# A RATIONAL B-SPLINE CURVES IN ROBOT COLLISION - FREE MOVEMENT PLANNING

Received 17th September 2007; accepted 7th April 2008

*Daniel Reclik, Gabriel G. Kost*


## Abstract:

*In this paper there is described the using of 2D robot workspace analysis method with smoothing process by using B-Spline curves [7]. The robot workspace analysis is necessary to generate the collision-free movement paths. This paper is connected with a project, which will be ended with fully functional added application in robot off-line programming. In the last part of this project the authors created the motion planner based on grading function and Markov chains. This planner allows determining the collision-free trajectory of robot bunch. Determining the collision-free trajectory generates a lot of step functions – all movements were possible in only one direction (parallel to Cartesian axes) [1,7]. The worked out planner choose the best path – by shortest movement length or shortest movement time criterion. Achieved path is optimised and smooth by using NURBS and B-Spline curves [2,4,5], which is the main goal of this paper. Smoothing and time-function making is very important, because it allows getting the speed and acceleration data, which are necessary in robot controlling. B-Spline curves fulfil continuous first and second derivate condition. Another advantage of using B-Spline is a possibility to enlarge the movement quality by using the minimal acceleration criteria. This allows growing the steadiness of movement.*


***Keywords:*** *robots' motion planning, non-collision trajectory, NURBS and B-Spline curves.*

## 1. Introduction

Defining the robots' trajectory is always based on the environment, where they are working in. It is so because in the robot's technological environment there are spaces and technological objects, which are a part of structure of manufacturing system [6]. The proper settings the optimal way of robot's transition between determined indirect positions in the task space, is basic quality of the robot programme. In case of concentration the robot task space with the technological objects roadblocks, the criterion of optimal robot's passage from start do final position is very important. It is so because it is necessary to perform the requirement of collision-free movement. That's why there must be used many different by-pass tracks, which often eliminate the geometrical shortest way. These two basic parameters of robot's movement, collision-free track length and time of its realization are very important in optimisation the robot's programme [1,4,6,7].

## 2. The collision-free robot movement planning

The general integration in manufacturing systems is realized on informative plane. Computer systems of off-line robot programming are not technological advanced. In these systems there is very weakly developed area for determining the collision-free robot passage between the objects in its task space. The problem of analysis the planned movement comes from difficulty of solution such key matters how [1,4,7]:

- the analysis of three-dimensional robot task space (with roadblocks arrangement),
- the way of interpretation the kinematical robot structure with movement analysis (kinematics chain, manipulated object, etc.),
- the opinion of way of shifting the robot in order to determine the optimal collision-free passage.

Because of complication of geometry level of the robot workspace/For the reason of complication level of geometry the robot workspace, its collision-free movement must be calculate in parts. Particular stages are determined between following point, which can make a collision [1,6,7,8]. From that reason, there was necessary to use iteration treatment of computer added collision-free trajectory planning. Special made agent of planning calculates the robot collision-free movement. Its tasks can be divided into following groups [1,4,6,7]:

- to analyse robot workspace for finding area where robot movement is safety,
- to determine in robot workspace its collision - free movement curve - the set of space points which guarantee the safety movement,
- to determine the collision-free trajectory, as time function of XY robot positions in workspace.

The first task of planning agent is geometrical analysis the robot workspace with collaborate objects configuration (coordinates and dimensions) support. The performed analysis allows agent to defining points reached in next run. Those points are the base of robot safe movement trajectory. Basis points can be typed only in area without collaborated objects (free space). The way of calculation implicated aspect of discovered trajectory. It is a stair-case function curve, created by step-by-step movement positions of robot arm. The positions are lin-ked with typed in free area basis points of trajectory (Fig. 1).

Obtained collision - free path with staircase form cannot be used to realize the robot work algorithm. It is so, because this path isn't perpetual in geometrical way [1,4,6,7,8]. The test of using calculated path to move the robot has been failure, because the motion along this path has permanently change direction (Fig. 1). From that reason, there is necessary to make smoother the determined path. That process is realized by eliminate those points from set of basis points, which don't change the trajectory direction. The task of finding the new set of basis points, which is a subset of those points determined in the beginning, is the next step in collision-free real robot trajectory calculation [1,8].

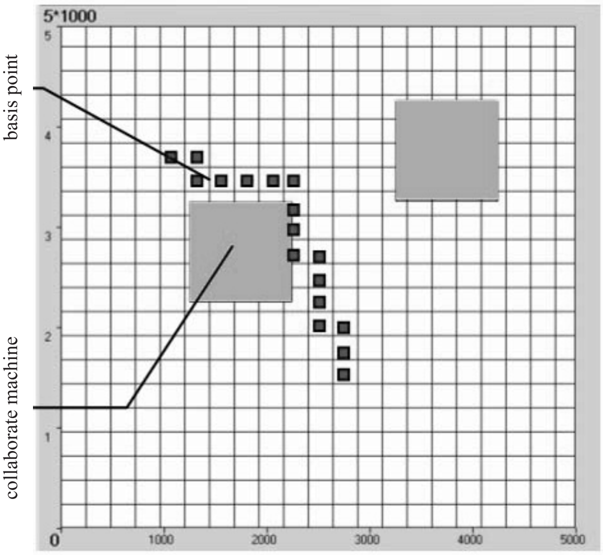


*Fig. 1. The result of the robot arm dynamic collision-free movement planning software work.*

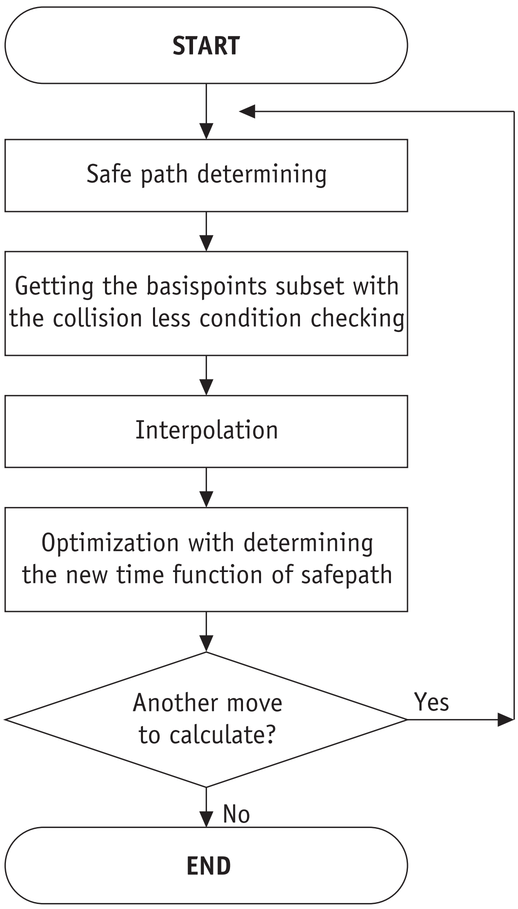


*Fig. 2. Algorithm of the robot arm dynamic collision-free movement planning software.*

After ending that process, there is possible to make a time function of discovered, new movement path (Fig. 2). That guarantees receiving equation with continuous second derivative (curve in $C^2$ class) [1], which is a condition to make possible in robotics usage. It is very important that the first derivate of position time based function is velocity and that the second derivate is acceleration of the robot arm.

## 3. The collision-free curve equation in flat space

Techniques of robot trajectory determining, which are used in robotics, are based on searching new basis points. It is very important, that the new basis points' set must allow to create the robot movement trajectory curve by interpolation way. It always generates the new shape, which is saddle with error relate to the safe path. One of the possible solutions to minimalize the error is enlargement of the basis points number, but unfortunately it takes a lot of computing time [1,4,5,8]. Interpolating techniques, used to solve that task, are mainly based on polynomial curves. It takes time, so that most often there are used low degree polynomials, but it guarantees only receiving equation with continuous first derivative at least or, more often, geometrical smooth shape. For that reason, except from polynomial interpolation, there is used Bézier's curves (Bernstein-Bézier's polynomial) and B-Spline function in robotics [2,3,4,5] and its rational forms [2, 3]. The rational form allows setting the weight of interpolation basis points.

### 3.1. Bézier curves

Interpolation methods, which are based on Berstein-Bézier's polynomial, have many disadvantages. The most troublesome are [2,3,4,5,9,10]:

- adding a new basis point of interpolation implicates the necessity of all earlier computing repeats,
- problems with interpolation that shapes as circle, ellipse which are typical for robot trajectory,
- lack of possibility of basis points weight individual modification to get better compatibility with collision - free trajectory.

Taking into consideration above disadvantages, there is suggested to use B-Spline functions, which fulfilled the robotics conditions. Its most important advantages are [2,3,4,5,9,10]:

- possibility to modify the part of curve, without necessity of all earlier computing repeat (there is only repeated the shorted part of curve near by modified),
- point - length of recomputed part is connected with the polynomial degree,
- for the 3rd and higher level of B-Spline degree, there is guaranteed getting the continuous second derivative in interpolation basis points - splining point.

In case of the necessity of changing the interpolated curve from B-Spline to Bézier's curve, there is possible to use the same algorithm. It is so because the maximum level of B-Spline must be smaller than the value of basis points items. For the level of B-Spline equal to number of those points received form of mathematical B-Spline form equation is an identity to Bézier's curve equation [9,10]. So that, changing the interpolation curve is equal with changing the level of B-Spline equation to number of interpolation basis points value [9,10].

### 3.2. NURBS curves (rational B-Spline polynomial)

Using the interpolating procedure, which is based on NURBS curves, is much larger in case of using the conventional B-Spline. It happens so, because every basis point of interpolation has its importance, which shows the level of matching the interpolated curve to the point. Moreover, the NURBS analytical algorithm is very universal, because attributing the equal importance for all points of collision-free trajectory causes automatically change of the track equation of robot to B-Spline (or if the B-Spline degree is equal to the number of basis points, changing the track equation of robot is to Berstein-Bézier polynomials).

## 4. The algorithm of generation the interpolated curves

Interpolated curves, based on Berstein-Bézier polynomials, NURBS curves and B-Spline, are composite functions. The base of generating those curves is basis function, which suitable linking up gives the proper shape of 3D curve with set parameters. Because basic functions are defined by the time, therefore, one can find that, the curve, which comes from the link, is also indirectly defined by the time. Therefore, to determine the interpolated curve of robot movement on collision-free trajectory, at first, there must be provided the total time of movement duration. This time (of movement duration) is indicated as base unit, it means that the movement lasts from 0 to 1 in time interval, that is from 0 to 100% of real time. Because the NURBS mathematical algorithm is very universal, therefore there is described the procedure of realization that algorithm [2, 7]. It is necessary to determine the kinematics pair at first during creating the polynomial of flexible interpolated function. Because the movement on curve must last from 0 to 1, therefore the interval time from that range is interpolated kinematics pair. To accept the right number of interpolated kinematics pair, there must be defined the degree of interpolated polynomial $n$ and number of ordinary points of interpolation $V$. Kinematics pair set can be passed mathematically in following way [2,3,10]:

$$K=\{t_0,t_1,t_2,t_3,t_4,t_5,\ldots,\ t_{u-1}\} \tag{4.1}$$

However, the number u of kinematics pair must be equal:

$$u=V+n+1 \tag{4.2}$$

To assign the right value of time from 0 to 1 there must be used the following procedure: the first $n+1$ of kinematics pair must be fulfilled by $0$ and the last $n+1$ must be fulfilled by $1$, however the remaining kinematics pairs must be fulfilled proportionally by ascending time values. The set of interpolated kinematics pairs for curve of third degree and for seven basis points has been shown below [7]:

$$K=\{\,0,0,0,0,0.25,0.5,0.75,1,1,1,1\,\} \tag{4.3}$$

After having defined the kinematics pairs of interpolation ($K$ points on time axis – formula 4.3), it is possible to determine the basic function component of rational B-Spline polynomial (NURBS). Defining the basic functions must be done iterative, because if one wants to have determined basic function of any degree, at first one must defined the basic function of lower degree. That's why the whole process of defining the basic functions starts from determining the basic function of 0 degree, which is defining as follows: for set time interval are 1, while for others time interval are 0, as it shown below [2,3,4,5,7,9,10]:

$$\chi_{i,0}=\begin{cases}1 & \text{for moment } t \text{ which belong to time interval among } t_i \text{ and } t_{i+1}\\ 0, & \text{for other time } t\end{cases} \tag{4.4}$$

Basic functions of higher degrees are defined by using the following formula [7,10]:

$$\chi_{i,p}(t)=\frac{t-t_i}{t_{i+p}-t_i}\cdot\chi_{\mathrm{i,\,p-1}}(t)+\frac{t_{i+p+1}-t}{t_{i+p+1}-t_{i+1}}\cdot\chi_{i+1,p-1}(t) \tag{4.5}$$

where: $p$ – is in turn: 1, 2, 3... etc. until to value of polynomial degree $n$.

Having computed values of basic functions, there can be determined the points' coordinates of interpolated polynomial $X(t)$, $Y(t)$ as superposition of values of particular Cartesian co-ordinate system. In case of conventional B-Spline polynomial, the values of coordinates are determined as product sum of basic functions and coordinate values of basis points $P$, in a way as it is shown below [9,7]:

$$X(t),Y(t)=\sum_j \chi_{j,n}(t)\cdot P(x,y)_j \tag{4.6}$$

For determine the rational B-Spline polynomial (NURBS) one must take into consideration the importance of the basic points $P$, which are indicated as $w$. Knowing the importance of basic points one can determine particular coordinates of interpolated polynomial by using below formulas [9,7]:

$$X(t),Y(t)=\frac{\sum_j \chi_{j,n}(t)\cdot P(x,y)_j\cdot w_j}{\sum_j \chi_{j,n}(t)\cdot w_j} \tag{4.7}$$

In order to display how the algorithm works, there are shown, in example, many different trajectories, which have random chosen basic points. The main property of generated curves is degree matching to basis points. If there is taking the higher degree of interpolated polynomial $n$, the generated curve is more standing-off from basic points. Enlarging the degree of interpolated polynomial causes reduction of curve's inflexion, what finally makes that the curve has soft rounding.

## 5. B-Spline algorithms in robot collision - free movement planning

The main goal of this paper was presentation of the application of interpolated curves based on NURBS polynomial for smoothing the safety trajectory of robot movement. The initial trajectories are represented by the set of basic points of trajectory of the robot movement. Received by the authors the procedure of generating the initial trajectory implicates the broken line, determined by the effected basic points $P_i$ [1,6,8]. The broken line (robot's path [6,8]) is always parallel to one of Cartesian axis of co-ordinate system, what unfortunately causes that one can't use it in direct way for programming the robot movement. The length between the $P_i$ and $P_{i+1}$ basis points ($l_i = |P_i P_{i+1}|$) is the most important criterion (geometrical criterion [1,6,8]) in choosing the shortest step path connected the starting ($P_0$) and ending ($P_k$) points of collision-free robot movement path. So the most important criterion is [6,8]:

$$L_{P_{OPT}} = \min_k \left( \sum_{i=1}^{k} l_i \right) \tag{4.8}$$

where: $k$ – number of items in set of basis points is the shortest step path.

The $P_i$ is set of basis points, which creates the shortest curve $L_{P_{OPT}}$ is basic in determining interpolating curves based on NURBS algorithm. In Fig. 3 there is shown the robot workspace with machine tools.

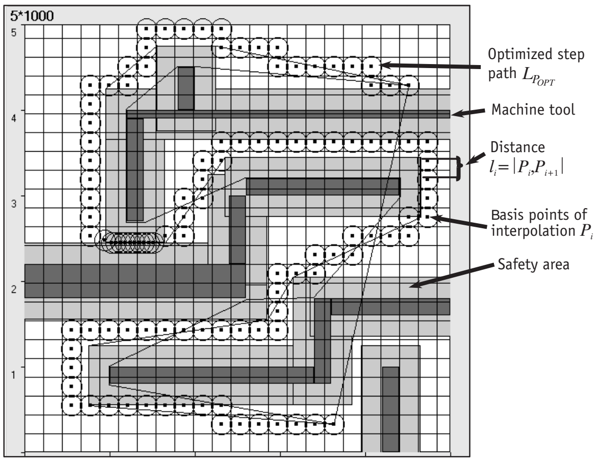


*Fig. 3. Top view of the robot workspace with the discovered basic safe trajectory.*

The machine tools (in Fig. 3 shown as dark blue blocks) are a part of technical infrastructure of robot-integrated system, however they are also a kind of road-blocks for free robot movement. The worked out agent of trajectory can determine the safety trajectory, based on mathematical recording of robot workspace and overall dimensions of manipulated object. Assuming that the robot movements are collision-free, the program - robot's movement planner - assigns for every machine tool the danger zone, which is passing by during generation the trajectory (in Fig. 3. they are shown as light blue blocks). Basis on overall dimensions of manipulated object there is computed the grid size and diameter of indirect position circle. In Fig. 3 there is shown the generated safety trajectory of passage from one to another machine tool, preserving the collision-free condition. The researches shown that there is necessary to use the additional module of created program which is an algorithm of selective, multiple choice of basic points of initial trajectory. This module displaces automatically corners from the basis points set of initial trajectory with checking each time the collision-free condition of new made trajectory. Applied mathematical algorithm is programmed to minimized number of basis points [1,6,7,8]. The end of work of additional module causes beginning the maximization process of degree of interpolated polynomial. It also gives a possibility to set the importance for others basis points of initial trajectory. In Fig. 4 there are shown stages of smoothing the trajectory of collision-free robot movement in workspace. The next process stages of optimisation the safety robot trajectory in workspace shows how smooth the initial trajectory (in Fig. 4 shown as blue block) can be by using presented algorithm.

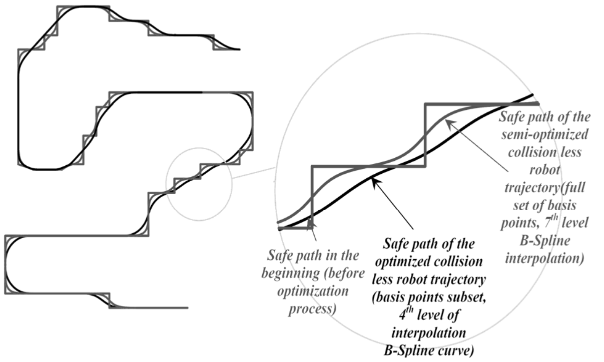


*Fig. 4. Stages of smoothing the trajectory of collision-free robot movement in work space.*

The final trajectory (stated by black colour) is created after selective displacing the 21 corner points. The final level of polynomial is determined on 4 ($n=4$); the continuous condition of second derivative was having the degree of interpolated polynomial larger than two). The third, indirect trajectory is stated in Fig. 4 by red colour and it shows how important the selective displacing is. This trajectory is made from all basic points and maximally smoothed interpolated polynomial (Bézier curve – the polynomial degree is equal to the number of basis points of initial trajectory) [2,3,9,10]. Unfortunately, although there was used the interpolated polynomial of the highest degree, there have been appeared oscillation round the optimal trajectory. In this project we are going to make an optimally and smoothest trajectory ($C^2$, see 2). To make it possible there are used the movement time and acceleration criteria, which is determined as follow [1,6,7,8]:

$$V_{OPT} = \frac{dL_{P_{OPT}}}{dt} \tag{4.9}$$

and

$$a_{OPT} = \frac{d^2 L_{P_{OPT}}}{dt^2} \tag{4.10}$$

After ending the process of the safety trajectory of robot movement in workspace, there is possible to show the algorithm results against an initial trajectory and robot workspace backgrounds (Fig. 5). In upper part of the Fig. 5 there is a trajectory, which crosses the safety area, round the machine tool and it was accepted by the system. It happens so because at first there was necessary to use technological objects with parallel edge (also with safety area of these objects).

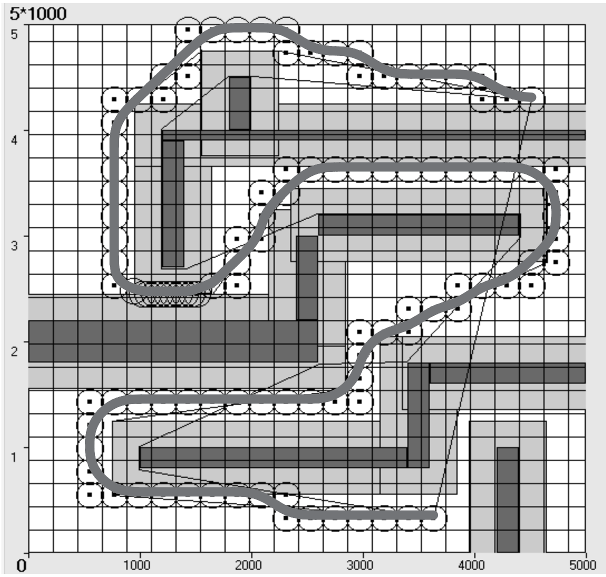


*Fig. 5. Optimal and safety trajectory in the robot work space.*

It was necessary to limit the time of scanning the task space, which was considerably shorter for rectangular corners [8,6,10]. To reach the cube outside profile by the spherical object there is used compensation with offset value equal to the diameter of manipulated object external smallest globe. Such a form has also the robot arm safety movement space. The algorithm, which checks the correctness of created trajectory, is using the real shape of safety areas. Because the checking module has defined the iterative control algorithm of work path (the actual smooth degree), it doesn't have to search it in 3D space. Therefore, having an algorithm, which can check the real safety area, doesn't cause the extension of computed time (it consists in checking additional logical conditions with stable parameters – positions and machines' dimensions) [1,6,7,8].

## 6. Conclusion

The worked out task is a part of collision-free movement planning process. This method allows determining the curve of collision-free trajectory in robot workspace. Using the geometrically optimised path curves that the received NURBS curve is also optimal in geometrical sense. Thanks to this the determined coordinates of robot trajectory can be used in off-line robots' programming system. The system of searching collision-free robot trajectories, which is presented in this paper, has not been fully finished yet. This system is constantly developed, especially there has been working on mechanism of 3D optimal searching the equation of robot's movement in its workspace. In effect, the authors will prepare the fully functional system, based on B-Spline, which can be added to one of off-line programming system of robot.

**AUTHORS**
**Daniel Reclik*** - Ph.D. student at the Silesian Technical University, Institute of Engineering Processes Automation and Integrated Manufacturing Systems, ul. Konarskiego 18A, 44-100 Gliwice, POLAND. E-mail: daniel.reclik@polsl.pl
**Gabriel G. Kost** - Professor at the Institute of Engineering Processes Automation and Integrated Manufacturing Systems of the Silesian Technical University, ul. Konarskiego 18A, 44-100 Gliwice, POLAND.
* Corresponding author

## References

[1] Latombe J.-C., *Robot motion planning*, Kluwer Academic Publishers, Boston-London 1993.
[2] Rosen K.H., *Discrete Mathematics and Its Applications*, 2nd edition, McGraw-Hill Publishing, New Yersey 1991.
[3] Piegl L.,Tiller W., *The NURBS book*, Springer-Verlag Publishing, Berlin-Heidelberg 2005.
[4] de Berg M., van Kreveld M., Overmars M., Schwarzkopf O., *Computational geometry algorithms and applications,* Springer-Verlag Publishing, Berlin-Heidelberg 2000.
[5] Predarata F.P., Shamos M.T., *Computational geometry an introduction*, Springer-Verlag Publishing, New York 1985.
[6] Kost G., Zdanowicz R., "Modeling of manufacturing systems and robot motions", *Journal of Materials Processing Technology*, Elsevier, vol. 164-165, May 2005, pp. 1369-1378.
[7] Boissonat J.D., Budrick J., Goldberg K., *Algorithmic Foundations of Robotics V*, *Springer tracts in advanced robotics 7*, Springer-Verlag Publishing, Berlin-Heidelberg 2004.
[8] Kost G.G., *The based on grading function and Markov Chains stationary and manipulation robots safety movement planning*, Wydawnictwo Politechniki Śląskiej, Gliwice 2004 (in Polish).
[9] Fortuna Z., Macukow B., Wąsowski J., *The numerical methods*, PWN, Warszawa 1999 (in Polish).
[10] Majchrzak E., Mochnacki B., *The numerical methods. Theoretical foundations, practical aspects and algo rithms*, Wydawnictwo Politechniki Śląskiej, Gliwice 2004 (in Polish).