HAL
open science

Development of a methodology to improve the performance of multi-robot pick & place applications: From simulation to experimentation

Gaël Humbert, Minh Tu Pham, Xavier Brun, Mady Guillemot, Didier Noterman

To cite this version:

Gaël Humbert, Minh Tu Pham, Xavier Brun, Mady Guillemot, Didier Noterman. Development of a methodology to improve the performance of multi-robot pick & place applications: From simulation to experimentation. 2016 IEEE ICIT, Mar 2016, Taipei, Taiwan. 10.1109/ICIT.2016.7475067 . hal-01333219

HAL Id: hal-01333219

https://hal.science/hal-01333219

Submitted on 30 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Development of a methodology to improve the performance of multi-robot pick & place applications: from simulation to experimentation

G. Humbert, M.T. Pham, X. Brun, M. Guillemot and D. Noterman

***Abstract*—This paper deals with a new simulation tool for the improvement of multi-robot pick & place applications performance combining behavioral simulation of multiple robots and product flows. A contribution of the proposed work is to take into account in the simulation not only the scheduling rules of each robot, but also the robots collaborative aspect to ensure the desired overall performance for a given task (high-level programming). The transition from simulation to implementation of pick & place strategies is also an important issue tackled in this paper. By using a typical example consisting of comparing techniques to optimize the workflow, the utility of the simulation tool is shown. Experimental results validate the simulation results and demonstrate the interest of the developed methodology.**

I. INTRODUCTION

In recent years, the customers' demand of productivity and flexibility for their production lines has largely increased. This is why robots and robotic pick & place cells are more and more present in some industrial fields such as the food industry. In high-performance applications, typical characteristic of a pick & place robot can reach the following values : velocity 10 m/s, acceleration 100 m/s^2 precision +/- 0.1mm, pick & place cycle 0.40s on average. To improve the performance of these applications it is necessary to improve the design of current production systems (number of robots, performance ...) whilst also improving the management of flows and workload management when several robots are used.

A pick & place application is usually composed of a series of several robots installed in a line one after the other taking products on a first conveyor and placing them in boxes located on a second conveyor, see figure 1 [17].

On a multi-robot packaging cell, when there is no workflow optimization system, "pick" instructions are divided equally between the first robots. A final robot is added to deal with the products that could not be taken by the previous robots. Products initially assigned to a robot may not be taken because they finally are out of the robot workspace because of a lack of boxes to fill, for example. This kind of flow management approach is widely used in industrial application because it is simple to use but it has three disadvantages noticed by Schneider Electric:

G. Humbert, M.T. Pham and X. Brun are with Université de Lyon, INSA-Lyon, Laboratoire Ampère, 20 Avenue Albert Einstein, Villeurbanne 69621, France (gael.humbert,minh-tu.pham, xavier.brun)@insa-lyon.fr

M. Guillemot is with INSA-Lyon, 8 rue de la physique, Villeurbanne 69621, France mady.guillemot@insa-lyon.fr

D. Noterman is with INSA-Lyon, Laboratoire DISP, 8 rue de la physique, Villeurbanne 69621, France didier.noterman@insa-lyon.fr

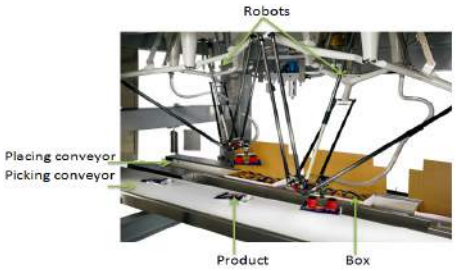

Fig. 1. Robotics cells with delta robots.

1) The first robots have an important individual workload close to 85%, which can lead to availability issues.
2) On the contrary, the last robot has a workload close to 25%.
3) A large number of products can not be picked and they are discarded. The average percentage of lost products could reach 1% in steady state and 5% in the transient state.

In industrial applications, field experiences are commonly used for the production line sizing. To the best of our knowledge, there are not efficient tools that can help either sizing the system or analyzing the performance analysis or improving the workflow. The main reasons are:

- The problem is complex: many parameters have to be taken into account (robots, conveyors, products...).
- Different skills must be used: robotics and flow management.
- There are no tools dedicated to both simulation and implementation. Some tools such as online programming exist but they do not support high level programming. This kind of feature could be interesting for pick & place applications where product flow simulation and collaborative aspects between several robots are taken into account.

However simulation approaches are beginning to be developed by some companies, but the performance of the software tools are difficult to assess:

- *Staübli*: *LineManager* software dynamically manages the workload [20].
- *ABB*: *Picking PowerPac* software provides a configurable environment to test different application [1].
- *Keba*: *Real World Simulation Software* Package software can test pick & place strategies [11].

- *Bosch Rexroth*: *IndraWorks* software dynamically manages the workload [11].

To the best of our knowledge, in industrial and academic context, there are no simulation tools that take into account the four following aspects: a behavioral simulation of the robots, a simulation of the work environment (products flow, boxes flow), the collaborative work of several robots and finally the possibility to go from simulation to experimentation. The developed code with the simulation tool has to be automatically generated for industrial controllers. The developed method in this article is use of a new tool based on the V-model, see Figure 2. The same tool is used for the sizing of a robotic cell, the development and the tests of the algorithms and for the passage from simulation and experimentation.

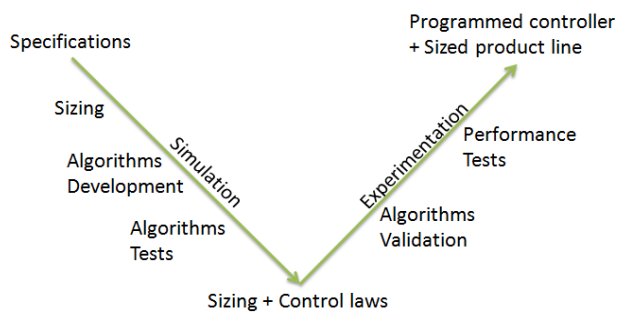

Fig. 2. V-model.

The first contribution of this paper is the development of a software interface that models the robotic cell part in a realistic 3D environment. The developed software is able to simulate realistic product and box flows, generate trajectories of the end effector to reduce the picking time and the placing time, propose several collaboration strategies between robots. This software is also modular and allows to simulate different configurations for a same application based on several parameters (number of robot, environment design, placement of the robots).

The second contribution is to propose a tool that includes experimental aspects in order to directly go from simulation to implementation. Simulation must be done in such a manner that the translation is as easy as possible to have a fast implementation in-situ, simple language and similar controller architecture are used in simulation and practice.

The third contribution is to show the impact of the presence or absence of collaborative strategies between robots following the safety margin as well as the impact of individual scheduling rules assigned to each robot.

Section II presents a new simulation tool dedicated to pick & place applications, the software environment and pick & place strategies are shown. Section III shows a comparative study between simulation and experimentation. A study on the influence of the presence of a pick & place collaborative strategy for multiple robots is also presented. Finally, a conclusion shows the interest of the developed methodology and tool.

II. SIMULATION OF PICK & PLACE APPLICATIONS

A. SIZING OF INDUSTRIAL APPLICATIONS

As indicated in section I, the sizing of pick & place robotic systems is usually done empirically. However, this empirical approach follows a decision sequence, which can be summed up in the following:

1) An input product flow is chosen by the customer, in terms of products per second FPs.
2) The customer generally knows the average time for a robot to do a pick & place task. This information gives an estimation of the product cadence per second $PPas$.
3) The system is oversized by adding a margin of safety Mfp to the products flows (10-15%).
4) The final number of robot NRf is calculated: $NRf = FPs*(1+Mfp)/PPas$ rounded up to the nearest integer.
5) The output box flow FBs is calculated based on the number of places NPb in one box: $FBs = FPs/NPb$.
6) The input Vi and output Vo speeds are either fixed by the customer or fixed arbitrarily, however, they need to depend on the flow and the spacing between the middle of the products EP and the boxes EB: $FPs = Vi/EP$ and $FBs = Vo/EB$.
7) There is no collaborative strategy between the robots, the individual scheduling rules are the same for all the robots and are "all-you-can-pick".

The industrial sizing method described above can be improved. One of the drawbacks is the first robots are more used than the last robots, which can cause a premature wear.

B. PICK & PLACE STRATEGIES

When a single robot is used, a queue is sufficient. Matton *et al.* [12] have proposed interesting online scheduling rules based on a queue. If several robots are used, more complex algorithms than a queue are necessary. The aim is to size the number of robots with respect to the number of products, boxes and conveyors. To do this, optimization algorithms could be used. Research works related to optimization algorithms used in robotic applications are numerous. Slim *et al.* [19] have compared three metaheuristic methods: ant colony optimization, genetic algorithm and particle swarm optimization. The aim was to maximize the number of picking products and to take also into account the execution time. Bonert *et al.* [2] have searched a solution of the Travelling Salesperson Problem with a genetic algorithm. Fujimito *et al.* [5] have used a genetic algorithm to seek the best combination of scheduling rules in order to obtain an appropriate flow. Mendelson *et al.* [13] have developed a decentralized palletizing multi-robot system using fuzzy logic. Edan *et al.* [4] have developed a fuzzy method. Isil Bozma *et al.* [3] have proposed an approach based on non-cooperative game theory to decide the robots actions in a multi-robots pick & place application. In all of these works, only simulation is used and the translation of the control laws from simulation to experimentation is unfortunately not tackled. Usually the optimization techniques above are computationally intensive and are not appropriate to real time application where

the computation cycle required is about a millisecond. Another alternative is to use this kind of off-line algorithms in simulation in order to schedule the best combination of rule then apply the result in practice without guarantee of results.

Petri nets have been used for pick and place system modeling. The aim of these works is mainly to synchronize two robots. Yasuda [22], [23] has developed a Petri net for a production system where two robots can exchange pieces. Zhou *et al.* [24] have proposed a Petri net which allows the work of two robot arms inside the same workspace. However in the pick & place applications, the robots are independent and their workspaces are not shared.

In the literature, a few patents are related to pick & place strategies. Izumi *et al.* [9] have filed a patent about conveyors sharing in order to balance the robots workload. Herzog [6] has filed a patent about a method of filling containers using the discreet event system. Wappling *et al.* [21] have filed a patent about a scheduling method for pick & place applications.

C. A NEW TOOL DEDICATED TO PICK & PLACE APPLICATIONS

There are several needs for this study. Firstly, the robots behavioral have to be simulated and the end effector has to be visualized to check if the end effector moves correctly and stops in the right position. Secondly, the work environment (product flow, boxes flow) has to be simulated in order to check the interaction between the robots and the environment and to verify the robot pick the right product. Thirdly, two levels of strategies (collaborative strategies and individual scheduling rules) have to be taken into account. Finally, the possibility to go from simulation to experimentation has to be taken into account in order to have an implementation in-situ the easiest and fastest possible.

In the literature, there are several works dedicated to robotic pick & place simulation, which are only used for visualization, to verify the kinematics and dynamics. They are also used for robot design to validate its behavior, its movements and its interaction with the environment (collision detection). Johari *et al.* [10] have used Workspace5 to visualize an entire robotic application system in order to detect collisions between robots and their environment. Sam *et al.* [18] have designed a pick & place robotic system using Solidworks Softmotion software to study the motion of a robot.

To improve the productivity of a pick & place multi-robots application, the flow management has to be improved. There are several programs that are able to simulate this. Mirzapourrezaei *et al.* [14] have also used Witness to evaluate various aspect of manufacturing systems. The objective was to enhance the productivity and efficiency of the line. Pegden *et al.* [16] have presented Simio, a software based on intelligent objects. Hindle *et al.* [7] have used Simul8 to answer the complex scheduling problem of sequencing part requirements through a composite manufacturing center. Nikakhtar *et al.* [15] have compared two simulation tools: Arena and Witness. However, all of these programs are mainly dedicated to flow simulation. Visualizations are very basic, and mainly focused on the flows. The displacement of products or boxes on conveyors can be visualized with Simio 3D or Witness but it can not have several products on the width of a conveyor and just the execution time of a robot is simulated and not the movement of the effector.

Unlike other works, the environment software developed in our work allows the creation of a virtual cell, working in 3D and in real time, based on the CAD model of a real cell. This simulated robotic cell possesses the same kinematics and the same dynamics as the real process, its environments can also be simulated: products arriving on a conveyor etc. Scenarii can be implemented to verify its behavior. A high-level layer can be used to implement a product pick & place strategy and a collaboration strategy between several robots if necessary. This software is also modular, it can configure the production system (robot, conveyor, etc.), its environment (products, boxes, etc.) and the different scenarii.

The simulation tool incorporates two levels of strategies shown in Figure 3. Simple individual scheduling rules (light red blocks) for a single robot which give to the robots (purple blocks) the products or the places position and can be *FIFO* (First In First Out). The robot picks the first product coming to its workspace.

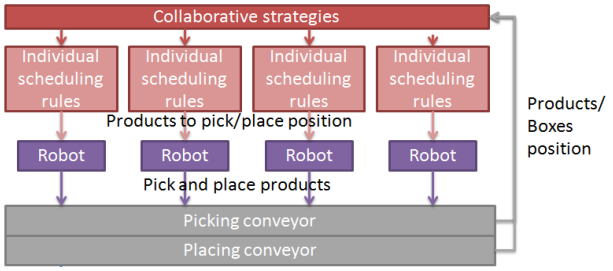

Fig. 3. Simulation architecture with the two levels of strategies (individual scheduling rules and collaborative strategies).

Then robots can pick or place products on the conveyors (grey blocks).

Collaborative strategies can also be used (dark red block). The products can be assigned to the robots before they arrive in the different workspaces. An example of different collaborative strategies between four robots is given in Figure 4. The red products are assigned to the robot 1, the green products are assigned to the robot 2, the blue products are assigned to the robot 3 and the yellow products are assigned to the robot 4. If there is an assignment by clusters (Figures 4.b), the last ones are sequentially treated. The lightest orange cluster is treated in the first row and the darkest orange cluster is treated at the end.

- *DownToUp*: Assign to the robots the products one by one from the downstream to upstream of the conveyor (Figure 4.a).
- *IntToExt Cyclic*: Assign to the robots the products one by one from interior to exterior of the conveyor for a number of products equal to the number of robots. And then the order of robots is cyclically changed (Figure 4.b).

Other assignment strategies were developed but will not be covered in this article for lack of space. They are presented in the article [8].

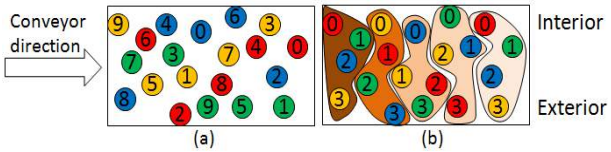

Fig. 4. Example of different collaborative strategy.

A collaborative strategy can also be applied for the placing. An example of placing strategy is *Linear* filling. A filling threshold is given for each robot. For example, for an application with four robots, the robot 1 fills the boxes until 25%, the robot 1 fills the boxes until 25%, the robot 2 fills the boxes until 50%, the robot 1 fills the boxes until 75% and the last robot fulfills the boxes.

The pick & place application creation consists of several steps, first the definition of the simulation model is carried out with 3D CAD software. At this stage the graphical objects and the kinematic behaviors of the application objects are defined in C++. The second step is the development of the individual scheduling rules for each robot and the collaborative strategies between robots in simulation in C++ (Step *Algorithms Development* of the V-model). These strategies will give to the robots, the position of the product or (the place) to reach. Finally, simulations could be run with the software(Figure 5), to test the model behavior and analyze the results of the different pick & place strategies (Step *Algorithms Tests* of the V-model). Once the parameters of simulation are optimized, experimental tests can be done to check the algorithms and test their performance (Step *Algorithms Validation* and *Performance Tests* of the V-model).

Fig. 5. Simulation example with four robots and two conveyors co-current with the *DownToUp* collaborative strategy.

After the simulations, algorithms and strategies translation in PLC language is carried out (passage between the simulation and the experimentation in the V-model). To facilitate this translation, the programs are written with the simplest possible functions, which also reduce the execution time. In addition, the controller software uses an object-based language similarly to the simulation software. The architecture of the Figure 6 is the implementation of that of the Figure 3. It is composed of the same scheduling rules (light red blocks) and collaborative strategy block (dark red block) than for the simulation. It includes the real robots (purple blocks) and conveyors (gray blocks). There is also the controller part with the control logic (green blocks).

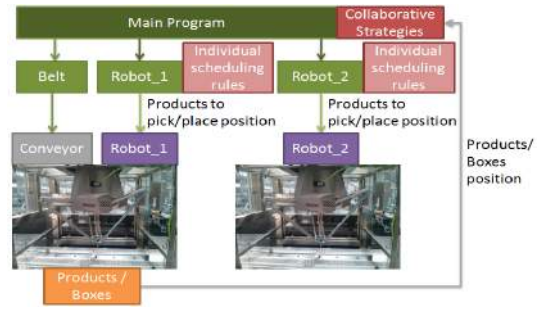

Fig. 6. Controller architecture with two level of strategies (individual scheduling rules and collaborative strategies).

III. COMPARATIVE STUDY BETWEEN SIMULATION AND EXPERIMENTATION

A study was conducted to compare simulation and experimentation. A demonstrator is used to check the program architecture, the algorithms and the strategies. It is composed of three Schneider Electric P4 delta robots, two parallel conveyors in co-current and is contolled by a Schneider Electric controller LMC600C see Figure 7.

Fig. 7. Demonstrator with three robots and two parallel conveyors.

Various parameters are analyzed and compared as the percentage of loss products, unfilled boxes, workload balance of the robots and the total workload of the system. The workload is defined by the following equation: $Workload = \frac{T_{Pick}+T_{Place}}{T_{Pick}+T_{Place}+T_{Wait}}$ where T_{Pick}, T_{Place} and T_{Wait} are respectively picking, placing and waiting time in second. The workload balance is defined by the difference of the robots workload and is calculated using the equation: $BW = \sum_{k=1}^{NR_f-1} |W_{k+1} - W_k|$ where W_k is the workload of the k_{th} robot. The more BW approaches 0 and the more the system is balanced. The total workload of the system is defined by the sum of the workload of all robots and is calculated using the equation: $TW = \sum_{k=1}^{NR_f} W_k$ where W_k is the workload of the k_{th} robot. The index TW gives an idea of the operation rate of the entire system. The more the index is big, the more the system is used.

The results of the Table I and II are obtained with the following assumptions:

- All the robots run with a maximum velocity equal to 5 m/s and an acceleration equal to 50 m/s^2. They pick and place only a single kind of product.

TABLE I
SIMULATION / EXPERIMENTATION RESULTS WITHOUT THE PLACING PRIORITY RULE.

Security Margin	Strategies	Lost Products (%)	Unfilled Boxes(%)	BW (%)	TW (%)
2.5 %	Nothing	1.57 / 1.36	20.15 / 18.32	0.053 / 0.111	279.9 / 280
	IntToExt Cyclic	2.17 / 1.74	24.33 / 20.15	0.126 / 0.197	280.1 / 279.9
	Linear	1.4 / 1.36	17.49 / 15.2	0.064 / 0.009	279.9 / 279.9
6.25 %	Nothing	0 / 0	0 / 0	7.62 / 2.51	272.3 / 277.1
	IntToExt Cyclic	0 / 0.13	0 / 0	2.88 / 1.26	277.5 / 278.6
	Linear	0 / 0.04	0 / 0	1.18 / 1.67	279.2 / 278.6
10 %	Nothing	0 / 0	11.9 / 11.1	14.9 / 14.9	265.5 / 265
	IntToExt Cyclic	0 / 0	11.1 / 9.91	3.54 / 6.91	277.8 / 274.1
	Linear	0 / 0	11.43 / 10.3	3.23 / 9.33	277.7 / 271.7

- The average pick & place cycle for a robot is performed in 0.74 seconds. $PPas = 1/0.74 = 1.35$ products per second can be pick by a robot.
- The picking and the placing logic follow the individual scheduling rules *FIFO*.
- The products in the input flow are randomly positioned.
- The boxes measure 0.13 m * 0.13 m and can contain 9 products ($NPb = 9$).
- The input conveyor speed is arbitrarily chosen equal to 0.2 m/s ($Vi = 0.2$).
- The distance between the boxes center is arbitrarily chosen equal to 0.15 m ($EB = 0.15$).
- The simulation time for each test is 10 minutes.
- The study is carried out in steady state.

The demonstrator consists of several robots, the impact of the presence or absence of collaborative strategies is studied. The strategies compared are *Nothing* (no strategy), *IntToExt Cyclic* and *Linear*. In parallel, a study was conducted to compare the influence of a placing priority rule. This rule stops the output conveyor if an unfilled box reaches the second half of the workspace of the last robot. Finally, the workload balance between the robots depending on the strategies and the safety margin is presented.

As indicated in the II-A, a safety margin is applied to the input flow to oversize the system. The influence of this safety margin is also studied. The comparative study is conducted for a margin of 2.5% to 6.25% and 10%. The input flow and the output speed are calculated respectively using the method described in the section II-A:3.95 Products/s, 3.80 Products/s and 3.64 Products/s and 0.0658 m/s, 0.0633 m/s and 0.0606 m/s.

The table I shows the results of simulation and experimentation according to different strategies and to different security margins. First we note that the results between simulation and experimentation are close and follow the same trend. There are some differences between the results because the movements of the end effector of the robot are not identical. The demonstrator and its program existed before that the simulation was created. The trajectories used in the demonstrator were not yet implemented in the simulation. However the simulation was conducted in such manner that the pick & place average times are very close between simulation and experimentation. Furthermore the demonstrator being a multi-use demonstration tool, there are intrinsic safeties that can slow the picking, reduce the working area etc. These securities are difficult to transcribe in simulation.

TABLE II
SIMULATION / EXPERIMENTATION RESULTS WITH THE PLACING PRIORITY RULE.

Security Margin	Strategies	Lost Products (%)	Unfilled Boxes(%)	BW (%)	TW (%)
2.5 %	Nothing	2.08 / 1.41	0 / 0	0.188 / 0.249	280 / 279.8
	IntToExt Cyclic	2.59 / 1.77	0 / 0	0.125 / 0.214	280.1 / 279.9
	Linear	4.03 / 1.46	0 / 0	0.002 / 0.043	280.4 / 279.9
10 %	Nothing	0 / 0	0 / 0	17.6 / 19.6	262.3 / 260.1
	IntToExt Cyclic	0 / 0	0 / 0	7.06 / 11.3	274.1 / 269.6
	Linear	0 / 0	0 / 0	6.04 / 9.42	275.1 / 269.7

We remark that for a safety margin higher or equal than 6.25% there are a very little lost products (less than 0.13%). Because there is a calculation time required for the robots to make the decision to pick or place a product, below this threshold, the system is undersized. In addition, beyond this threshold there is still the presence of unfilled boxes, which is due to rounding errors in the calculations.

We can add that for an oversized system, the collaborative strategies can fill more boxes than without any strategies (below 10.3% against 11.1% of loss for a margin equal to 10%). In addition the workload balance between robots is better (less than 9.33% against 14.9% for a margin equal to 10%). However, the total workload is increased (greater than 271.7% against 265% for a margin of 10%). For an undersized system, which can happen during an increase of the input flow, the strategy *Linear* allows to lose fewer products and boxes (15.2%).

Table II shows the results of simulation and experimentation based on different strategies and following different safety margins using placing priority rules. First we note that the placing priority rule allows to have no unfilled boxes. To an undersized system, it increases slightly the number of lost products (1.46% against 1.36% with the strategy *Linear*). While for an oversized system, the number of lost products is always zero. The placing priority rule add a slight decrease of the workload balance between robots (9.42% against 9.33% with the strategy *Linear* and a margin equal to 10%).

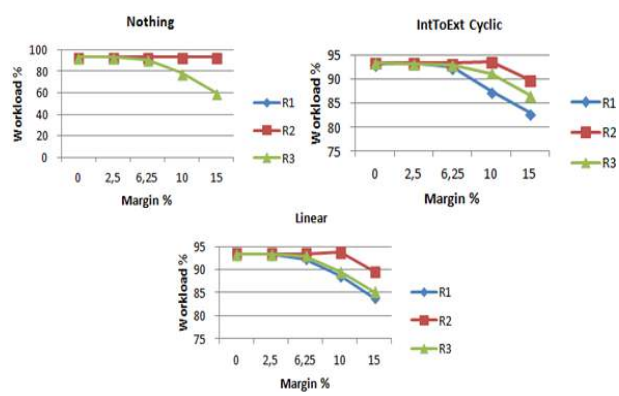

Fig. 8. Distribution of the workload between the three robots according the security margin.

Figure 8 shows the distribution of the workload between robots

according different strategies and safety margins. When there is no strategy, the first robots work at 100% regardless of the safety margin. The workload of the last robot decrease when the security margin is higher than the threshold between under-sizing and over-sizing. With a collaborative strategy, the workload of all robots decreases. However, the workload of the middle robot decreases after the other robots.

IV. CONCLUSION

In this paper, we proposed a new tool to improve the performance of multi-robot pick & place applications. This tool is based on the real-time 3D simulation of the robot tool and of its environment, allowing also the implementation of individual and collaborative control strategies. This tool allows the incorporation of all the steps of the creation of a pick & place application: robot behavior, workflow management, environment and transition from simulation to implementation on site. It allows to take into account all these parameters early on the final program and so to avoid making new development between each step.

A study was done to compare different individual and collaborative strategies in simulation and experimentation. The study showed that the simulation results are close to the reality and follow the same trend. In addition, this study allowed to see the impact of the presence or absence of a collaborative strategy. For a multi-robots system, the presence of a collaborative strategy balances the workload between robots and thus reducing premature wear of the last robot.

An interest of this tool is to test different algorithms before the implementation on site, to ensure that they function correctly and to know what is best. This avoids stopping a production line for these tests or save time if the line is in development because these tests can be performed in parallel with its construction and therefore help designing the production line. Another interest is the consideration of the following four aspects: simulation of the robots behavior, simulation of the working environment, collaborative strategies of several robots and the ability to go from simulation to experimentation.

Future work will be focused on the study and analysis in simulation of different configurations (number of robots, product flow, simulation time) to reinforce the results shown in this article. Another work is to improve the ability of the application to be independent in order to cope with a changing of the input products flow for example. Another aspect is to improve the simulation to be more realistic (better trajectories). Automatic code generation between simulation and experiment can be also considered because the simulation and experimentation architectures are similar and the simulation program uses simple functions.

ACKNOWLEDGEMENT

This work was partially supported by Schneider Electric with the CIFRE 158/2013.

REFERENCES

[1] ABB. Abb robotstudio picking powerpac simplifies the design and programming of complex robotic pick-and-pack systems, April 2014.

[2] M. Bonert, L.H. Shu, and B. Benhabib. Motion planning for multi-robot assembly systems. *International Journal of Computer Integrated Manufacturing*, 13(4):301–310, Nov 2010.

[3] H. I. Bozma and M.E. Kalalioglu. Multirobot coordination in pick-and-place tasks on a moving conveyor. *Robotics and Computer-Integrated Manufacturing*, 28:530–538, Aug 2012.

[4] Y. Edan, S. Berman, E. Boteach, and E. Mendelson. Distributed multi-robot assembly-packaging algorithms. *Intelligent Automation and Soft Computing,*, 10(4):349–357, Mar 2004.

[5] H. Fujimoto, I. Tanigawa, K. Yasuda, and K. Iwahashi. Applications of genetic algorithm and simulation to dispatching rule-based fms scheduling. In *Proceedings 1995 IEEE International Conference on Robotics and Automation*, volume 1, pages 190–195. IEEE, May 1995.

[6] Florian Herzog. Method and apparatus for filling containers with piece goods, Feb 2003. Patent US 2003/0037515 A1.

[7] K. Hindle and M. Duffin. Simul8-planner for composites manufacturing. In *Proceedings of the Winter Simulation Conference, 2006. WSC 06.*, pages 1779 – 1784. IEEE, Dec 2006.

[8] G. Humbert, M.T. Pham, X. Brun, M. Guillemot, and D. Noterman. Comparative analysis of pick & place strategies for a multi-robot application. In *20th IEEE International Conference on Emerging Technologies and Factory Automation ETFA 2015*, Luxembourg, Sep 2015. IEEE.

[9] Tetsuro Izumi, Kenichi Koyanagi, Kenji Matsukuma, and Yukio Hashiguchi. Robot system, May 2013. Patent EP 2 586 576 A1.

[10] N.A.M. Johari, H. Haron, and A.S.M. Jaya. Robotic modeling and simulation of palletizer robot using workspace5. In *4th International Conference on Computer Graphics, Imaging and Visualization (CGIV 2007), August 14-16, 2007, Bangkok, Thailand*, pages 217 – 222. IEEE, Aug 2007.

[11] Keba. Real world simulation package, 2014.

[12] R. Mattone, L. Adduci, and A. Wolf. Online scheduling algorithms for improving performance of pick-and-place operations on a moving conveyor belt. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA-98, Leuven, Belgium, May 16-20, 1998*, volume 3, pages 2099 – 2105 vol.3. IEEE, May 1998.

[13] E. Mendelson, O. Nayer, S. Berman, and Y. Edan. Behavior-based control of multi-robot assembly palletizing systems. In *Proceedings of the 5th Biannual World Automation Congress, 2002*, volume 14, pages 1–6. IEEE, 2002.

[14] S.A. Mirzapourrezaei, M. Lalmazloumian, A. Dargi, and Kuan Yew Wong. Simulation of a manufacturing assembly line based on witness. In *Third International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), 2011*, pages 132 – 137, July 2011.

[15] A. Nikakhtar, Kuan Yew Wong, M.H. Zarei, and A. Memari. Comparison of two simulation software for modeling a construction process. In *Third International Conference on Computational Intelligence, Modelling and Simulation (CIMSiM), 2011*, pages 200 – 205. IEEE, Sept 2011.

[16] C.D. Pegden and D.T. Sturrock. Introduction to simio. In *Simulation Conference (WSC), Proceedings of the 2009 Winter*, pages 314 – 321. IEEE, Dec 2009.

[17] Hakan Sahin. *Design of a secondary packaging robotic system*. PhD thesis, Middle est technical university, Dec 2005.

[18] R. Sam, K. Arrifin, and N. Buniyamin. Simulation of pick and place robotics system using solidworks softmotion. In *International Conference on System Engineering and Technology (ICSET), 2012*, pages 1 – 6. IEEE, Sept 2012.

[19] Daoud Slim, Chehade Hicham, Yalaoui Farouk, and Amodeo Lionel. Efficient metaheuristics for pick and place robotic systems optimization. *Journal of Intelligent Manufacturing*, 25:27–41, Feb 2014.

[20] Staubli. Robot linemanager system, 2013.

[21] Daniel Wappling and Steve Murphy. Pick and place, Jun 2012. Patent US 2012/0165972 A1.

[22] Gen'ichi Yasuda. An object-oriented multitasking control environment for multirobot system programming and execution with 3d graphic simulation. In *International Journal of Production Economics*, volume 60-61, pages 241–250. Elsevier, 1999.

[23] Gen'ichi Yasuda. Implementation of distributed control architecture for multiple robot systems using petri nets. In *Petri Nets - Manufacturing and Computer Science, Dr. Pawel Pawlewski*, pages 75–94. InTech, 2012.

[24] MengChu Zhou and Ming-Chuan Leu. Petri net modeling of a flexible assembly station for printed circuit boards. In *Proceeding IEEE International Conference on Robotics and Automation*, volume 3, pages 2530–2535. IEEE, April 1991.