IOP Conf. Series: Materials Science and Engineering **227** (2017) 012050 doi:10.1088/1757-899X/227/1/012050

Manipulation and handling processes off-line programming and optimization with use of K-Roset

G Gołda[1] and A Kampa[2]
[1,2] Silesian University of Technology, Faculty of Mechanical Engineering, Institute of Engineering Processes Automation and Integrated Manufacturing Systems, Konarskiego 18A, 44-100 Gliwice, Poland

E-mail: grzegorz.golda@polsl.pl

Abstract. Contemporary trends in development of efficient, flexible manufacturing systems require practical implementation of modern "Lean production" concepts for maximizing customer value through minimizing all wastes in manufacturing and logistics processes. Every FMS is built on the basis of automated and robotized production cells. Except flexible CNC machine tools and other equipments, the industrial robots are primary elements of the system. In the studies, authors look for wastes of time and cost in real tasks of robots, during manipulation processes. According to aspiration for optimization of handling and manipulation processes with use of the robots, the application of modern off-line programming methods and computer simulation, is the best solution and it is only way to minimize unnecessary movements and other instructions. The modelling process of robotized production cell and off-line programming of Kawasaki robots in AS-Language will be described. The simulation of robotized workstation will be realized with use of virtual reality software K-Roset. Authors show the process of industrial robot's programs improvement and optimization in terms of minimizing the number of useless manipulator movements and unnecessary instructions. This is realized in order to shorten the time of production cycles. This will also reduce costs of handling, manipulations and technological process.

1. Introduction

Nowadays, during design processes of creation the new automated and robotized flexible manufacturing system (FMS) or reconstruction of existing production and internal logistic systems (PILS) in modern enterprise, the saving of every second and every "cent" is only way to gain the significant position on the competitive market. The same, when the FMS or PILS is working in progress, then every noticed and predictable wastes should be found and eliminated. Approach, like this, is consistent with the "Lean Production" [1].

In the article, we show off-line programming process (using, so-called, block teaching and dedicated AS-Language) and its optimization with computer simulation for single robotized cell with Kawasaki RS005L industrial robot, located in one of the ours, newest Laboratory of Multimotion Control, which is created together with Astor, the leading company of robotic market in Poland (Figure 1). Described in this elaboration studies are conducted in the one of many dedicated or general purpose virtual simulation software: Kawasaki K-Roset.

The readers should note that real technologically advanced system is fully automated and robotized cells or lines like that (often over several dozen), so the savings (time and money) can be considerable in scale of manufactured batches and product series or whole production time (eg. a month, year).

Figure 1. The laboratory of Multimotion Control.
The research lab station with Kawasaki RS005L robot.

Except that, used off-line programming and simulation software K-Roset is only a sample dedicated for Kawasaki. Other software can be useful with any industrial robots of different producers, but perhaps users will have to find required, proper or dedicated software connected with high-level robot programming language. The approach is only proposal to solve problem of minimizing execution time of robot manipulation and handling, which extend time of whole technological process realization. So-called inter-operational brakes generate only interest or transport/storage cost, not added value for the products. The time of inter-operational brakes, connected with wrong, not-optimal (or not-accepted) trajectory, during industrial robots manipulation and handling processes can range up to several or more percent of production cycle time in typical solutions, especially in rapid processes eg. like in sheet-metal stamping in automotive industries [1,2,3].
Nevertheless, the industrial robot and automated system programming using off-line method for different applications is still difficult and time-consuming for programmers and of course expensive, especially in the context of software costs and salaries of experts [4,5].

2. Off-line industrial robot programming: the requirement whether a facilitate for the future

The main question is: Are the off-line industrial robots programming method and computer simulations the only correct way in the future? Of course, these are, in the context of saving programming time, trajectory optimization and minimizing "service" instruction, but skill of on-line (re)programming is indispensable for every programmers, especially that way of off-line programming is usually similar and even identical like in real teach-pendant with dedicated software or programming language. However, nowadays on-line programming is rarely used, just for minor changes in robot trajectory or other basic parameters on working robotized cell and during learning process of a new programmers.
The advantages of off-line programming (OLP) is the fact, that we do not have to use industrial robots in real time and possibility of creating a program libraries before robotized system implementation. During computer simulations operator can check the operation of robot programmes and optimize these without risk of collisions and other troubles. Readers should note, that design of the model, for off-line programming and computer simulations, requires creation of 3D CAD models of whole cell equipments, often with their kinematics (e.g. industrial robots, CNC machine tools, conveyors, warehouses, handling tools, manipulation parts and other so-called obstacles or work parts). The approach is defined as a CAD based off-line robot programming, in brief: "from CAD to path" (Figure 2) [6, 7].

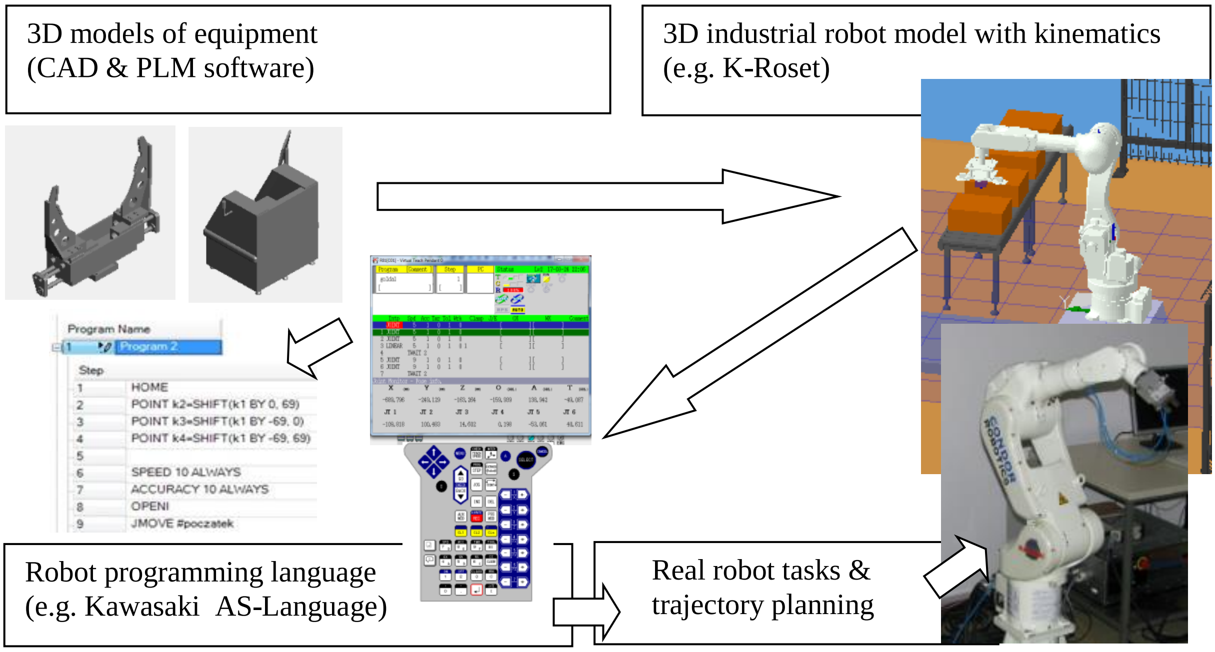

Figure 2. The scheme of CAD based Kawasaki off-line robot programming.

Despite the time consuming during CAD modelling and equipment library creation, the off-line programming method with use of software like: K-Roset (Kawasaki), Roboguide (Fanuc), Robot Studio (ABB), KUKA Sim (Kuka), Process Simulate, Robcad, Delma (Fanuc, ABB and other producers), gives possibility of time saving in the future changes of handling or technological processes operated by robots in automated production cell. Therefore, in this aspect the off-line robot programming and automated processes simulation is a facilitation for the future applications in similar solutions. However, in area of trajectory and robot tasks optimization, it is requirement, because of the ease of making changes in hardware layout or cell geometry and the ability to create alternative solutions of robot programmes in the meaning of accepted paths and trajectories with different operational procedures. This is consistent with the desire to get the only, optimal solution.
Literature [4,5] gives many examples of the theory of "ideal", optimized trajectories, but at the time of "rapid manufacturing" (from a customer order - through modelling, programming and simulation of automated and robotized production - to implementation in reality, manufacturing and delivery the goods to the client), complete optimization can be disastrous because of the lack of time.
Hence, dedicated software, also equipped to detect hypothetical collisions, to avoid them in reality (cell or system downtime and significant costs of damaged equipment repairing). The software meets the needs of users to various ways. The leader, for us, seems the software Siemens Process Simulate (especially for Fanuc and ABB robots with their "collision clouds"), but Kawasaki K-Roset, also has its own solutions like: motion limits for axis, rectangular working space, moving area limits, tool and work part areas.

3. Kawasaki RS005L off-line programming with use of K-Roset software

As mentioned earlier, the first step in design of robotized production cell or whole automated manufacturing system was creation of CAD database of objects representing elementary equipment in real system. The standard *.stl* or internal *.krprj* file format are correct for 3D modelling to use the elements in Kawasaki K-Roset. During database creation we use FreeCAD software and *.stl* format. The next stage was the correct placement of the *Kawasaki RS005L robot* equipped with *handling tool* and other "*obstacles*" (conveyor, forklift, pallet warehouse, loading stations, safety fences) and "*works*" (pallets, parts for manipulation processes). Besides registration of TCP position (using "*entry by hand method*" or "*experimental 4-point method*"), the handling tool and work part, should be described by rectangular tool area, in order to detect hypothetical collisions with scene elements. Only now, trajectory planning will be safety. On the Figure 3, authors show simulation model of robotized palletizing system and present four steps of off-line Kawasaki robot programming with basic and advanced optimization.

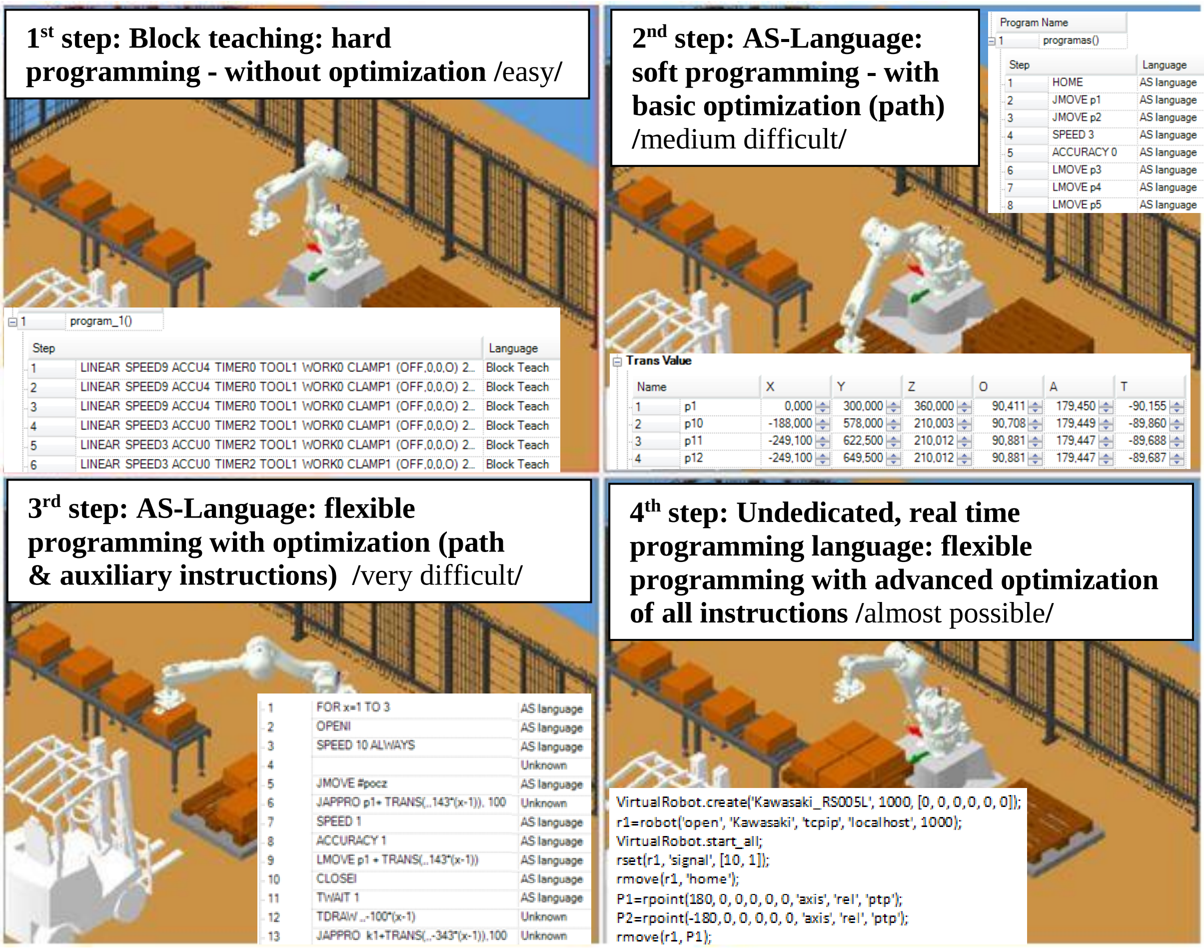

Figure 3. The K-Roset model of palletization with Kawasaki robot. 4 steps of program optimization: Block-teaching, AS-Language programming and optimization of trajectory and auxiliary instructions.

For simple robot tasks (the 1st step of teaching), operators can use "*Block teaching*" off-line programming method by marking destination points of trajectory with these simple motion parameters and elementary auxiliary instructions for handling operations. The programming is easy, but primitive, with many instructions, changes are hard and path optimization is difficult or even impossible. Template (bolded) and sample (italic) of program line is following:

Line No/Interpolation/Speed/Accuracy/Timer/TCP-Tool No/Work No/Clamp-No-State/Out/In Joint Position

9 LINEAR SPEED3 ACCU0 TIMER2 TOOL1 WORK2 CLAMP1(OFF)2(ON) OX=15 WX=1010 #[-26.746,5.6725,-108.14,1.1434,-66.459,26.753] ;

However, use of block teaching method is impossible in case of more complicated trajectories. Then, in 2nd step of optimization (path only), we have dedicated Kawasaki AS-Language which should be connected with *position* and *transformation registers*. They both are created outside the robot program in external terminal (called: the keyboard) or read from the properly designed documentation created in CAD software. The use of position and transformation registers (treated both, more often as a *Cartesian* or rarely as *joint coordinates*) gives possibilities for design the optimal or (at least) an accepted TCP motion path and trajectory. Inside robot program we use *variable & position instruction* like: TOOL, BASE, HERE, POINT, *motion instructions* like: JMOVE, LMOVE, CMOVE, DRIVE, DRAW, SHIFT, TRANS, LAPRO, HOME and *motion auxiliary instructions:* SPEED, ACCELERATION, ACCURACY. Of course, other auxiliary and service AS-Language instructions (OPEN, CLOSE, SIGNAL, TWAIT, SWAIT, RESET) should be used for correct task realization. An example:

Registers outside the program:
#STARTPOINT_P0 [25, 60, -65, 30, 25, -50] - position register in joint coordinates for 6 axes,
CONVEYOR_P1 [-300, 120, 100, 90, -180, 0] - position register in base Cartesian coordinates,
#ROTATION_T1 [45, 0, 0, 0, 0, 90] - transformation register in joint coordinates for 1 and 6 axes,
TRANSFORM_T4 [100, 0, 0, 0, 0, 0] - transformation register in base Cartesian coordinates.
Instructions inside program:
SPEED 50 ALWAYS; ACCURACY 0.1 ALWAYS; JMOVE #STARTPOINT_P0;
POINT P5 = #STARTPOINT_P0 + #ROTATION_T1; JMOVE P5;
LMOVE CONVEYOR_P1; POINT P6 = SHIFT (CONVEYOR_P1 BY 0, 0, 200);
POINT P8 = CONVEYOR_P1 + TRANSFORM_T4; LMOVE P8; CLOSE;

The step described above, give us better and safer TCP paths, by elimination of unnecessary points of robot trajectory, but it does not minimize the number of instructions used inside robot program.
The 3rd step includes the procedure to minimize the number of whole robot instructions not connected with path. For this AS-Language has a set of a *program control instructions*: CALL, GOTO, IF, THEN, ELSE, ON, FOR. Operator can create his own macros and solutions for different application (e.g. palletizing, spot welding, painting and others. A sample for palletizing process:

AS-Language loop procedure
FOR x=1 TO 3;OPENI; SPEED 10 ALWAYS; R=0;
10; JMOVE #pocz; JAPPRO p1+ TRANS(,,143(x-1)), 100; SPEED 1; ACCURACY 1;*
LMOVE p1 + TRANS(,,143(x-1)); CLOSEI; TWAIT 1; ...;*
TDRAW ,,-100(x-1); JAPPRO k1+TRANS(,,-343*(x-1)),100);*
IF R<5 GOTO 10 ELSE CALL HOME; .END

An advantage would be the expansion of the system procedures written in any undedicated programming language (the 4th most difficult step for future researches), assuming it will be understandable to the robot and possibility of program changing in real time on the base of sensors signals. An interesting fact is use of Matlab software as a support for Kawasaki robots programming [8]. Flexible robot programming with advanced optimization in real time is a main aim for every programmer.

4. Summary
In conclusion, it is difficult to find a software that meets every customer's needs. K-Roset is only a sample of advanced, but easy to use and intuitive software for off-line programming, dedicated for Kawasaki robots. Depending on the specific application, we think, that looking for only one, the best, "ideal" trajectory (especially in typical solutions like handling operations or machinery loading) is not justified, because of model creating, programming and optimization time and costs. Of course,

industrial robot trajectory must be safe (for staff, robots, equipments, work parts) and off-line built program should be readable and easy to understand for the operator or programmer. It is necessary for its transformation, for the future use, after the changes of work part or technology. Nevertheless, it should be free from redundant instructions, especially motion instructions - wastes of time and energy and that way - financial losses. The program optimization can be realized in subsequent iterations, understood as next changes in the program libraries, during real system exploitation.

Otherwise, the problem is when we design trajectory connected with technology (welding, machining, rapid prototyping, etc.). Then, all the requirements for motion precision and process parameters, supervised by robot control system and program, must be restricted. So, the problem is more complicated, and optimization should be full.

Summarizing, off-line programming software for industrial robots, independent to used methodology and its given powers, should make easier the design of manipulation processes during any manufacturing technology realization. Created database of work parts, robot tools or equipment, cell or robot scene objects and robot programs should give possibility of rapid reorganization of flexible manufacturing system. It is necessary for realization rapid production of new good, strive to shorten the time from customer order, through start manufacturing, to final product delivery to client.
The specific data, reasonably extracted from dedicated off-line programming and simulation software, e.g. like Kawasaki K-Roset, especially cycle times, are helpful (even, if not necessary) during modeling and computer simulation of integrated, flexible manufacturing system, with use of external software recommended for process planning and control (PPC), as well as, industrial logistics advanced and undedicated modern software (e.g. Enterprise Dynamics, FlexSim or other) [1,3]. Taking into account, the real cycle times of the robot, during manipulation processes in automated cells, will allow to get more realistic results obtained in tests of very complicated micrologistics systems and foreseeing further, a complex (internal or even external) supply chains. This can significantly help the management of the production enterprise and gives possibility of make correct decisions related to accepted customers orders or even investment decisions for future.

5. References

[1] Gołda G, Kampa A and Paprocka I 2016 The application of virtual reality systems as a support of digital manufacturing and logistics *IOP Conference Series: Materials Science and Engineering* **145** 042017

[2] Gołda G, Kampa A and Paprocka I 2016 Simulation model of robotic manufacturing line *Annals of Computer Science and Information Systems* **9** 2300-5963

[3] Gołda G, Kampa A and Paprocka I 2016 Modelling and simulation of manufacturing line improvement *IJCER Int. J. Comput. Eng. Res.* **6 iss. 10** pp 26-31

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

[5] Shiller Z 2015 On-line and off-line trajectory planning *Motion and operation planning of robotic systems. Mechanism and Machine Science* **29** eds. G Carbone and F Gomez-Bravo pp 29-62

[6] Foit K and Ćwikła G 2017 The CAD drawing as a source of data for robot programming purposes - a review *4th International Conference On Computing And Solutions In Manufacturing Engineering - COSME'16 MATEC Web of Conferences* **94**, UNSP 05002

[7] Baizid K, Meddahi A, Yousnadj A, Cukovic S and Chellali R 2016 Industrial robotics platform for simulation design, planning and optimization based on off-line CAD programming *MATEC Web of Conferences* **68**, UNSP 03002

[8] Pawletta T, Freymann B, Deatcu C and Schmidt A 2015 Robotic control & visualization toolbox for Matlab *8th Vienna Conference Mathematical Modelling - MATHMOD 2015* **48** Issue 1 pp 687-688