position time reached the camera for three different bottles. An input bit could also stop the process after which a timer, that varied depending on the inspection results, controlled the time when the conveyor was stopped. If the bottle passed inspection, the timer was only set for five seconds. There were two inputs for passing the inspection, which were activated after the camera had taken a picture and the inspection had passed the cap and label requirements. A move function was used to auto-reset the timer whenever it was greater than the specified value.

Subsequent rungs were programmed based on when the bottle failed the inspection; in this case, the time assigned was 121 seconds, since that is the time it took for the robot to move to that location on the conveyor, pick up the bottle, and send it to another location out from the production line. It also had an auto-reset that went to zero after the timer had reached its limit. The trigger for the camera was activated when the three bottles reached the position time specified for each bottle. In the Insight Explorer program, there was a one-second delay so that the camera had time to focus on the bottles. A timer was created along with an output for each variable. The variables were set for when the bottle failed to have the cap and when the label failed to meet the requirements specified. Other variables were also created for when the bottles passed both conditions. The variables created in the earlier rungs were used in the next rungs in order to keep track of the total number of bottles that passed and failed each condition. After the bottle passed or failed, an inspection was required. As note previously, if the inspection passed, the drive would start immediately. However, if the inspection failed, the robot would pick up the bottle and remove it from the production line. The next group of rungs started the robot’s inputs, if the inspection failed.

To start the robot, it was important to clear any fault that may have existed before running. A timer of one second was included for this purpose. After the timer was done timing, it would enable the robot to select the program and start running. Once the fault had been cleared, the robot was enabled and the input terminals for selecting the program were activated. The PNS2 and PNS3 would send the binary number, 00000110, to the robot. The robot then read the number in decimal corresponding to PNS 0006. The program initialized after the robot was enabled. This signal enabled the PNSTROBE, which activated the signal that started the robot. The input to start the robot activated a timer to keep track of the time that it took to pick up the bottle and send it to the new position. This timer was used to deactivate other rungs while this process was happening. This had a move function that reset the time once it reached 121 seconds. Before programming the HMI, it was necessary to assign the path from which the tags would be dragged into the functions for the HMI programming. Figure 5 shows the layout of the HMI.



*Figure 5*. Inspection and monitor control system.

After the program was created, it was transferred to the HMI by creating a runtime application using the version that was compatible with the HMI. In this case, the version in the HMI was five, so the runtime created was 5.0. The file created was used in the transfer utility to download to the HMI. Insight Explorer had to be configured to get data from the camera and send it to the PLC. Figure 6 shows the first application step and consisted of selecting the In-Sight sensor to be connected. On the same page, a saved program could be opened to continue with previous work.



*Figure 6*. In-Sight Explorer application step.

A second pattern tool was created for inspection of the label. The inspection system detected when the label did not meet the requirements specified. Some of the common label defects included a folded label, stains, color fading, etc. In the last application step, the program could be saved as a file on the computer. The last step was when running the job: information could be obtained on how many bottles passed or failed the inspection using the first pattern.

Troubleshooting

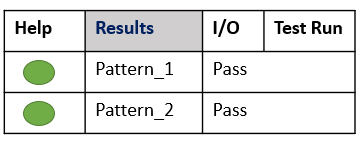
Changing the address for the PROD\_START robot UI.

When first testing the program to run the robot by selecting a PNS program, the UI did not start the robot. After verifying with the multimeter that all corresponding outputs and inputs were getting the required 24V, it was determined that the problem was internal; by switching the input to an available terminal, the problem was solved. In the first test, three bottles were used to test the inspection system. The bottles had different characteristics that included a label that was folded, the wrong cap color, and no defect at all. During the inspection of the bottle that had the folded label, the results from the software showed that pattern\_1 passed, which corresponded to the cap inspection; pattern\_2, which corresponded to the label, did not pass, due to the label being folded. Figure 7 shows that, in the I/O pallet function, the outputs assigned showed a value of 1 for pattern\_1 (pass) and for pattern\_2 (fail).



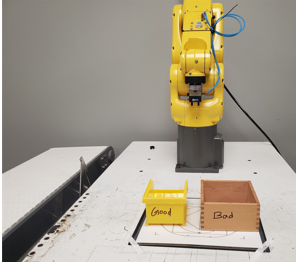
*Figure 7*. Failing inspection test.

The second bottle had a darker cap color and the results of the inspection showed a fail result for pattern\_1 and a pass result for pattern\_2. In the I/O section, the outputs had a value of 1 for pattern\_1 (fail) and pattern\_2 (pass). The results were sent to the PLC inputs, which used the data to keep track of the total number of bottles that passed or failed inspection. The next bottle that passed along the conveyor belt had everything under specification. The results on the palette function showed pass results for pattern\_1 and pattern\_2. In this case, the output results were the value of 1 for pattern\_1 (pass) and for pattern\_2 (pass). Figure 8 shows that the bottle would pass the inspection when both conditions were true.



*Figure 8*. Passing inspection test.

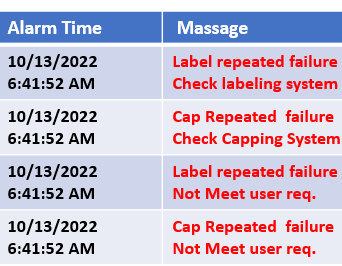
The first bottle with a folded label reached the position specified for inspection. Figure 9 shows that the robot took the proper action, based on the results obtained from the camera, then placed the bottle in the correct bin.



*Figure 9*. Robot placing the bottle in the proper bin.

Alarms

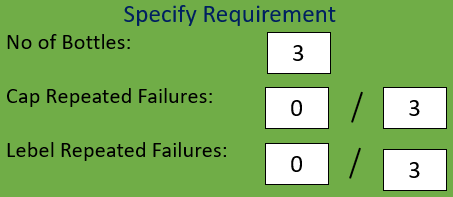
The alarms were configured on the alarm setup, which required creating tags and assigning the addresses that would trigger the alarms. In the messages section of the alarm, setup notifications were created for each triggered alarm. Bottles with caps and label defects were tested and the alarm notifications were shown on the HMI. If the user inputs were out of range, the alarm displayed a message regarding the problem so that an operator could act, analyze, and determine if the problem was with the machine or the process. The alarm appeared even if the user was looking at another screen. Figure 10 shows that, once the alarm was acknowledged, it would automatically close the pop-up notification.



*Figure 10.* Alarm notifications results.

Results

The HMI consisted of a numeric control and a numeric indicator for the number of bottles. Figure 11 shows that the HMI had numeric fractions that the user could manipulate to monitor repeated failures before the alarm was activated.



*Figure 11*. Update the HMI for monitoring the process.

System Testing

The testing application involved conducting various scenarios to validate the accuracy of the system. These scenarios included different combinations and arrangements of good and bad labels and caps. The camera successfully detected numerous defective labels and caps, as well as rejected deformed bottles that did not match the expected pattern. The inspection system effectively identified when a label did not meet the specified requirements, including common defects such as folded labels, stains, and color fading.

Results and Future Recommendations

The results of the testing led to the creation of different programs in the robot to sort the defects, which helped visualize the detected defects. Implementing an automated cart at the end of the production line to pick up the defects and distribute them for rework can significantly reduce the time required for reworking a certain item. However, there was a delay in running the program from the HMI, so in an industrial setting, the authors recommend using a SCADA interface for faster response. The camera required calibration and testing to ensure that it accurately recognized when something passed or failed inspection. After adjusting some parameters, the camera was able to provide accurate results. It is important to adjust the camera in an industrial scenario based on the items being inspected and the lighting conditions in the room.

Conclusions

Once every step mentioned in the report was completed, the result obtained was a system that collected data from an integration inspection process. To prove this concept, the authors designed a project that consisted of three bottles that moved along a conveyor belt and stopped in front of the camera to get inspection results. A robot was integrated to remove failed bottles from the conveyor. The entire process was controlled by a PLC. Program information was sent to the HMI software and displayed on a screen that showed real-time information about the total number of bottles that passed or failed.

References

Ahad, A. (2017). Software Inspections and Their Role in Software Quality Assurance. *American Journal of Software Engineering and Applications*, *6*(4), 105. <https://doi.org/10.11648/j.ajsea.20170604.11>

Andersen, E. (1985). A review of: *Total quality control* (3rd ed.). McGraw-Hill, New York, 1983. *Quality and Reliability Engineering International*, *1*(3), 214-214. <https://doi.org/10.1002/qre.4680010315>

El-Agamy, M., Awad, M., & Sonbo, A. (2016). *Automated Inspection of Surface Defects Using Machine Vision*. *Proceedings of the 17th International Conference on Applied Mechanics and Mechanical Engineering*, *17*, 1-13.

Fathi, Y. (1989). A review of: *Poor-quality Cost*. Marcel Dekker, Inc., New York, NY and ASQC Quality Press, Milwaukee, WI. *The Engineering Economist*, *34*(3), 264-264. <https://doi.org/10.1080/00137918908902992>

Ye, R., Chang, M., Pan, C., Chiang, C., & Gabayno, J. (2018). High-resolution optical inspection system for fast detection and classification of surface defects. *International Journal of Optomechatronics*, *12*(1), 1-10. <https://doi.org/10.1080/15599612.2018.1444829>

Michigan Technological University. (2017). Research and Development of Industrial Integrated Robotic Workcell And Robotrun Software For Academic Curriculum. <https://digitalcommons.mtu.edu/cgi/viewcontent.cgi?article=1446&context=etdr>.

Onita, D., Vartan, N., Kadar, M., & Birlutiu, A. (2018). Quality control in the porcelain industry based on computer vision techniques. 2018 International Young Engineers Forum (YEF-ECE). [https://doi.org/10.1109/YEF-ECE.2018.8368943](https://doi.org/10.1109/YEF-ECE.2018.8368943" \t "_blank)

Biographies

**MAGED MIKHAIL** is an Assistant Professor of Mechatronics Engineering Technology at Purdue University Northwest. He earned his BS from Cairo Egypt in 2001, MS in electrical engineering science in 2007 from Tennessee State University, and PhD in computer information system engineering in 2013 from Tennessee State University. Dr. Mikhail’s research interests include vision systems, robotics, and control. Dr. Mikhail may be reached at [mmikhail@PNW.edu](mailto:mmikhail@PNW.edu)

**KAREN ABAD** is a student at Purdue University Northwest. Mrs. Abad is pursuing a PhD in mechatronics engineering. Mrs. Abad may be reached at kabadnav@pnw.edu