Offline programming with intelligent vision system of KUKA robot

Anouar MABROUK#1, Ramzi MEHREZ#2, Yassine BOUSLIMANI#3, Mohsen GHRIBI#4, Azeddine KADDOURI#5

Electrical engineering department, University of Moncton, Moncton campus

18 Av Antonine-Maillet, Moncton, E1A 3E9, New-Brunswick, CANADA

(1Eam7496, #2emr2775, 3Yassine.bouslimani, 4Mohsen.ghribi,#5Azeddine.kaddouri)@umoncton.ca

***Abstract*—In this paper, an offline programming approach for controlling a manipulator robot is presented using an open source software tool called RoboDK. A graphical user interface is considered in order to create, simulate and generate a program of virtual robot manipulation (off-line programming). This task is important in terms of reliability, flexibility and security. The main idea here is to give robots more decision-making autonomy, mobility, the implementation of exteroceptive sensors, including systems of artificial vision, for the perception of the environment. A solution based on artificial intelligence (AI) for controlling the robot is presented. This solution will allow robot manufacturers to master new markets in the industrial and service sectors. The Python interpreter was mainly used to develop a vision application, run an artificial intelligence-based detection and tracking algorithm and remotely control the industrial manipulator. The real-time implementation on a Kuka KR 6 R900 six-degree-of-freedom robot demonstrated the validity of the approach and its effectiveness for an industrial application.**

***Keywords*— Offline-programming; RoboDK; Python; Arificial-intelligence (AI); Vision**

I. INTRODUCTION

In recent years, technical improvements in industrial robots have been considerable. Robotization becomes more and more crucial in the manufacturing industry and robot programming in an industrial environment is limited to online programming methods, like KRL language for KUKA robots and KAREL programming for Fanuc Robots. This type of programming is useful, but it is difficult to handle and can waste a lot of time in the industrial manufactories, since the robot cannot be used for production purposes during its programming. So, this type of programming is more recognized in teaching fields.

The industrial robot is considered as a flexible production equipment, which is efficient, relatively inexpensive and reliable. In today's manufacturing industry, the three factors for maintaining competitiveness are the cost, productivity and speed. Reducing the time of production is a major challenge, since the variation in costs is proportional to the duration of manufacturing. Using off-line programming can reduce the time required for the creation of new programs from several weeks to few days. So, this type of programming can be presented as the best way to maximize the return on investment. That's what we called the robotization of short-run production in the industrial environment.

The robotic applications development requires appropriate software to facilitate the integration of robots into existing or new industrial processes, even in small and medium-sized enterprises. RoboDK is thus proposed among other tools or software. It is an open-source simulator that makes robot control easier and more affordable even for those who do not have wide experience in robot programming. RoboDK's application program interface (API) allows to manipulate different robot brands with the advantages of using high level programming languages such as: Python, C, C++, Visual Basic and MATLAB.

Using powerful programming languages like Python allow complex solutions to be developed and implemented in many robot applications.

Robotization and the use of modern technologies such as advanced vision systems and artificial intelligence applications will ensure the well-evolution of industries.

Following this introduction, the section two introduces the development of automation using industrial robots in manufacturing industries throughout history. The first part of the section three presents a small description of the KUKA robot used, then we will talk about the offline programming integration in the automation with industrial robots, RoboDK the development environment used in this project as well as the choice of the programming language Python. The process of this approach will be presented first, and offline programming will then be used to interact with an advanced vision system, 3D simulation software, and with the robot using sockets and API socket in Python. Details will be presented secondly while the experimental results will be analysed, and a conclusion will be given.

II. BRIEF LITERATURE REVIEW

The concept of an industrial robot really took shape only when engineers realized the difficulty of giving a machine the behaviour and appearance of a human being. Efforts have therefore been concentrated on the design of mechanical arms to carry out industrial tasks.

ISSN 2356-5608

For decades, research in industrial robotics has been progressing very fast and according to the field of their applications the researches are diversified. Currently, large companies compete to dominate the global market for industrial robotics. The field widens considerably until incorporating several areas such as medical, service and strategic applications. Since its integration in the United States for more than 50 years, industrial robotics has become very popular. In 2007, in the Atlanta Conference of Robotics Science and Systems (RSS), To address this issue, a group of experienced researchers decided to follow a coordinated strategy to monitor the process of integrating robotics into the US industry. They divided the work into four categories: manufacturing and logistics, emerging technologies, medical robotics and healthcare and robotics service [1].

Anis Meguenani worked on the development of robotic intervention and assistance systems that can interact with humans in complete safety. The applications presented in his works focus mainly on the problem of control [2]. Duchaine Vincent studied the commands of the robots intended for the interaction with the human [3], he found solutions to the problems of the cooperative movements and to the reaction to the collisions.

With the development of computer systems and sensor technology, researches are directed more towards the new generation of robotic cells, the control of manipulator arms by vision and artificial intelligence (AI).

One of the major obstacles to the integration of industrial robotization in small and medium enterprises is the complexity of programming. Recent researches are progressing on different programming methods as shown in the following table [4].

TABLE I
EXAMPLES OF PROGRAMMING METHODS

programming method	description
Operator assisted online programming	Completely manual process. An efficient and cost-effective method but only for a simple robotic system. For a specific application, the programming of the robotic system remains very difficult, time-consuming and expensive.
Development of OLP (offline programming)	More reliable and offers flexibility to changes in product design. Based on the use of the 3D model of the robot's working cell. The person in charge of programming becomes the engineer instead of the operator.
Programming using Augmented Reality	Created to make the OLP more interactive and flexible. It brings together the features of both types of online and offline programming.

A. *Off-line programming (OLP)*

OLP is a method of programming real robots in a virtual environment that overcomes time constraints and avoids the total shutdown of the process and its production. It facilitates the development by the integration of several development tools and programming languages that are often considered at the basis of learning for engineers. In this context, several achievements are made to facilitate the programming task, J. Golz et al have used the toolbox RoBO-2L on MATLAB to implement a control interface of a KUKA robot. This interface shows the advantages of the programming flexibility of robots which makes these systems available to universities for educational purposes [5].

Still based on researches, we noticed that off-line programming is becoming more and more exploited, several software programs have been developed and they are based on an accurate modelling of the robot. The following table shows some software dedicated to the programming of different types of industrial robots.

TABLE II
EXAMPLES OF OFFLINE PROGRAMMING SOFTWARE

Software	Robot Brands
Roboguide	Fanuc
KUKA-Sim, CAMrob	KUKA
RoboStudio	ABB
RobCAD	Technomatix
RoboDK	All brands

Konukseven and Abid create a control interface for the ABB IRB2000 industrial robot, the operator can develop a program with a different language than the appropriate language dedicated for this type of machine, and then he uses the interpreter to execute it on the robot.

This interface can be used for different control applications. Welding and machining work are characterized by specific tasks for each type of operation which requires great flexibility in programming, M. Bruccoleri, and al presented a control approach based on the integration of a simulation software that contains a control interface, the work intended for the welding operations of the different pieces. This solution is reliable in terms of flexibility, the operator can generate the models of pieces without interrupting the work of the cell [6].

Offline programming makes it possible to test scenarios by using simulation to avoid collisions and singularities. The integration of the usual programming languages allows the implementation of the computer vision as a powerful tool in the recognition of objects. But the computer tools used are usually connected with robots through servers that are not provided by the manufacturer, which presents a non-robust solution in terms of communication.

III. VISION SYSTEMS

The application of vision systems in the industry aims at quality control (detection of conformity of products), control of the industrial process. Several processing tools are

ISSN 2356-5608

developed, the lighting system, the image acquisition process and the processing tools are key factors in the success of a vision application. Choi Kyung-Hwa et al. Studied the influence of the lighting system on the results of their vision application for remote monitoring [7]. The results show the efficiency of choice of lighting type based on the light emitting diodes, but the lighting system remains relative to the intended application types.

For vision systems oriented towards the control of industrial robots, three types of this device are listed in the literature: 1D vision, 2D and 3D. COGNEX is one of the largest manufacturers in visual sensor design.

Due to the complexity of the detection spots and the requirement for the accuracy of industrial robots, several applications are based on the approach of strengthening the vision system with computer detection processes. Kinnell Peter et al. Worked on the combination of a 3D camera with object recognition algorithms to control a robot arm, they use cloud-based algorithms simulated situations to optimize the location of the camera mounted on the robot [8]. The results of this solution are impressive in terms of accuracy, while the processing time is important which is not motivating in industrial processes.

The most used systems in industrial environments are the 2D, 3D systems, thanks to their ability to inspect the work surface and to accurately indicate the location of the target in the work cell. Sometimes, depending on the sensitivity of the application, researchers combine the two systems to obtain more efficient solutions. Xinjian et al addresses a new technique for the control of a manipulator arm intended for the collection of different parts on a conveyor. They exploit the 2D vision for the speed of detection of the localization, which reduces the processing time, and the 3D vision to generate the spatial coordinates of the target to the robots [9]. Used two vision systems improve the speed, accuracy of vision process, but this solution is convenient for static objects on the conveyor and not for the case where objects are in a state of motion.

IV. MATERIALS AND METHODOLOGY

This part is to declare the used materials and method.

A. *Kuka Robot*

In this project, we worked on the Sixx KUKA KR6 R900 robot from the Sixx KR AGILUS family. The robot includes all the assemblies of an industrial system with the manipulator arm, the controller (a control cabinet), the tool (end effector) and the connection cables. It also includes a SmartPAD teach pendant, software with several options and accessories. The robot is classified as a small robot with six degrees of freedom. Its weight is equal to 52 Kg, able to carry a load of up to 6 Kg and has a high reach of 901.00 mm. The KUKA Robot language, also known as KRL, is a dedicated programming language used to control the KUKA robot.

B. *Integration of off-line programming*

To program robots online, the programmer must master the programming language specified for each type of robot such as KRL for the KUKA robot. On-line programming requires that the operator controls the robot with a control handle equipped with force sensors and attached to the robot controller.

Fig.1 The configuration of the KUKA KR6-R900 SIXX robot

In the programming phase, the robot cannot be used in the production line, which makes the programming process very long, so that the control programs are related to the level and skills of the operator. The complexity of programming and the high cost of skilled workers create many challenges for small businesses in automating their industrial processes [10].

The limitation of online programming is pushing researchers to develop new methods and tools for offline programming.

C. *RoboDK software tool*

RoboDK is a software used for offline programming of industrial robots. With this software we can program any robot from any brand. RoboDK will automatically optimize the robot path, avoiding singularities, axis limits and collisions. We have access for an extensive library of industrial robot arms, external axes and tools from over 30 different robot manufacturers.

From an external computer, the user can interface with the KUKA robot through RoboDK using the Graphical User Interface (GUI) that allows to create, simulate and generate programs. A complete simulation model can be established, and the program developed can be simulated virtually on the RoboDK software interface as shown in figure 2 before being applied directly to the real robot.

The RoboDK API (Application Program Interface) represents the different commands available for offline programming. Far from the standard use of the specific programming methods exposed by the provider, RoboDK API allows us to program the robot with a high-level programming language such as Python [11].

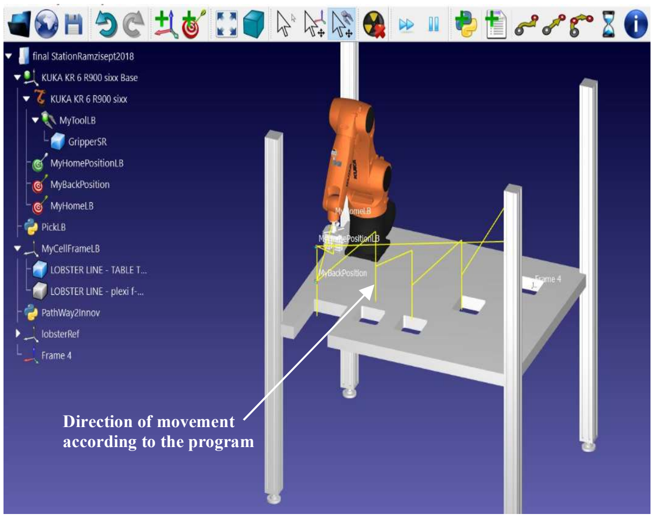

Fig.2 Virtual robot with RoboDK

Regarding the choice of the programming language to use, we chose to work with Python for several reasons. In fact, by installing RoboDK, Python is already installed by default so that we can start directly developing our program. More than that, Python is one of the most used programming languages for general-purpose programming. We also find that programming in Python with RoboDK make the work faster, help to integrate the systems more effectively and it is more useful given the availability of the necessary documentation.

We use RoboDK as software that can communicate with the robot, and to solve the problem of connectivity between the robot and the used computer for the development and programming. We decided to replace the existing connectivity based on a TCP / IP communication protocol (TCP (Transmission Control Protocol) and IP (Internet Protocol)), through connectivity based on the robot controller communication interface.

D. Vision system

Artificial vision systems and image recognition algorithms for robotic applications have evolved in recent decades [12]. Research in these fields is oriented towards the use of these systems to provide industrial robots with autonomy and intelligence to make complex processes more flexible and efficient. The development of vision systems begins with one-dimensional vision, then research is more oriented towards 2D and 3D vision, due to their efficiency and precision.

Environmental constraints in some industrial application areas affect the operation of the vision system. Taking as an example, underwater work and wetlands of the fishing industry. To ensure the accuracy of vision control in similar environments, we must use specified vision systems.

The disadvantage of these researches is that each vision system is intended for specific applications according to the constraints in the field of application. So, there is not a robust system that is valid for different processes. We must adapt the vision system to the working environment and the products to be handled, which can be diversified in colours, shapes and materials.

The major objective of the researchers is to minimize the processing time. In this project, the process is based on learning the shape of objects to the system that will create a database in which the classification of different targets is formed. The system compares the detected object in real time with the patterns stored in the database [13].

1) *The VisionTech package:* VisionTech represents a complementary technological package providing the object detection function of the KUKA robot. It includes an image acquisition and processing package for performing a basic correction to adjust the position of the robot manipulator arm relative to the location of detected objects. It also includes a plug-in for KUKA smartHMI. It is often used for detecting the position of components, deracking and for depalletizing.

The vision system includes a KUKA robot's KR C4 controller and one or more cameras as needed. In this project we used a single GigE camera. The communication between the controller and the camera is established via the image processing package.

2) *The WorkVisual software:* For offline configuration, programming and commissioning, we used the workVisual software on an external PC (Laptop service) which is connected to the KR C4 controller. KUKA WorkVisual offers a uniform interface and has comprehensive diagnostic options. It’s simple to understand and easy to operate since the requirements and the necessary steps for use are contained in the WorkVisual documentation delivered by the supplier.

By executing the programming steps, we find that the code is checked in the background because the errors are automatically overwritten. This contributes to minimizing risks as well as reducing system start-up time. This means that with KUKA WorkVisual we can move forward in the project in a coherent and efficient way [14].

We worked with a 2D vision system for this application so for a 2D configuration we need to use only one camera. During this configuration, several image processing tasks will be automatically assigned to the camera. Among these tools there is an extension file. In this file we find the inputs such as the reference image on which the processing task is based and the number of components to be detected. The outputs such as the position and the coordinates of the image, the value of the detection accuracy of the objects as well as a graphic representation of the image processing as shown in figure 3.

In this project, we work with two types of 2D templates. The template "Locate Parts Known Position", for this model the location of the different parts must be known, and the detection

ISSN 2356-5608

International Journal of Control, Energy and Electrical Engineering (CEEE)
Vol.8 pp.1-6

area is limited to the position of these objects. We have also worked on the template "Locate Parts One Stage", which is suitable for the components defined by the structures. For this model, the search area is limited to the entire image as shown in the figure 4.

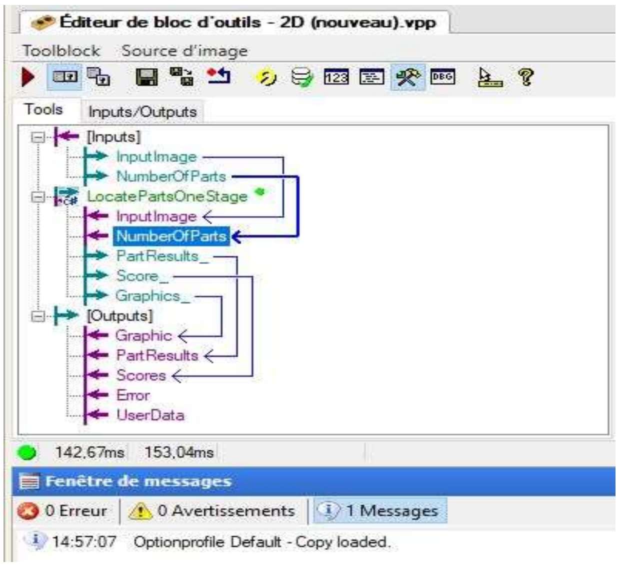

Fig.3 VisionTech plug-in

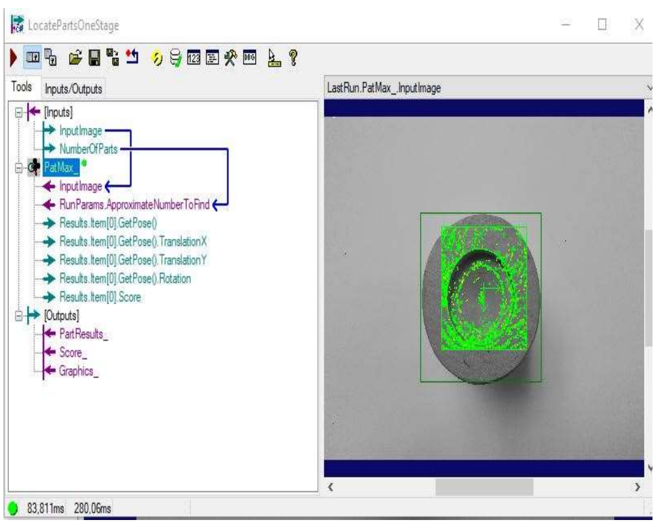

Fig.4 Vision process using the template "Locate Parts One Stage"

As a result, we have developed a vision system that respects the safety of operation, by the implementation of a recognition algorithm which eliminates the problems of variation in the colour and shape of objects to be detected and the detection system 2D makes it possible to increase the precision of our cell and optimizes the execution time, which was reduced to 280.06 msec as shown in Figure 4.

V. IMPLEMENTATION

An Artificial Intelligent application was developed to detect and to track the position of moving objects on a moving conveyor and functions as a trigger for the robot motions. The controlled robot arm picks the detected object and places it in a desired location. The AI application is built based on tensorflow and keras implementation for faster-rcnn [15]. We trained the first model on a small number of labelled images. According to the simulation, the training time of the model is equal to 76.833 msec if the recognition of the piece is 100% and it can go up to 132.8 msec if the recognition is 60% or more, as shown in Figure 5.

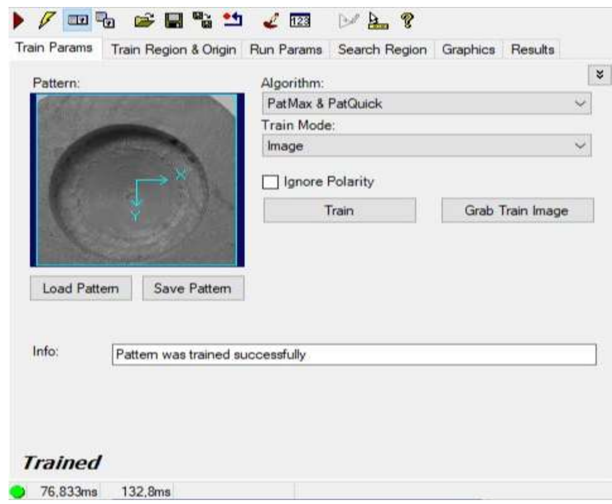

Fig.5 Training a model of images with WorkVisual

We used 300 images of the real object and labelling was made for 5 different futures. This model was prepared just to be trained in a real environment using live images that will be collected in the future. But the prediction has succeeded on most of the 30 images left in the test dataset and the next step was spending more time focusing on the robot programming. To work separately from the robot cells, simulation was essential. We have tested safely our applications on virtual robots to avoid collisions and to validate the used speeds, accelerations, paths, singularities, and the joint limits.

Indeed, we used RoboDK API for Python to develop and simulate an application in an industrial part sorting process. Subsequently, RoboDK was used to communicate with the Kuka KR 6 R900 sixx robot and its compact KRC4 controller to control them in real time.

VI. CONCLUSIONS

In this paper, we presented an approach for the control and the offline programming of an industrial robot. We have integrated different simulation environments which allowed us to exploit a high-level programming language.

The goal of this research was to implement a real-time control algorithm of an industrial robot arm based on an intelligent vision system. The quality criteria are the

ISSN 2356-5608

processing time of the task and the reliability of the control in terms of accuracy.

The communication interface with the robot has been used in a real cell for the detection and displacement of objects from different shapes, which approves its real efficiency in improving the accuracy, the robustness and the reduction of the time of implementation according to simulation results.

REFERENCES

[1] Henrik I. Christensen. *Formulation of a U.S* National Strategy for Robotics. IEEE ROBOTICS & AUTOMATION MAGAZINE. 6 Juin 2012.

[2] Meguenani Anis. *Safe Control of Robotic Manipulators in Dynamic Contexts*. A doctoral thesis. Pierre and Marie Curie University, Paris VI. Graduate School of Mechanical Sciences, Acoustics, Electronics and Robotics of Paris. October 13, 2017.

[3] V. Duchaine, *COMMANDE DES ROBOTS DESTINÉS À INTERAGIR PHYSIQUEMENT AVEC L'HUMAIN* , p. 229.

[4] Pan, Z., Polden, J., Larkin, N., Van Duin, S. & Norrish, J. (2012). *Recent progress on programming methods for industrial robots*. Robotics and Computer Integrated Manufacturing, 28 (2), 87-94.

[5] S. Guo, S. Pan, L. Shi, P. Guo, Y. He, et K. Tang, *Visual Detection and Tracking System for a Spherical Amphibious Robot*, Sensors, vol. 17, no 4, p. 870, April. 2017.

[6] M. Bruccoleri, C. D'Onofrio, and U. La Commare, *Off-line Programming and simulation for automatic robot control software generation*, in 2007 5th IEEE International Conference on Industrial Informatics, Vienna, Austria, 2007, p. 491-496.

[7] K.-H. Choi, J.-S. Kong, S. Lee, S.-H. Jo, and J.-K. Shin, *Light-adaptive vision system for remote surveillance using a smart vision chip*, in 2010 IEEE International Conference on Imaging Systems and Techniques, Thessaloniki, Greece, 2010, p. 270-272.

[8] P. Kinnell, T. Rymer, J. Hodgson, L. Justham, et M. Jackson, *Autonomous metrology for robot mounted 3D vision systems*, CIRP Annals, vol. 66, no 1, p. 483-486, 2017.

[9] X. Fan, X. Wang, and Y. Xiao, *A combined 2D-3D vision system for automatic robot picking*, in Proceedings of the 2014 International Conference on Advanced Mechatronic Systems, Japan, 2014, p. 513-516.

[10] L. Guillaume et C. Nicolas, « Robotique industrielle », p. 88. (2019), [online]-Availble:https://robodk.com/doc/en/RoboDK-API.html#RoboDKAPI

[11] Alexander Perzylo, Markus Rickert, Björn Kahl, Nikhil Somani, Christian Lehmann, Alexander Kuss, Stefan Profanter, Anders Billesø Beck, Mathias Haage, Mikkel Rath Hansen, Malene Tofveson Nibe, Máximo A. Roa, Olof Sörnmo, Sven Gestegård Robertz, Ulrike Thomas, Germano Veiga, Elin Anna Topp, Ingmar Kessler, and Marinus Danzer. SMEROBOTICS : Smart Robots for Flexible Manufacturing. IEEE ROBOTICS & AUTOMATION MAGAZINE .3 January 2019.

[12] J. Lee, S. Kang, K. Kim, J. Kim, and J. B. Kim, *A development of easily trainable vision system for the multi-purpose dual arm robots*, in 2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Jeju, Korea (South), 2013, p. 609-614.

[13] (2019) [Online]. Availble: https://www.kuka.com/fr-be/produits-et-prestations/syst%C3%A8mes-de-robots/logiciels/logiciel-syst%C3%A8me/kuka_systemsoftware/kuka,-d-,workvisual

[14] (2018)[Online].Availble:https://www.analyticsvidhya.com/blog/2018/11/implementation-faster-r-cnn-python-object-detection/