ELEKTROTEHNIŠKI VESTNIK 79(3): 123-128, 2012
ENGLISH EDITION

Justification of using simulation software in robotised palletising applications

Karl Jerman[1], Boštjan Murovec[2]

[1] ABB d.o.o., Koprska ulica 92, 1000 Ljubljana, Slovenia
[2] University of Ljubljana, Faculty of electrical engineering, Tržaška 25, 1000 Ljubljana, Slovenia
E-mail: karl.jerman@si.abb.com

Abstract. The industrial robots of today are programmed by using programing units or designated software programs on separate computers (off-line). Despite the fact that manufacturers of industrial robots offer a very capable simulation software, off-line programing is not commonly used for palletising with industrial robots. One of the reasons are also the costs associated with the purchase of the simulation software. The aim of our investigation tries to establish when using the simulation software in projects, involving palletising with industrial robots, is cost-efficient. Each of the variables significantly affecting projects are specified. Project phases in which using software has an important role are the phase of line building, to be validated from supplier's point of view and the phase of production running, to be validated from the customer's point of view. A threshold calculation is made in three parts. Recommendations for the robot solution providers are given with the goal of increasing the use of the simulation software in robotics applications.

Keywords: palletizing, industrial robot, robot programing, simulation software

1 INTRODUCTION

High market demands are forcing manufacturers, to offer new products on the market in shorter time intervals. This is also reflected in automatisation of manufacturing processes, where suppliers plan manufacturing equipment for new products which often includes industrial robots [1]. At the time of inquiry, products which will be manufactured by robots are often only in the design phase, so they cannot be physically tested or validated. This reduces the time, needed by manufacturers to design, install and start up the robotics equipment. The same applies when adapting the existing manufacture for new products or making changes on the existing ones.

Another important aspect is the cost of projects, using industrial robots.

Optimization can be achieved by reducing the amount of materials and the time for project completion, using of cheaper building blocks and simplificating solutions. By using a simulation software for programing industrial robots [2], savings are considerable, fast and measurable.

All major manufacturers of industrial robots are offering simulation software, the so-called programs for off-line programing [3-6], which are used by suppliers of robotised manufacturing equipment. Suppliers are manufacturers of industrial robots or their integrators. Nevertheless the simulation software is not often used in robotic palletising. There are several causes for that, the following three are the key ones.

1. Programmers programing in the local programing language of robot manufacturer do not want to make any changes, because they can be very complex. Complexity comes from the possibility of using complete range of commands and operations provided by the standard programing language.
2. If additional tasks are to be added to palletising itself, or when tolerances of products should be bigger there may be some, limitations on the purpose-made simulation software. Additional tasks are much easier to handle in hand-written programs.
3. Customers find the simulation software to be an additional unacceptable cost, for which reason they do not order it.

In this paper we are trying to establish if using the simulation software is economically justified. Because of the many effecting variables being cooped with in individual projects, we must first define measurable variables. Using the simulation software is investigated from the view point of the end customer and from the view point of the supplier of the manufacturing equipment. By evaluating investigation results, the increase in the use of the simulation equipment is assessed and the optimal way of achieving it is defined.

2 SIMULATION SOFTWARE

Each manufacturer of the robotics equipment has its own approach to the simulation software. Consequently, most of software equipment can only be used for products of a certain robot manufacturer because of the kinematics of individual manipulators [7]. There are also programs on the market which enable simulation of robots of different manufactures [8], but the simulation capabilities are usually small compared to the specific software offered by individual robotics manufacturers.

Received August 2, 2012
Accepted October 23, 2012

Cost-efficiency is based on capabilities of the currently available software called RobotStudio of the ABB manufacturer [9]. It contains a virtual copy of the software used in the ABB controllers. They enable robot program to be directly transferred from the computer to the controller. The structure of the program is the same as that where the operator manually controls the robot, storages points and connects them into trajectory.

2.1 Simulation software for palletising

For packaging applications, ABB has developed a special add-on called the RobotStudio Palletizing PowerPack RSPPP [10]. There is now the third generation of this palletising software available. It was launched for the first time in 1998.

When using RSPPP, the operator does not have to know how to program by using the ABB programming language RAPID [11]. He programs one level higher and uses graphical symbols and rules to graphically define the complete robot program for palletising on a separate computer. RSPPP creates the RAPID programing code according to set parameters and defines robot trajectories taking in account the operator's selection of the pallet pattern, position of the in-and out-feeders, etc. The program is created on a computer and then transferred to the robot controller. The operator then defines the actual position of in-feeders, out-feeders and pallets in a real robot cell and starts up the robot program. RSPPP also enables correction of the program on the robot controller and transfer back to RSPPP.

RSPPP simulates the robot movement, defines the cycle time and verificates the reach of the robot. It can be used in many phases of project design, including palletising with industrial robots.

3 DEFINITION OF THE SEGMENT, APPLICATION AND AFFECTING VARIABLES.

Manufacturers of the robotics equipment divide the market into segments, such as foundry, plastic, food and beverage, pharmacy, electronic or automotive industry. [12]. In our case, focus of our analysis will be on one of the market segments and one application, where there is a relatively low number of affecting variables and considerable availability of the simulation software for realisation.

3.1 Definition of the segment and application

We choose to investigate the food and beverage market segment in which three major robotics applications are known: picking, packing and palletising [3]. For the picking applications a dedicated software is always used, because needed connection between the camera, belt conveyor, products on the conveyor and parallel robot [7]. When applications with the robot cycles are in the range of 0.5 s to 1.5 s, it is almost impossible to program the robot without dedicated software. Among the above specified three application types, packaging is probably the least demanding one, because products are inserted into boxes, to be later put on the pallets. The most suitable application for our investigation is palletising. A typical example is taking boxes from manufacture line and stacking them on the pallet according to a given pattern and height specifications. Simplicity of the selected application enables to determine savings when using more complex applications in which the number of the robot targets increases to several thousands.

No additional operations, such as application of glue, laying of carton sheets between layers, application of stickers and weight control will be taken into account. Palletising of different products in different packaging on one pallet will not be analysed, because this demands a completely different approach [13].

3.2 Definition and evaluation of the affecting variables

The time, taken to implement a palletising application is affected by the following factors.

1. *Number of the different pallet positions in the robot cell.* There are two types of palletising. The end of the line and the centralised palletising. In first type palletising of the same product is done on one or more pallets and the second type, is done by using more pallets, each containing different type of the product. Since the case of the same multiple pallets can be easily solved with definition of the new coordinate system and transfer of complete trajectory [11], our focus will be on palletising multiple pallets with each heaving its individual products.
2. *Different sizes of pallets.* EU standard [14] specifies six types of pallets. Among the different pallet sizes, the most often used dimension is 1200x800 mm. The customers wanting to place their pallets directly from the warehouse to their stores bethought transfer of products from pallets to shelves, find the smaller pallets to be most appropriate (such as Dusseldorf 600x800 mm). Different pallets, of course, define different pallet patterns.
3. *Number of different pallet patterns.*Usually, customers specify their own pattern which depends on dimensions of the boxes used, their mass and option to use carton sheets between layers.
4. *Number of different products on individual pallet positions.* As we mentioned above, we will not analyse mixed palletising.
5. *Number of different palletising heights for individual products.* Customers usually demand the pallets to

be of a specific height. As it is quite simple to adjusts the height of a pallet stack or the number of layers on the pallet with a single variable in the robot program, this does not considerably affect the programing time in will therefore not be analysed in our case.

To sum up, the affecting variables which define the number of different assemblies (different robot programs) can be defined as a number of pallet positions in cell K, number of different products on individual pallet I_k and number of different patterns for an individual product on pallet Z_k. As the time for an individual assembly will be defined, the number of assemblies N on an individual pallet position can be defined according to equation (1).

$$N = \sum_{k=1}^{K} I_k \cdot Z_k \qquad (1)$$

4 PROJECT PHASES

In our analysis, the project completion was divided into the following phases:

1. offer phase,
2. execution phase,
3. training phase and
4. production phase.

To define the level of economical justification of using simulation software the first to be defined are the phases of the project in which the software impact is justified and measured.

4.1 *Offer phase*

Simulation made in the offer phase graphically presents the project and gives answers to questions about a proper selection of a robot type (working range, load), peripheral equipment construction tests (grippers, robot stand, in-and out-feeders, etc.), evaluation of the movement time (robot cycle), etc. A general definition of the costs in this phase is not possible, for their depending on too many unmeasurable variables (knowledge of the seller, robot application knowledge of the customer, project scheduling, cost of changing the robot, if the working area or the load of the robot selected prior to simulation is not big enough).

However it can be concluded, that using the simulation software in this phase makes sense, because of the peripheral equipment which is nowdays designed in 3D CAD programs; customers are interested in seeing visualization of their projects before placing their order. Though a general definition of the costs is not possible, the long experience of ABB proves, that using the simulation software positively affects the project development in the quotation phase, for giving the customer a clear idea, of what kind of equipment is being offered and how it will function.

4.2 *Execution phase*

Besides delivery and setup, robots are in this phase also programmed for palletising of above products. Customers usually demand one program for one product and training to be able to add programs for new or some other products in future. For the project also whole programing involves a certain cost. Therefore if the supplier’s programming turns out to be inadequate, his offer might be unsuccessful.

4.3 *Training phase*

Customers are usually trained just before the robot cell is delivered to them. ABB provides five-day courses for palletising applications. The first two days are dedicated to basic work with the robot and the cell and the following three days to programming. If a customer purchases also s simulation software, training is prolonged for three more days for basic knowledge and two more days for training in palletising add-on. The price of the ABB basic training according to their price list [15] is 1.800 EUR. This is also the sum taken into account in our analysis.

4.4 *Production phase*

This phase starts, when a customer has fully accepted the working robot cell and it already produces the products by itself. When a customer decides to produce new products, or to have existing ones changed, he provides himself with new robot palletising programs, or modifies the current ones. As the time spent for modification of the existing robot programs or adding new ones directly affects the production costs, it will be accounted for in our analysis.

5 DEFINITION OF THE PROGRAMMING TIME

Certain amount of the time needed for implementing robotics project takes a mechanical setup of the equipment. As using the simulation software here does not bring any significant impact, our focus will only be on the time, taken for robot programming.

Robot programming can be divided in two phases: definition of the core program and definition of the points in the robot working area. The core program controls the application. It consists of gripper and sensor control, pattern definitions, product selection, sticker position rules, etc. In the second phase the right trajectories for the robot movement are defined.

Let’s now define the time, needed to manually program palletising of one product on one pallet, i.e. time for one assembly. Palletising bethought intermediate carton sheets is done by using different layouts of boxes in individual layers, thous assuring stability of boxes on a pallet. In their past projects [16], it took ABB eight hours to define the core program.

After completing the core program, definition of robot positions, trajectories and testing begins. To enable the very complex grippers to grip multiple boxes at the same time, it takes 32 hours of the programming time on average. This also includes testing of the program and fine tuning of individual points, i.e. trajectories. Since the core program is made only once per application, it takes additional 32 hours to add a new program for a new product.

We must now define the time for programing palletising one product on one pallet by using the simulation software. In the past projects, the time to program the core was reduced to only three hours. The time taken for programing and testing the trajectories was decreased to eight hours. This again includes testing of the program and fine tuning of individual points, i.e. trajectories. Since the core program is made only once per application, it takes additional eight hours to add a new program for a new product.

There are many more affecting variables present in execution phase which affect the project development. Variables like inappropriate functionality of the peripheral equipment and change in the customer's and supplier's demands, always extend the programming time. Though these variables cannot be exactly evaluated, they have to be kept in mind, when interpreting analysis results.

6 PROGRAMMING COSTS

6.1 Execution phase

Costs are affected by the number of hours taken for programming core program T_{core} and trajectories for additional assemblies $N{\cdot}T_{\text{add}}$, price of supplier's programing hour P_{prog}, time spent by programmer the road T_{road} (when programming is not done at the supplier), price of the programmer's hour on road P_{road}, price of basic programming software P_{basic}, price of palletising add-on P_{pall} and price of training P_{tra}. Due to it's small value, the price of daily allowances [17] does not affect calculation, presented in equation (2) it will therefore be disregarded.

$$cost\_execution\_phase\ [\text{EUR}] = P_{\text{prog}} \cdot T_{\text{core}} + \sum_{k=1}^{K} I_k \cdot Z_k \cdot P_{\text{prog}} \cdot T_{\text{add}} + 2 \cdot \left(1 + \sum_{k=1}^{K} I_k \cdot Z_k \cdot \frac{T_{\text{add}}}{8}\right) \cdot P_{\text{road}} + P_{\text{basic}} + P_{\text{pall}} + P_{\text{tra}} \quad (2)$$

For time on road T_{road} we can define two times one hour per day of execution. The price of the time on the road is according to ABB price list 50 EUR and the price of the programming hour is 80 EUR. As in the execution phase the production is not running, the cost of the lost production is not calculated. Based on above prices for manual programming, equation (3) applies.

$$cost\_execution\_manual\ [\text{EUR}] = 640 + N \cdot 2560 + (1 + N \cdot 4) \cdot 100 \quad (3)$$

If the simulation software is used, the cost is the one given in equation (4) and presented in Table (1) and Chart (1).

$$cost\_execution\_simulation\ [\text{EUR}] = 240 + N \cdot 640 + (1 + N) \cdot 100 + P_{\text{basic}} + P_{\text{pall}} + P_{\text{tra}} \quad (4)$$

Table 1: Cost calculation according to the number of assemblies in EUR

Assemblies	Exec. manual	Exec. simulation
1	3.700	4.875
2	6.660	5.615
3	9.620	6.355
4	12.580	7.095
5	15.540	7.835

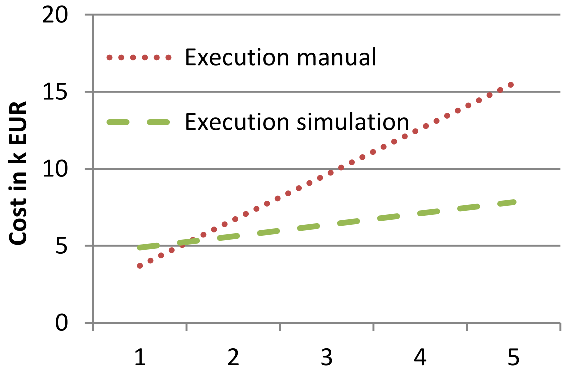

Chart 1: Correlation between the cost and the number of assemblies in the execution phase

For the execution phase, analysis results clearly show that programing with the simulation software is cost efficient already when creating a program for just two assemblies.

6.2 Production phase

For production to be successful it should always reflect the current market demands. There is a great possibility that after a successful start-up of the robot cell, product will have to be either modified, replaced with a new one or additional products will have to be added to the cell. If such be the case, when programming is manual, the production line has to be stopped and a new program created.

In the production phase, the cost of the programming hour varies for being done by the customer himself. So we can define price of the customer's programing hour P_{cprog}, which is usually lower than price of supplier's programming hour P_{prog} and is according to the ABB project documentation on average 50 EUR. At the end, the cost of stopped production P_{ustop} is added and equation (5) defined.

$$cost\_production\_phase\ [\text{EUR}] = P_{\text{cprog}} \cdot T_{\text{core}} + \sum_{k=1}^{K} I_k \cdot Z_k \cdot P_{\text{cprog}} \cdot T_{\text{add}} + 2 \cdot \left(1 + \sum_{k=1}^{K} I_k \cdot Z_k \cdot \frac{T_{\text{add}}}{8}\right) \cdot P_{\text{road}} + P_{\text{basic}} + P_{\text{pall}} + P_{\text{stop}} + P_{\text{tra}} \quad (5)$$

Equation (6) defines the cost of the stopped production as a multiplication of individual products profit P_{i}, number of products in a single package N_{p}, number of packages per hour P_{h} and stop time T_{stop}.

$$cost\_production\_phase\ [\text{EUR}] = P_{\text{cprog}} \cdot T_{\text{core}} + \sum_{k=1}^{K} I_k \cdot Z_k \cdot P_{\text{cprog}} \cdot T_{\text{add}} + 2 \cdot \left(1 + \sum_{k=1}^{K} I_k \cdot Z_k \cdot \frac{T_{\text{add}}}{8}\right) \cdot P_{\text{road}} + P_{\text{basic}} + P_{\text{pall}} + P_{\text{i}} \cdot N_{\text{p}} \cdot P_{\text{h}} \cdot T_{\text{stop}} + P_{\text{tra}} \quad (6)$$

Actual analysis to be made by various companies might be much more extensive and complex to include the actual profit per product, production capacity, number of production shifts and any other cost connected with production (electricity, air, maintenance,). Despite that, their results are still comparable with those of our analysis, presented below. For example, if we palletise beer cans packaged in plateaus with a cycle of 5 s per plateau, we palletise 240 cans per minute, which means 14.400 cans per hour. Assuming, that the beer end price of 1.5 EUR consists of the producer production cost of 0.5 EUR, producer profit of 0.5 EUR and storage profit of 0.5 EUR, the cost of the lost production is 7.200 EUR/hour.

Two possibilities will be studied. In the first case production is run in three shifts and with no possibility of programing wethought production stop. In the second case, production is run only in one shift and the other two can be used for programming.

Entering the assumed cost and simplifying the equation for manual programming with production stop equation (7) applies:

$$cost\_stopped\ production\_manual\ [\text{EUR}] = 400 + N \cdot 1600 + (1 + N \cdot 4) \cdot 100 + 7200 \cdot (8 + N \cdot 32) \quad (7)$$

The cost when using the simulation software is given in equation (8). The results of the two equations are presented in Table (2) and Chart (2).

$$cost\_stopped\ production\_simulation\ [\text{EUR}] = 150 + N \cdot 400 + (1 + N) \cdot 100 + P_{\text{basic}} + P_{\text{pall}} + 7200 \cdot (3 + N \cdot 8) + P_{\text{tra}} \quad (8)$$

Table 2: Cost calculation according to the number of assemblies in EUR

Assemblies	Prod. manual	Prod. simulation
1	290.500	83.745
2	522.900	141.845
3	755.300	199.945
4	987.700	258.045
5	1.220.100	316.145

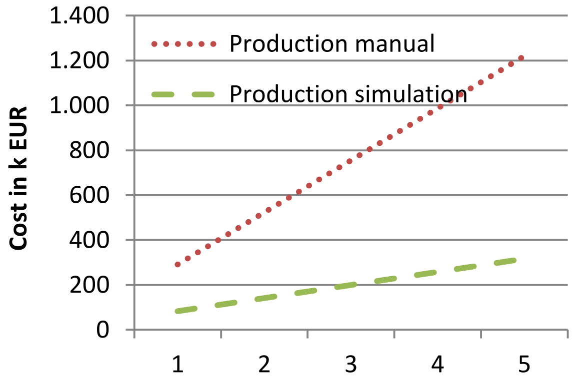

Chart 2: Correlation between the cost and the number of assemblies in production phase

As clearly seen from above, programing in the execution phase is much cheaper than programming in the production phase. This is especially obvious when production has to be stopped, because the cost of the stopped production immediately exceeds all other costs. The conclusion therefore is that using the simulation software in the production phase is strongly recommended.

When there is no need for production to be stopped, the cost of manual programing is calculated according to equation (9).

$$cost\_running\ production\_manual\ [\text{EUR}] = 400 + N \cdot 1600 + (1 + N \cdot 4) \cdot 100 \quad (9)$$

The cost when using the simulation software is calculated according to equation (10). The results of the two equations are presented in Table (3) and Chart (3).

$$cost_{running}production_{simulation[\text{EUR}]} = 150 + N \cdot 400 + (1 + N) \cdot 100 + P_{\text{basic}} + P_{\text{pall}} + P_{\text{tra}} \quad (10)$$

When production needs to be stopped for additional programming, using the simulation software is strongly recommended. In project execution and production where programming can be done in one of the free shifts, using the simulation software pays off after two or three assemblies. To be noted is that the impact of the simulation software in the offer phase where it positively affects the project development and completion is not validated.

Table 3: Cost calculation according to the number of assemblies in EUR

Assemblies	Prod. manual	Prod. simulation
1	2.500	4.545
2	4.500	5.045
3	6.500	5.545
4	8.500	6.045
5	10.500	6.545

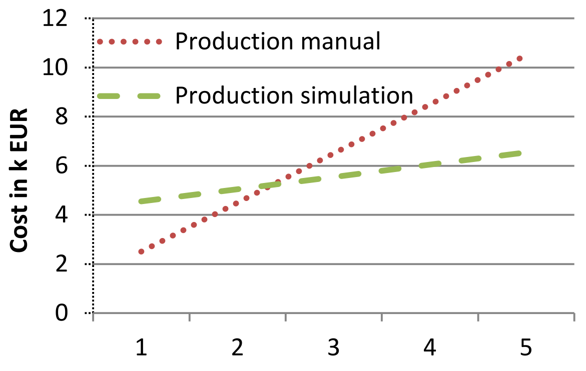

Chart 3: Correlation between the cost and the number of assemblies in the production phase

7 Conclusion

Cost-efficing of using the simulation software when programing industrial robots depends mostly on the type and detailed requirement of an individual application as well as on the software supplier and his ability of using it by the supplier and the end user. A general equation, that would take in to account all the possible variables cannot be defined, as these variables vary from application to application and cannot be accurately or even at all validated.

In our analysis some simplifications are made. From one manufacturer of robotics equipment we chose only one industrial segment and one application. Two project phases were investigated where the impact of using simulation software is the biggest. All the variables defining a number of different assembles were validated all together.

It is shown, that using the simulation software in palletising applications with industrial robots meets the demands of the modern market. An example of a complex application is applying gelcoat on sailboats moulds at ELAN [16]. There programing by using simulation software of more than 2500 robot targets for one trajectory was done fifteen times faster than wethought it. For the suppliers of robotics solutions and end customers to increase their competitiveness on the market, it is strongly advised to increase using the simulation software in their projects involving palletizing with industrial robots. One of the ways of achieving this increase is to present to customers advantages of using the simulation software already in the project quotation phase. The related calculations should be long-term based, and not only for the execution phase.

References

[1] T. Bajd, M. Mihelj, J. Lenarčič, A. Stanovnik, M. Munih, »*Robotics*«, Springer 2009.

[2] Ž. Majdič »*Robotska celica za brušenje izdelkov iz ogljikovih vlaken*«, Univerzitetno diplomsko delo, Ljubljana 2008.

[3] Home page of manufacturer ABB, http://www.abb.com (01.06.2012).

[4] Home page of manufacturer Yaskawa Europe GmbH, http://www.yaskawa.com (01.06.2012).

[5] Home page of manufacturer Kuka roboter GmbH, http://www.kuka-robotics.com (01.06.2012).

[6] Home page of manufacturer Fanuc corporation, http://www.fanuc.co.jp/ (01.06.2012).

[7] J. Lenarčič, T. Bajd, »*Robotski mehanizmi*«, Univerza v Ljubljani, Založba FE in FRI, Ljubljana 2003,

[8] Home page manufacturer CNC software, inc., http://www.mastercam.com/ (01.06.2012)

[9] »*Product specification RobotStudio 3HAC026932-001 Revision E*«, ABB AB Robotics Products, Västeras, June 2012.

[10]»*Product specification RobotStudio Paletizing PowerPack*«, ABB AB Robotics Products, Västeras, June 2012.

[11]»*Technical reference manual RAPID 3HAC16581 Revision M*«, ABB AB Robotics Products, Västeras, June 2012.

[12]IFR statistical department c/o VDMA Robotics + Automation, »*World Robotics Industrial robots 2011*«, Frankfurt, 2011.

[13] Home page of manufacturer AXIUM, http://www.axiumsolutions.com (01.06.2012).

[14] ISO Standard 6780.

[15] Price list of company ABB d.o.o.

[16] Project documentation of company ABB d.o.o.

[17] Uredba o davčni obravnavi povračil stroškov in drugih dohodkov iz delovnega razmerja (Ur. l. RS, št. 140/2006, 76/2008).

Karl Jerman graduated in 2005 from the Faculty of Electrical Engineering, University of Ljubljana. He is currently working as a sales manager for ABB. Most of his time he supports integrators of robotics equipment and investigates robotics solution in the food and beverage industry.

Boštjan Murovec received his B.Sc., M.Sc. and Ph.D. degrees in Electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, in 1996, 1999 and 2002, respectively. He is currently an assistant professor at the Machine Vision Laboratory at the same faculty. His research interests include image processing, pattern recognition, embedded design, combinatorial optimization and sequence analysis.