

Automated programming robot component transfer system utilizing machine vision as detection interface

H J Vermaak and L Rogers

RGEMS Research Unit, Department of Electronic Engineering, Central University of Technology, Free State, Bloemfontein, South Africa

Abstract. Modern day automation systems rely on fixed programming routines to carry out their operations. If an automated flexible system is introduced onto such a production line, the complete reprogramming process required for new products needs could be automated with limited loss in production time. Therefore, instead of reprogramming each new position for the robot system the system takes over real-time control of the robot and carries out the required steps autonomously. The benefit with such a system would be that the robot would not need to be reprogrammed for every new routines but is controlled in a real-time environment to carry out new procedures based on external vision sensors. Using a real-time system could remove the need for a fixed programming environment and replace it with an automated changing programming setup. This could result in a system automatically adapting to a new product introduction through real-time machine vision processing techniques.

1 Introduction

The Assembly industry is a demanding, yet expanding market. The influence of the 4th Industrial Revolution is creating new opportunities within this sector. Flexibility within the product assembly industry is one of the most important features of the modern day assembly system. The assembly industry is driven by the need for new consumer products and the ability to respond to product changes rapidly [1]. Flexible assembly systems (FAS) must be able to convert quickly to the production of new product families and thus produced an increased variety of products in unpredictable quantities [2]. A FAS system can be a single product line or it can be reconfigurable to be flexible as needed by product demand [3] [4]. Flexibility can be expressed as the ability of a system to reconfigure itself for different product lines in response to changing requirements with limiting the down time [5].

The challenge for a flexible system that includes a robot arm is the reprogramming of the routines to adapt to the new product that need to be assembled [6]. Maintaining accuracy, speed and safety is very importance to ensure flawless operation of the system. This project will investigate an automated self-learning environment to allow for the control and programming of an assembly system.

2 Automated programming system

The aim of this project is to investigate the use of a vision system as detection sensor in developing a system that can learn new product designs in real-time, allowing the robot system to adjust accordingly without the need for

manual reconfiguration of the system. This in result can lead to an increased output in production in newly introduced products, while also having alternative implementation options. The vision system would scan the design pallet detecting the different components and calculating optimal movement paths for the robot system.

The designed system should be able to build a new product design completely autonomously without any user intervention or manual robot reprogramming. Once a new product is introduced onto the existing assembly line, the system should automatically adapt and build the new product considering that all preconditions are met such as the components needed to build the design is available within the component catalogue on the component pallets on the conveyor.

2.1 Test environment

The test setup needs to be developed to accommodate the vision system, the robot arm as well as the conveyor system that acts as the component transfer system. The test setup need to use the vision system to learn new product designs in real-time, allowing the robot arm to adjust accordingly without the need for manual reprogramming of the robot system. The test setup is shown below in figure 1.

2.2 Automated learning robot system

A KUKA robot was used in this case study with the RobotSensorInterface (RSI) software installed on it. The RSI software allows an external sensor system to take real-time control of an industrial robot. This ability of

the software allows an external sensor system to take real-time control of the robot and this creates the opportunity that the fixed programming of the robot can be replaced with an automated changing programming setup. A flexible component transfer system can be created using this self-learning robot system.

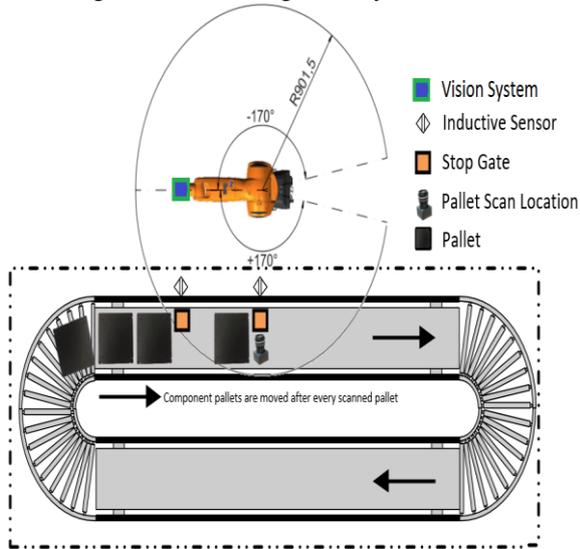


Figure 1. Test system layout including a KUKA Robot Arm.

RSI allows for continual influence over the robot motion by means of sensor data. In figure 2 the fundamental principles regarding Ethernet exchange are depicted. An object relates to a certain property or variable that will be updated through the RSI system. A certain object will be configured as either an inbound object or alternatively an outbound object. Different modes of correction can be configured namely [6]:

Motion suppressed sensor correction:

- Axis angle correction (absolute or relative),
 - Cartesian correction (absolute or relative),
- Sensor-guided motion

- Axis angle correction, absolute or relative
- Cartesian correction, absolute or relative

2.3 RSI system setup

This principle relating to the RSI setup utilizing a visual aid with its inbound and outbound objects will now be explained. The sensor cycle rate is the rate at which the signal processing is calculated. In other words, it's the cycle time defined that all work should be completed to and from the sensor system. The RSI system supports two sensor processing rates namely Input Processing Output (IPO) mode which sets the processing rate to 12ms. A faster processing rate is available using IPO_FAST mode which fixes the processing rate to 4ms. All data needs to be processed within this cycle rate in order for it to be a valid operation. There is a constant packet transfer between the robot and the sensor system even if no correction is currently in progress [7].

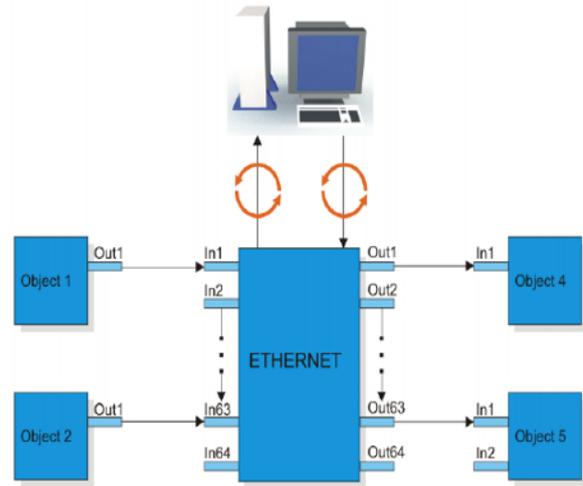


Figure 2. Data exchange configuration setup via Ethernet [6].

3 Vision system

The vision system is used as a remote detection and image acquiring sensor to the RSI software. It acquired images of components on the pallet and calculated each items relative position to the KUKA gripping device [6]. The centre of the pallet is taken as default position. This process was done by calculating the centre of gravity for each component. The centre of gravity was needed as it was used as the area to pick the components to ensure stability during moving operations [6]. The Accord.net framework software was implemented and used to calculate the orientation data for each component. These raw and central moments to the blob in question are implemented by the framework in order for a resultant angle to be calculated [8]. This calculation is very important when the need arise to rotate a components to fit in the new product template. The calculated centre of gravity for different shape components can be seen in figure 3 below.

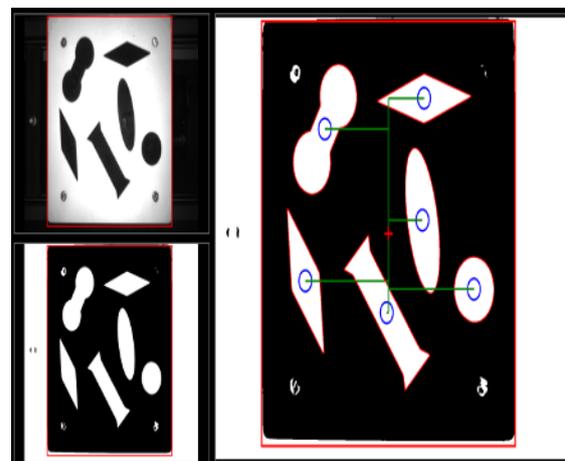


Figure 3. The calculated center of gravity for each component (small circle in each component).

3.1. Vision and robot calibration

First the robot calibration (positioning) was done taking into account that the camera need to be directly above the pallet when taking an image. The region of interest was then calibrated for the camera taking in account that the

region need to be within the borders of the pallet for the calculating purposes. Combining the calibration of the robot and the vision system produce a flexible vision-processing environment that can be used in different situations when needed.

Table 1. Overall Average accuracy results based on all builds

Build	X (mm)	Y(mm)	Angle(°)
1	1.7810555	1.391437	0.36042
2	2.6764228	1.134341	1.159303
3	1.3676975	1.421224	0.799562
4	1.5639524	1.54429	0.774123
5	1.7865532	1.797749	0.639775
6	1.289422	0.904378	0.965501
Overall Average	1.7441839	1.365569	0.783114

Table 2. Overall average time results based on all builds

Build	Time Taken(s)
1	342.54
2	340.94
3	342
4	334.6
5	342.51
6	341.01
Overall Average	340.6

4 System validation

The system compromise of multiple sub-systems (that can operate independently) but are required to work as one unit to achieve the objectives for this case study. In figure 4 below the block diagram for the system is shown. It is clear from the figure that all systems communicate is over Ethernet as previously explained. Each sub-system has the ability to operate independently, but the operating instructions are produced by the software controller as indicated in figure 4.

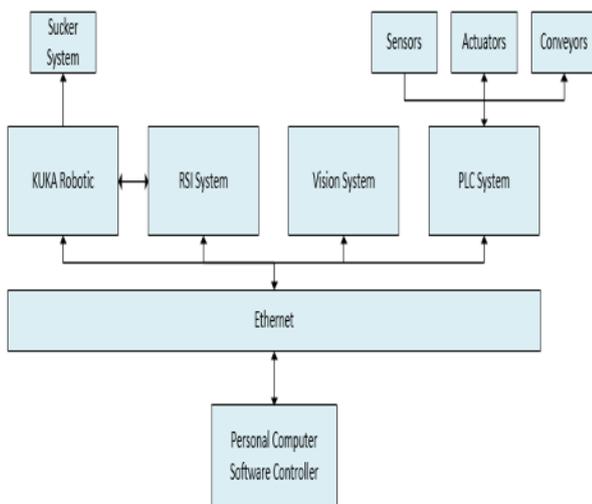


Figure 4. Block diagram of complete system indication the different sub-systems.

The system was challenged through various experiments that would require it for example to build a “Design” in the quickest most efficient manner possible. The speed and accuracy were recorded under different circumstances such as unique designs, similar components within the designs. Comparisons were conducted between the manual procedure of configuring the robot system and the automated process that this project presents when introducing a new design on an existing assembly system. Factors such as ease of setup, time consumption, reliability were all taken into account while maintaining an innovative thought process to allow leeway for the introduced system to thrive.

5 Results discussion and conclusion

It should be noted that the speed when picking and placing components was reduced due to the real-time control of the KUKA robotic system, movements between fixed points are still capable of running at 100% speed as these are fixed points within the coordinate system on the KUKA Robot.

The tables below summarize the builds data in order to get a more overall average of the accuracy and speed of the system. In terms of average the system achieved an average of within 1.74mm of the X-axis and within 1.37mm on the Y-axis. The angle of placement of the components achieved a more desirable average of within 0.78°. The system was calibrated to the best possible level, but theoretically if an even more accurate

calibration is achieved, the system could certainly improve on these averages. Many factors are taken into account that could affect calibration, such as movement of the system, camera accuracy (focus, lens) etc. with all these factors possibly affecting the calibration of the system. The speed of the system also yielded excellent results, with the system taking an average of 5.67 minutes to learn a design and re-build it into the correct configuration. This is most definitely beneficial in environments where constant unique pick and placing tasks are required, as time is saved reprogramming the KUKA robot system and the system can immediately rebuild the seen design in a short period of time with no changes in software needed. This time included the learning of the new design as well as pick and placing six components from the component conveyor over to the design conveyor based on the required configuration.

Overall the system performed optimally and completed all given tasks without any issues occurring. The systems accuracy and speed was impressive with “everyday” calibration in a changing environment. As stated before if the system could be calibrated to perfection, the results could be even more impressive. The accuracy of the system depends on the accurate calibration and the system could respond even more accurately when picking and placing components if the calibration is done more accurately. Future work may include providing the system with multiple components present and then need to search for correct components to assemble new product design.

References

1. Raj T, Shankar R and Suhaib M, 2007 “A review of some issues and identification of some barriers in the implementation of FMS”, *Int. J. Flex. Manuf. Syst.*, vol. 19, no. 1, pp. 1–40.
2. Vermaak H and Niemann J, 2017 “Virtual Commissioning: A Tool to Ensure Cost-Effective System Integration” *Int. Works. of Electr., Control, Measurement, Signals and their application to Mechatronics (San Sebastian)(Spain: IEEE)*, pp. 247-255.
3. Koren Y, 2006 "General RMS Characteristics. Comparison with Dedicated and Flexible Systems," *Reconfigurable Manufacturing Systems and Transformable Factories*, ed Berlin, Heidelberg: Springer, pp. 27-45.
4. Koren Y, 2010, “The Global Manufacturing Revolution: Product-Process-Business Integration and Reconfigurable Systems vol. 80: John Wiley & Sons.
5. Gyulai D, Vén Z, Pfeiffer A, Váncza J and L. Monostori, 2012 "Matching Demand and System Structure in Reconfigurable Assembly Systems," *Procedia CIRP*, vol. 3, pp. 579-584.
6. Rogers L and Vermaak H, 2017 “Automated Adapting component transfer system using real-time robot control within a KUKA ROBOTSENSORINTERFACE environment,” *Int. Conf. on Elec. Elect. and IT research (Cape Town)(Cape Town: IEEE)* , pp. 1474-1479.
7. Martin Edeberg PN, 2010 "Implementing Multi-Touch Screen for Real-time Control of a Robotic-Cell," Chalmers University of Technology, Sweden.
8. Luwes NJ, 2010 “Artificial Intelligence Machine Vision Grading System, DTech Thesis, Bloemfontein: Central University of Technology.