ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
UDC/UDK 004.896:004.94]:621.791.763

ROBOT PATH OPTIMIZATION FOR SPOT WELDING APPLICATIONS IN AUTOMOTIVE INDUSTRY

Pavol Božek

Preliminary notes
The paper discusses the problem of effective motion planning for industrial robots. The first part deals with current method for off-line path planning. In the second part is presented the work done with one of the simulation systems with automatic trajectory generation and off-line programming capability. A spot welding process is involved. The practical application of this step strongly depends on the method for robot path optimization with high accuracy, i.e. the path transformation into a time and energy optimal robot program for the real world, which is discussed in the third step.

Keywords: *body-in-white, motion optimization, off-line programming, path planning, robotic simulation*

Optimiziranje putanje robota kod točkastog zavarivanja u industriji motornih vozila

Prethodno priopćenje
Rad se bavi problemom učinkovitog planiranja kretanja kod industrijskih robota. U prvom se dijelu razmatraju postojeće metode planiranja off-line putanje. U drugom se dijelu opisuje rad s jednim od simulacijskih sustava s automatskim generiranjem putanje i sposobnosti off-line programiranja. Uključen je postupak točkastog zavariranja. Praktična primjena ovoga uveliko ovisi o metodi za optimiziranje putanje robota s velikom točnosti, za transformiranje putanje u optimalni program robota u odnosu na vrijeme i energiju, prilagođen za stvarni svijet.

Ključne riječi: *body-in-white, off-line programiranje, optimiziranje kretanja, planiranje putanje, simulacija robotom*

1 Introduction

In the past and even today the technology of automotive industry operations is to a certain amount performed by man. The advantage and disadvantage is also that the speed of the work of man depends on his skills. Today spot welding is an important technology used in the production of body-in-white in the automotive industry. With the increase in competition, this technology is increasingly required in automated or robotic form. For its application in practice by current demands computer aided technology must often be used. Modern virtual technologies are examined and applied in various sectors of development and production. Body-in-white consists of more than 300 smaller different shaped parts connected by many welding points (over 4000) [1]. Therefore, there are high demands for precise positioning of welding points, the optimal trajectory planning of robot motion and perfect synchronization of robotic workstations cells.

Off-line programming robots and manufacturing facilities represent a significant technological and time advantage not only in introducing new products, but also in changing the existing production applications. Emulated environment allows programming the robot without stopping production well in advance to prepare robot programs, which increases overall productivity. We are able to create realistic simulations using real robot programs and configuration files identical to those used in production, although currently used simulation tools, such as Robcad, IGRIP or Catia, are unable to solve the problem of optimal robot motion planning [2]. In [10] a case study of a new, cooperative, collision-avoidance method for multiple, nonholonomic robots based on Bernstein–Bézier curves is given. For example the velocities and accelerations of the mobile robots are constrained and the start and the goal velocity are defined for each robot. This means that the proposed method can be used as a subroutine in a huge path-planning problem in real time, in a way to split the whole path into smaller partial paths [10, 11]. Planning spot welding robot trajectories requires much more experience and subjective decision-making, leading to a continued need for on-line correction of robot programs. The speed of the machine work depends not only on its parameters, but mainly on the ability of humans to teach this machine to work quickly.

Model construction of manufacturing equipment, as well as the establishment of relevant robot programs by simulation system represents a true picture of reality. An absolute compliance with reality cannot be assumed. Ideally, it would be to load the robot program without any adaptation. However, there are essential differences between the computer model used to implement graphical simulation and real environment. Deviations may be caused by:

- errors in position of the workpiece and the environment due to the position of the robot,
- errors in tool precision with regard to the robot flange,
- errors in the relative position of robot axes.

For these reasons, correction of the robot path is required, i.e. adapt the simulation to actual geometric conditions. It is an integrated process of modelling, measurements, numerical identification of robots real physical properties with the implementation of a new model [5].

2 Basic principle of path planning

Traditional robot path planning problem can be described as a task in the initial and target configuration q, create in the final time between them a continuous sequence of valid configurations or evaluate an error if no such sequence. Robot configuration is valid if its construction does not intersect any obstacles or itself. The set of all valid configurations defines the topology of continuous motion [3, 4] .

The trajectory planning of industrial robots uses the principle of interpolation in the configuration, or Cartesian space. The simplest and quickest method of interpolation is the point interpolation. Point interpolation guides the robots end effector from one point to another, regardless of the trajectory between goal points. To avoid collision with obstacles between the goal points via points are defined [5, 6].

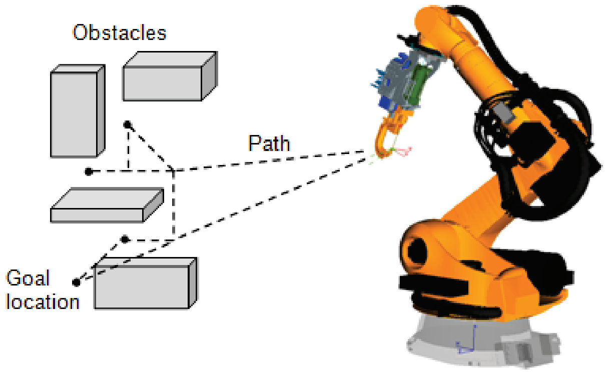

Figure 1 Searching collision-free path between the goal configurations

Most of the off-line motion planners are based on an explicit representation of the free C-space. The free C-space computation consists of the obstacle transformation into the C-space and the construction of a free-space representation [4]. Both tasks are very time and memory consuming, and their calculation effort increases with the robots DOF. In order to avoid these time consuming obstacle transformations, one can search in an implicitly represented C-space and detect collisions in the workspace. This strategy enables the planner to cope with on-line provided environments, moving obstacles and grasped objects. For searching in the implicit C-space, we apply variation of the A*-search algorithm shown in Fig. 1. The main task of the A*-algorithm consists of the expansion and the processing of configurations, which are C-free. In the first step, the algorithm searches for collision free path between goal configurations. The next step is to connect the starting location of the trajectory to the nearest node path between goal configurations. Obstacles are circumvented along the perpendicular vector from actual location toward the next goal location.

3 Methodology of robotic workcell simulation

Robotic workcell simulation is a modelling-based problem solving approach that aims to sufficiently produce credible solutions for a robotic system design [1]. The methodology consists of six steps, as shown in Fig. 2. The aim of most manufacturers is that all programming of robots runs in off-line mode. This allows to make quick changes in production (about tens of hours), to run coming models (product) production and to reach full production with high quality within a few weeks [2]. Programming, as well as simulation can be performed before starting installation work because programs are created offline. Programs can be tested first and after the robots installation is finished all the programs are ready to start production.

Part model is a low-level or geometric entity created using basic elements of solid modelling features of Robcad. These parts consist of the components of the robot and the devices in its workcell.

Device models represent actual workcell components and are categorized as robotic device model and non-robotic device model. The building device model starts by positioning the base of the part model as the base coordinate system. This step defines links with the robot. Each attached link is subjected to a parent – child relationship [1, 8].

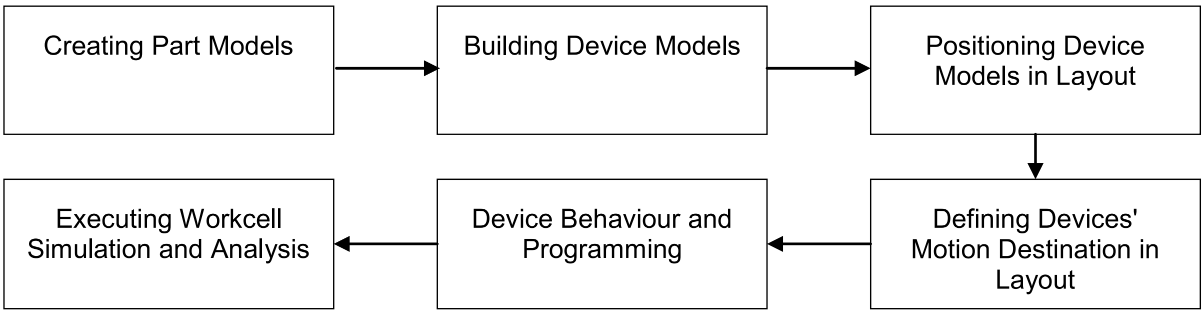

Figure 2 A methodology for robotic modelling [1]

The layout of the workcell model refers to the environment that represents the actual workcell. As in this case, the coordinates system being applied is the Hand Coordinates System of the robot. Placement of the model and devices in the environment is based on the actual layout of the workcell.

The motion attributes of the device model define the motion limits of the joints of the device model in terms of home, position, speed, accelerations and travel. A joint is defined by linking the Link or Rotor of the robot. Each joint has its own motion limits. Once the joints have been defined, Robcad will automatically define the kinematics for the robot.

Device motion refers to the movement of the robot's arm during the spot welding process. The movement is determined by a series of Geometry Points that create a path of motion for the robot to follow.

The simulation focuses only on the position of the robot's arm, not its orientation. After being programmed, the device model layout can be simulated over time. The simulated model is capable of viewing the movement of the robot's arm, layout checking, the robot's reachabilites, cycle time monitoring, and collision and near miss detection [1, 8].

4 Design and analysis of case study

The first prerequisite for solving the trajectory planning of robotic spot welding is the existence of N different target configurations q_i where $i = [1 \dots N]$. These configurations are the solution of inverse kinematics for a set of welding points. Based on the location of welding points we can create different model situations. One such is shown in Fig. 3 below. The next step is to try to create collision-free path between goal points. To move the robot we use PTP interpolation using via points for avoiding collisions.

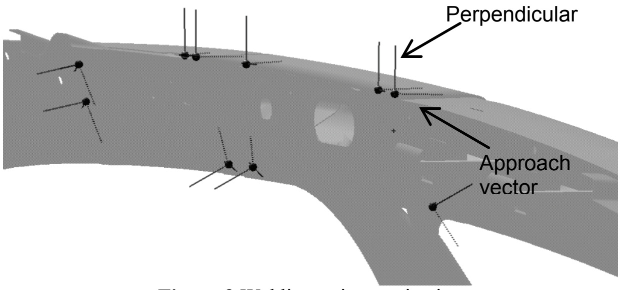

Figure 3 Welding points projection

To circumvent the obstacles we choose the path along approach vector from that point toward the next goal point. If we always connect the nearest goal points with a straight line and all points of the lines belong to the set C-free, we choose the shortest direction around the obstacles. In this case we reach the goal point at the direction of the approach vector in perpendicular angle (Fig. 4). Passing to the next goal point we use the previous via point from which we came to the current destination.

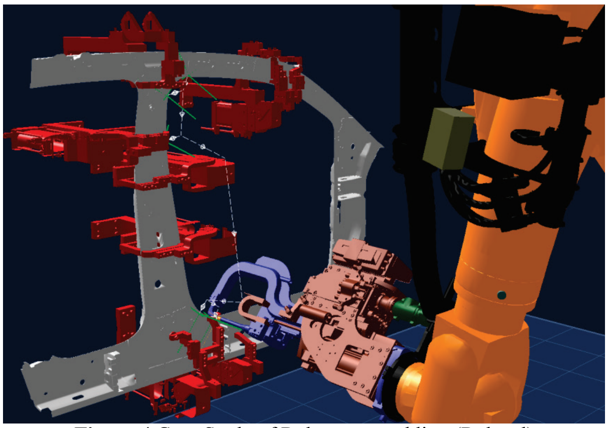

Figure 4 Case Study of Robot spot welding (Robcad)

In practice welding points are almost always concentrated in groups due to technological requirements [5]. Looking at the trajectory under consideration in the previous procedure, we find that it is divided into segments by which we can move between any two goal points. Goal points can be grouped into sets of nearest neighbours and for each set we can test the fastest sequence always from one destination point. This testing can be divided into subsets and so accelerate the process of finding the optimal sequence.

5 Movement optimization

Trajectory planning examined only the kinematics of the robots. In the following optimal velocity profiles must be computed, necessary for a time and energy optimal path execution. Therefore, several robot constraints like maximal joint velocities, maximal joint accelerations, admissible motor- and gear moments, etc. have to be considered. In the body-in-white welding line most of welding or placement operations are implemented by point to point (PTP) movement. At PTP movement all axes start moving in the same time and also stop simultaneously after a time frame that is necessary by the axis that requires the most time to reach the target angle. Since most placement operations expect a specific trajectory only in a matter that any collisions are avoided, there is an open space for trajectory optimization.

A. Point approximation

Most of robot controllers allow to fly-by the programmed locations of TCP within the predefined range and without stopping there. This effect is called point approximation and is used by movement between work locations using the so-called via location. It creates the movement smoother, often shorter and therefore quicker. An example of approximation between two TCP linear movements is shown in Fig. 5.

The actual optimization possibilities are dependent on initial robot's program. The disadvantage at large approximation distance is the deviation from the original path. However, 100 % approximation does not mean that the TCP will always be going to fly-by the via location in the same distance as long as the deviation does not lead to any collisions it is a practical approach to save up to 20 % of movement costs.

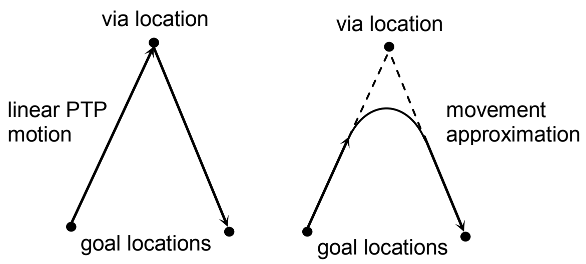

Figure 5 Principle of point approximation

B. Speed and acceleration

Another point of optimization is to evaluate movement profile in correlation between tasks cycle time and the potential energy savings. Running the robot at lower speed, the total consumption decreases, running a robot at constant speed and decreasing the acceleration, the consumption decreases as well.

From Fig. 6 may be concluded that if there is a spare time to do the task slower, it is worth to do so. For example, if it takes 10 seconds to do the pick the component and place somewhere else, and after that the robot is waiting another 5 seconds for some external signal, there is actually 50 % extra time to complete this movement, and by moving slower may be saved 20 % of energy per this cycle time.

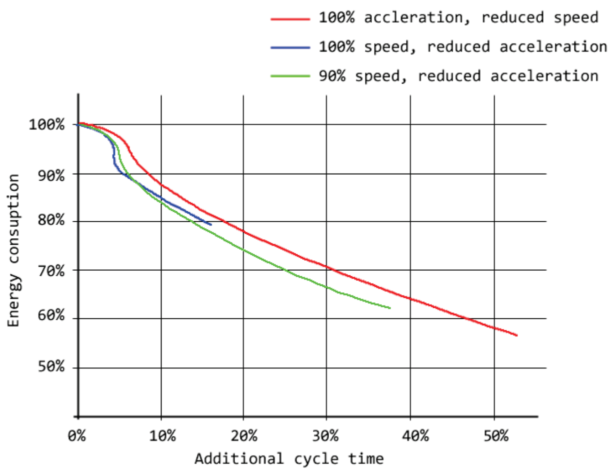

Figure 6 Cycle time and energy saving comparison

6 Model adaptation

To ensure accuracy of the real technological process it is necessary to calibrate off-line programs. Calibration has the most important influence on the acceptability of off-line programming as only in the case the virtual environment can be precisely mapped into the real environment, we can use automatically generated program in practice.

It is necessary to measure the exact position of the workpiece in relation to the robot before calibration. Cell alignment is a method for determining the location of the robot relative to the coordinate system of the workpiece. We can easily determine the TCP position by using calibration tool. For calibration purposes it is sufficient to measure and preset 3 coordinates , of which max. 2 may be located on the same line. The greater is the mutual distance of points, the more accurate will the calibration be. It is important to point out that the 3 coordinates must have an equivalent in the virtual environment, which must be exactly identified.

The main problem by off-line programs implementation is robot tolerances. Since the robot measurement system measures the position of the rotor axis in joints, the total deviation of mechanics is not taken into account. Due to these factors it is good for the accuracy of measurement to choose also the points, where the configuration of joints does not reach extreme position (robot is too "stretched" and it increases the value of deviations). By creating items in a virtual environment, this effect appears to be a real mistake. The tolerance for uploaded via locations is ± 1 mm and ±0,2°. If there are deviations existing among the nominal locations and the uploaded program, there may occur different cases which require different solutions [7]:

1) All uploaded locations are ok except via locations which are not in tolerance; Solution: Update of via locations in the cell.
2) All the uploaded locations have a constant deviation; Solution: Check the accuracy of robot in relation to mechanical installation requirements for functional packages and related objects. In case the solution is not found or it is time consuming and expensive, we can use calibration.
3) Random deviations of uploaded locations; Solution: In case that no other solution (similar to point 2) is found, there may be a geometric inaccuracy of the tool, or fixture. NB: nominal working locations may never be treated in simulation model.

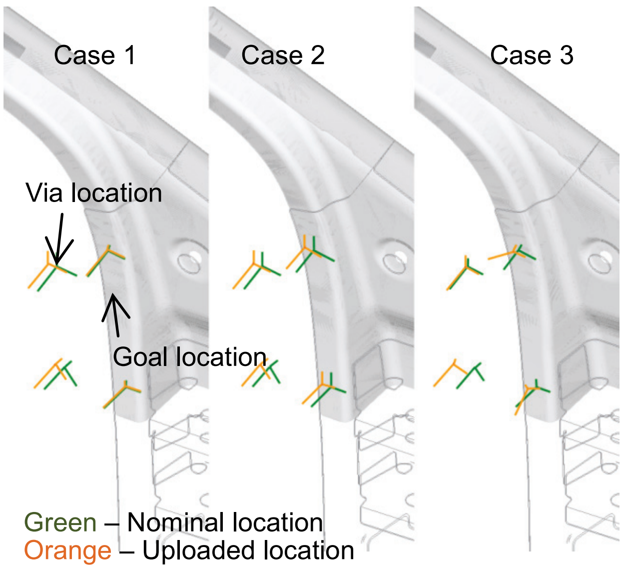

Figure 7 Comparison of deviations

Generally we distinguish among the 3 calibration methods [7]:

1. Device calibration;
   a) Robot is calibrated relative to the position of the fixture where the number of robots is associated to one fixture.
   b) Fixture is calibrated in case the number of fixtures is associated to one robot.
2. Tool calibration; by production conditional deviations of tool from nominal values.
3. Robot calibration by mathematical correction of positional error, that may occur by producing contingent dimensional and angular deviations of the robot axes.

By off-line programs implementation the device or workpiece calibration is mostly used. As mentioned above, since the off-line programs are created in workpiece coordinate system, the calibration is triggered by a linear transformation of calibration pairs to put the origin of the workpiece coordinate system. In fact, its position does not change either in simulation or in real world. New values of its origin just offset mismatch between robot working off-line locations toward reality. The obtained value replaces the nominal value set by simulation. The advantage of the method described is that there is no change in coordinates of working locations, but the whole coordinate system is moved in order to obtain the same values of robot joints rotation after the program is uploaded as by the simulation. At the same time it is needed to keep the nominal value of working point coordinates of a particular technology.

7 Experimental results

In the automotive industry more than 95 % work in the body shop is done by robotics-related applications. The measurements show a logical trend in the relation of the energy savings and cycle time loss. Because of the high degree of automation and cyclically reparative behaviour of robots, even little improvements in the efficiency of their systems may result in significant time and energy reduction in whole production. Combining the described usage approaches like appropriate optimal trajectory search algorithm choice, trajectory optimization methods, total energy savings can exceed 40 % according to reference programs without the use of these modifications.

Our proposal is based on the Robcad simulation model and therefore we do not present a proposal of real robotic workcell. Because of the fact that the actual savings are hardly application dependent and one method may increase or decrease the influence of others, only the statistic averages are estimated. Future research work will involve studying the optimization effect of several planning algorithms.

8 Conclusion

Optimal motion planning of robots using different methods and algorithms is getting great attention of planning theory, however due to the computational complexity many solutions are rarely used in practice. Often solution accuracy depends more on the actual experience of the programmer. In the case of robotic spot welding the aim is to achieve a reasonable compromise between finding a long path that can be executed faster and a short trajectory, which requires movement at lower speed. By using one algorithm it is obviously impossible to achieve the desired result and combination of several methods can be time and hardware consuming. Based on the requirements of the car body welding line, we can say that it is necessary to deal with so much precision solutions to achieve a sufficiently rapid trajectory in a relatively short time.

Result of the model adaptation may not always be a new program upload. In many cases it is sufficient to move the coordinate system of the robot working area. An important part of the methodology is an accurate kinematic model with easily identifiable characteristics. This is secured by off-line programming primarily in the workpiece coordinate system. Calibration may still be a long-term process and its outcome is not always accurate. Mostly it is caused by measurement errors and wrong setup of tool centre point. Further errors may occur by the introduction of the nominal load during production. This may be due to the excessive load of robot during the measurement. Therefore, it is preferable to carry out measurements by real-load robot, which requires the use of different technologies (laser measurement, optical sensors, etc.). It is suitable to deal with a different implementation of calibration, which is often time consuming and thus suppressing the primary benefits of off-line programming. One option would be to implement the calibration functionality directly into the robot controller, while the actual standard of measurement, nominal values measured locations in the virtual environment, can be uploaded first. It would be possible to evaluate calibration results without downloading calibration path immediately after the measurements. This would significantly speed up the calibration process, which is often limited by the absence of robot offline programmer directly by on-site installation.

9 References

[1] Wang, Z. et al. Off-line programming of Spot-Weld Robot for Car-Body in White Based on Robcad, International Conference on Mechatronics and Automation, China (2007).

[2] Dillmann, R. et al. Robotik in Deutschland - Lehre, Forschung und Entwicklung, Shaker (1998), pp. 9-1. ISBN 3826531868.

[3] Pires, J. N. Industrial Robots Programming, Building Applications for the Factories of the Future, Springer (2007). ISBN 0387233253

[4] Lavalle, S. M. Planning Algorithms, Cambridge University Press (2006), viewed January 6 2012, http://planning.cs.uiuc.edu/.

[5] Leonardo da Vinci Programme, Automatizační a robotická technika, SjF Košice (2011), viewed January 15 2012, http://www.sjf.tuke.sk/vitralab/files/vystupy/prirucka.

[6] Goga, V.; Kľúčik, M. Optimization of Vehicle Suspension Parameters with use of Evolutionary Computation.: Modelling of Mechanical and Mechatronics Systems. 5th International Conference. Zemplínska Šírava, Slovakia, ISSN 1877-7058.

[7] Jambak, M. et al. Robotic Modeling and Simulation: Theory and Application. InTech, 2010. ISBN 9789533070735. viewed October 13 2012, http://www.intechopen.com/articles/show/title/robotic-modelling-and-simulation-theory-and-application.

[8] Moore, A. W. Robot Motion Planning. School of Computer Science, Carnegie Mellon University, 2007. viewed January 9 2012, http://www.autonlab.org/ tutorials/motion07.pdf.

[9] Cheng, F. S. A methodology for developing robotic workcell simulation models. In Winter Simulation Conference, Washington, 2000.

[10] Klančar, G.; Škrjanc, I. A case study of the collision-avoidance problem based on Bernstein-Bézier path tracking for multiple robots with known constraints. // J. intell. robot. syst. 60, 2(2010), pp. 317-337.

[11] Škrjanc, I.; Klančar, G. Optimal cooperative collision avoidance between multiple robots based on Bernstein-Bézier curves. // Robot. auton. syst. [Print ed.]. 58, 1(2010), pp. 1-9.

Author's addresses

Pavol Božek, Assoc. Prof.
Institute of Applied Informatics, Automation and Mathematics
Faculty of Material Science and Technology STU
Hajdóczyho, 917 24 Trnava, Slovakia
Tel.: 00421 903 240 686, E-mail: pavol.bozek@stuba.sk