Agronomy Research **14**(3), 862–872, 2016

Optimization of vehicles' trajectories by means of interpolation and approximation methods

D. Novák[1,*], J. Pavlovkin[1], J. Volf[2] and V. Novák[2]

[1]Matej Bel University, Faculty of Natural Sciences, Department of Technology, Tajovského 40, SK 974 01 Banská Bystrica, Slovakia
[2]Czech University of Life Sciences Prague, Faculty of Engineering, Department of Electrical Engineering and Automation, Kamýcká 129, CZ 16521 Prague 6, Czech Republic
[*]Correspondence: daniel.novak@umb.sk

Abstract. The need to optimize the trajectory of vehicles is still highly topical, regardless weather the means of transport are robots, forklifts or road vehicles. It is not only important the safety by passing obstacles, but also the energy balance, i.e. the energy expended on the movement of the vehicle and on the change of its direction. This paper presents a mathematical approach to solving this problem through interpolation and approximation curves.

Key words: means of transport, trajectory optimization, interpolation curves, approximation curves.

INTRODUCTION

Movement of vehicles only rarely proceeds in a straight line. On the contrary – regardless weather transporting material or people into smaller or larger distances, it is almost always necessary to deal with obstacles on the path. This includes both safe avoiding obstacles and selecting the best possible trajectory from several possible options. Choosing the optimal trajectory makes thus the movement safer, may reduce the transportation costs and last but not least it may also save time.

Mathematically it is possible to perform an interpolation or an approximation of the trajectory. These mathematical procedures are used in this case as generating principles, which allow to model continuous arcs of the line. While by an interpolation the curve always passes all the associated points, by an approximation the curve passes only the first and last point, and does not have to include necessarily other associated points, which depends particularly on the given approximation function. From the mathematical point of view, it does not matter weather it is about a movement of a mobile robot in a production hall, a forklift in a storehouse or a road vehicle on a street (Kvasnová, 2008).

MATERIALS AND METHODS

Ferguson interpolation curve

Ferguson interpolation curve of third degree allows an easy following of individual sections. The mathematical description of Ferguson curve bases on the position vectors

$\vec{G}$ a $\vec{H}$, respective points G and H, as well as on the tangent vectors $\vec{g}$ and $\vec{h}$ of the curve at these points. Ferguson curve is then given by equation (1) (Farin, 1993),

$$P(v) = \vec{m}.v^3 + \vec{n}.v^2 + \vec{p}.v + \vec{q}\,, \tag{1}$$

where: $\vec{P}(v)$ – position vector of a point of the curve; $\vec{m}$, $\vec{n}$, $\vec{p}$, $\vec{q}$ – coefficients' vectors; v – a parameter, for which is true that $\vec{P}(0) = \vec{G}$ a $\vec{P}(1) = \vec{H}$.

Performing the corresponding calculation, we obtain the vectors $\vec{m}$, $\vec{n}$, $\vec{p}$, $\vec{q}$, expressed by four equations (2), (3), (4) a (5)

$$\vec{m} = 2\vec{G} - 2\vec{H} + \vec{g} + \vec{h} \tag{2}$$

$$\vec{n} = -3\vec{G} + 3\vec{H} - 2\vec{g} - \vec{h} \tag{3}$$

$$\vec{p} = \vec{g} \tag{4}$$

$$\vec{g} = \vec{G} \tag{5}$$

Ferguson curve can also be expressed in form:

$$\vec{P}(v) = A(v)\vec{G} + B(v)\vec{H} + C(v)\vec{g} + D(v)\vec{H} \tag{6}$$

where: $A(v)$, $B(v)$, $C(v)$ a $D(v)$ are third degree polynomial, for which is true:

$$A(v) = 2v^3 - 3v^2 + 1 \tag{7}$$

$$B(v) = -2v^3 + 3v^2 \tag{8}$$

$$C(v) = v^3 - 2v^2 + v \tag{9}$$

$$D(v) = v^3 - v^2 \tag{10}$$

If we select in equation (7), (8), (9) and (10) the parameter v of the interval {0,1}, then we obtain a smooth curve that starts at point G and ends at point H. This type of curves is relatively suitable for modeling the trajectory of vehicles, since it ensures – due to appropriate choice of control points – safe passing of obstacles, although the length of the trajectory may increase.

Bezier interpolation curve

Bezier interpolation curves allow simple networking of following segments because the first two and the last two control points define a tangent to the curve at the endpoints. The touch vectors at the endpoints are determined by equations (11) and (12) (Pavlovkin & Jurišica, 2003a):

$$C'(0) = n(B_1 - B_0) \tag{11}$$

$$C'(1) = n(B_n - B_{n-1}), \tag{12}$$

where: n is the degree of the curve.

On the other hand, Bezier interpolation curve may cause – by selecting identical control points as by Ferguson curve – a risk of collision with an obstacle, moreover, the length of the trajectory increases.

Interpolation B-Spline curve

B-Spline curves exhibit many useful properties, in particular the parametric continuity C^2 of third degree curves, so that they can also be used as interpolation curves. The parametric continuity C^i defines in which way are the respective curves connected; the index of the continuity indicates the equality of respective i-derivates of the end-points of the individual curves; i.e. the continuity C^0 indicates that the curves are connected with an edge (the first derivatives are not equal), the continuity C^1 enables a smoother connection of the curves (as the first derivatives are equal) but with different convexity or concavity and thus with an abrupt change of centripetal acceleration. The continuity C^2 ensures that the connected curves have the same convexity (concavity), as the both second derivates are equal.

The computation can be performed by means of two methods – matrix inversion or searching for Bezier's control points.

Matrix inversion is a general method which can be used for all curves. If we can – based on the control points – calculate the coordinates of some points on the curve, then it is possible by the inverse procedure to determine the control points from known curve's points, too. The point, where the respective segments are continuing, lies in the anti-centroid of the triangle, defined by three consecutive control points.

Searching for Bezier's control points is basically an extension of Cardinal curves method, allowing to obtain a continuous C^2 curve. Bezier's control points V_i are located at the distance d_i from the interpolation points P_i; this ensures C^1 continuity. If the curve C^2 is to be continuous, it must be satisfied (13):

$$P_1 - 2(P_1 - d_1) + (P_0 + d_0) = (P_2 - d_2) - 2(P_1 + d_1) + P_1 \tag{13}$$

The sections d_0 and d_n we have to choose. Subsequently, we calculate the coefficients A_i and B_i and then we recursively calculate also the remaining sections $d_{i-1} = A_{i-1} + B_{i-1}.d_i$, thus obtaining the Bezier's control points. The possibility to choose the tangential vectors at the endpoints is a great advantage by vehicles, since the initial vector should have the same direction, as the vehicle is oriented. Thus it will not be necessary to turn the vehicle before starting the movement along the trajectory.

B-Spline curves obtained by both of these methods are almost the same (as we are looking for the same control points), and they differ only at the edges (different choice of tangential vectors at the endpoints). However, the method of searching for Bezier's control points is more preferred, as it is significantly faster than the matrix inversion method. Additionally, interpolation B-Spline curves are like Bezier curves susceptible to creating 'loops' and therefore they are used only where the development of such drawbacks does not mind or is excluded (Demidov, 2003; Pavlovkin & Jurišica, 2003a; Boonporm, 2012).

Bezier approximation curves

General Bezier curves allow an approximation of $n+1$ given points by an *n-degree* curve. The curve is described by the equation (14):

$$C(t)=\sum_{i=0}^{n} B_i^n(t)P_i \qquad t\in<0,1> \tag{14}$$

The basis functions of Bezier curves $B_i^n(t)$ constitute Bernstein base polynomials:

$$B_i^n(t)=\binom{n}{i}^{i}(1-t)^{n-1} \tag{15}$$

General Bezier curves have a relatively high smoothening ability, so that they are only marginally nearing to the individual control points. This is considerably disadvantageous in some applications, but elsewhere it may be useful; it depends on the specific conditions in which the vehicle is moving.

The general disadvantage of Bezier curves is the non-locality of changes – each point of the curve is influenced by all control points; i.e. changing an individual control point changes the shape of the whole curve. Therefore Bezier curves often consist of shorter segments. This way it is possible to obtain the locality of changes and to simplify the difficulty of the calculation, while maintaining all the advantages of the curves. To connecting individual sections, Bezier curves of third degree are mostly used. Basis functions can be determined in advance, since the order of the curve is always known at the beginning (Hwang et al., 2003).

B-Spline

Classic B-Spline curve is formed by linking Coons curves in such a way that the last three control points of one segment are identical to the first three points of the next section. In most cases there are used Coons curves of the third degree. The first segment is then determined by the points P_0, P_1, P_2 and P_3, the second segment by the points P_1, P_2, P_3 and P_4. The last point of the first segment and the first point of the second segment are identical, as they lie in the anti-centroid of the same triangle; thus the C^0 continuity is ensured (Demidov, 2003).

Joining of the individual sections is very smooth. B-Spline curves ensure the continuity C^{k-1} in the joint point, where k means the degree of the curve; i.e. B-Spline curve of the third degree guaranties a C^2 continuity. Using a Bezier curve, only the C^1 continuity is ensured. B-Spline curve therefore retains all the advantages of Bezier curves and it is a lot smoother when connecting the individual sections. B-Spline curve, however, has one major disadvantage – it does not pass the outermost points of the control polynomial. It can be removed by any of the control points will be multiple (Demidov, 2003).

If one control point is double, then the curve is significantly closing to that control point, and in a certain section it may even overlap the control polynomial. If the control point is triple, then the curve passes directly through this control point and it in the surroundings of such point it is identical with the control polynomial; however, this feature is useful only for the endpoints. So if the endpoints of the control polynomial are triple, the curve will interpolate the endpoints. The disadvantage is that near the endpoints the curve degenerates into line segments and it loses its smoothness. Another, more efficient method is to use different basis functions for the first two and the last two sections of the curve so that the curve passes through the endpoints. However, this method requires at least seven control points, so it cannot be used for simpler trajectories (Elbanhawi et al., 2015).

Interpolation by Ferguson curve

The interpolation by Ferguson curve, which is depicted in Fig. 1, is a suitable method for optimizing specific vehicles' trajectory, but it must be expected that the length of the trajectory gets extended compared to the direct path. The vehicle does not have to stop at the edges of the control polynomial; it has only to slow down sufficiently respected to the radius of turn. With this option of control points, the trajectory passes in a safe distance from individual obstacles and thus the risk of collision with one of the obstacles is eliminated.

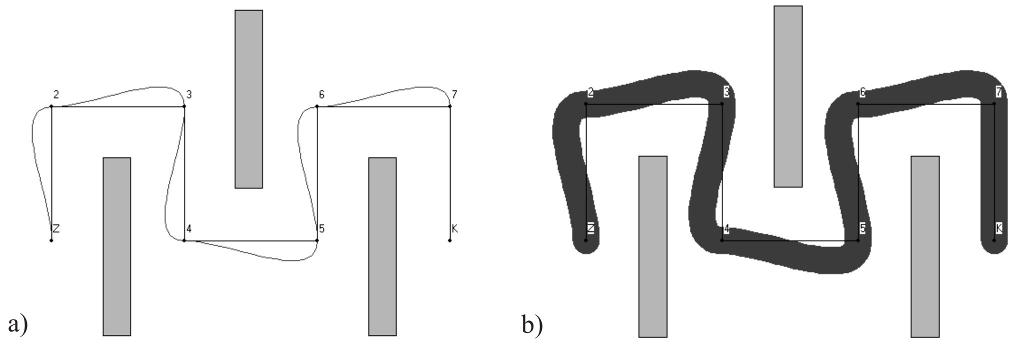


Figure 1. Interpolation by Ferguson curve. a) for a pointwise vehicle; b) for a real vehicle.

The calculation of the interpolation is always performed every second point. An element of the array has the coordinates of the point (x, y); an empty element of the array has the coordinates (-1, -1). The drawing of the interpolation curve is solved by means of the C++ graphics program Borland Delphi 2.0. This program draws the Ferguson curve basing on two given points and respective direction vectors at these points.

Interpolation by Bezier curve

Interpolation by Bezier curve, shown in Fig. 2, is by the specified setup of control points inappropriate for generating the trajectory of a vehicle, as it causes collisions with obstacles. Total length of the path is also substantially greater than by the interpolation by Ferguson curve. For use in a real environment, it would be necessary to the change are the coordinates of points 3, 4, 5 and 6 to achieve the desired path. The collision-free path of the vehicle for this way changed points is demonstrated in Fig. 3. From the comparison of trajectories in Fig. 2 and Fig. 3 it is apparent that the selection of the supporting points affects significantly the length and the shape of the trajectory. However, a suitable arrangement of the individual control points enables creating a usable trajectory, provided it is possible in respect to the location of the obstacles.

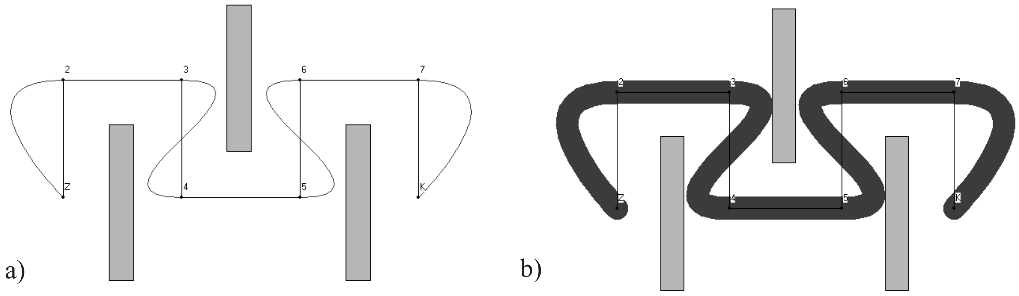


Figure 2. Interpolation by Bezier curve. a) for a pointwise vehicle; b) for a real vehicle.

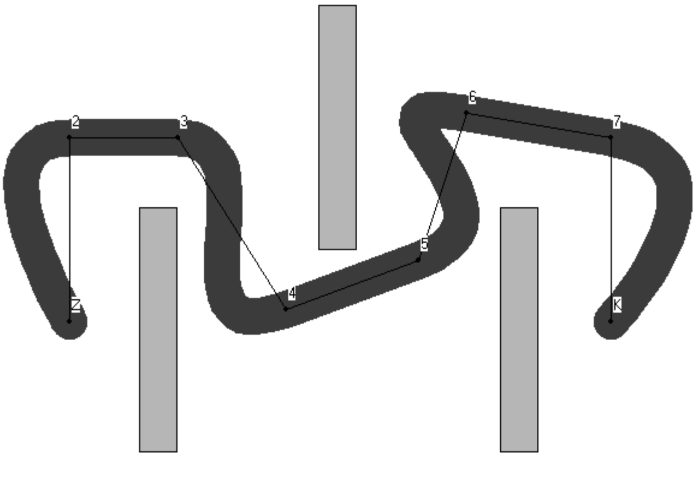


Figure 3. Interpolation by Bezier curve after changing the coordinates of the control points.

Approximation by Ferguson curve

Unlike the preceding interpolation cases, by an approximation the trajectory does not necessarily include the control points along the path. Approximation by Bezier curve, which is depicted in Fig. 4, is more convenient and shorter than the preceding two cases, but a large-size vehicle may interfere with an obstacle, as shown in Fig. 4b. The possibility of such a conflict can be avoided by changing the coordinates of the control point 4; the subsequent change in trajectory is demonstrated in Fig. 5. In such setup of control points, it is also possible by an appropriate shifting of the point 6 to shorten the overall length of the trajectory.

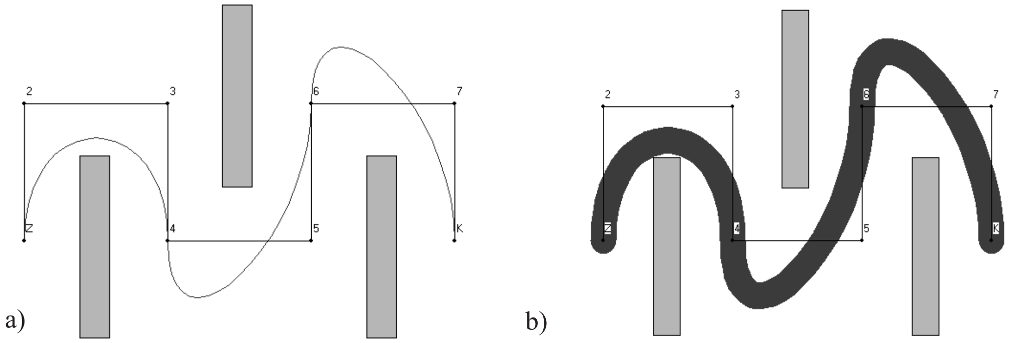


Figure 4. Approximation by Ferguson curve. a) for a pointwise vehicle; b) for a real vehicle.

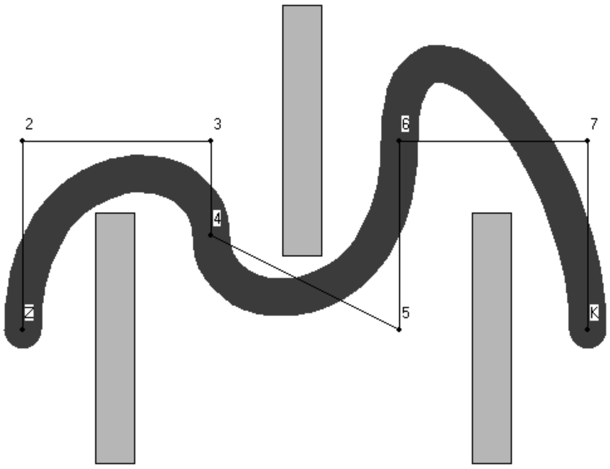


Figure 5. Approximation by Ferguson curve after changing the control point 4.

Approximation by Cubic B-Spline

By approximation of a piecewise linear trajectory by means of Cubic B-Spline curve we obtain a trajectory, which is shorter and smoother, and thus less time- and energy-consuming. The vehicle moves smoothly along such trajectory, i.e. with a smooth change of direction and speed of its movement, as depicted in Fig. 6 (Pavlovkin & Sudolský, 1999; Demidov, 2003).

The basic principle of generation of B-Spline curves is that we define Bezier curves of degree n at intervals (u_i, u_{i+1}); where n is the degree of the polynomial of the respective B-Spline curve and L is the number of segments of the B-Spline. So we create a sequence of points, namely the sequence $u_0 \ldots u_{L+2n-2}$. Not all points u_i, however, are different; if $u_i = u_{i+1}$ then it is a multiple point.

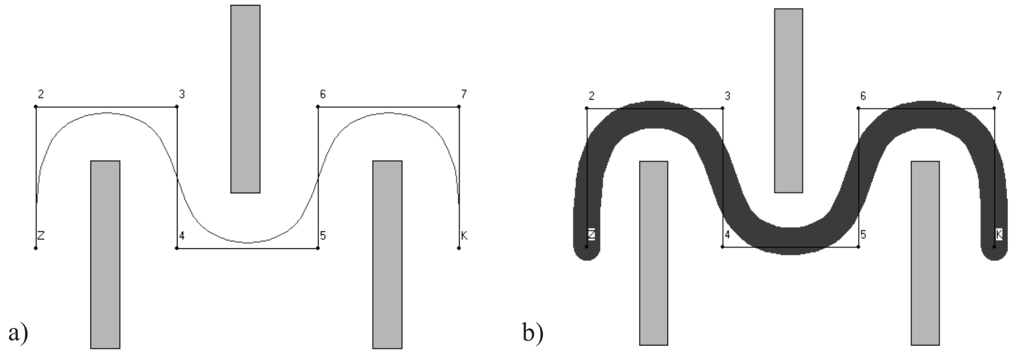


Figure 6. Approximation by Cubic B-Spline. a) for a pointwise vehicle; b) for a real vehicle.

To define B-Spline we use the interval (u_{n-1}, u_{n+L-1}) as its domain, these points are called domain points, while *L* means the potential number of segments of the curve. If all domain points are simple, then *L* is also the number of domain intervals. For every multiplicity of a domain point, the number of domain intervals reduces by one. The sum of multiplicity of all domain points corresponds with *L*, as it true that:

$$\sum_{i=n-1}^{L+n-1} r_i = L+1, \tag{16}$$

where: r_i means the multiplicity of domain points u_i.

For generating the B-Splines we used De Boor's algorithm. Let's true that: $u \in [u_I, u_{I+1}] \subset [u_{n-1}, u_{L+n-1}]$. We define:

$$d_i^k(u) = \frac{u_{i+n-k} - u}{u_{i+n-k} - u_{i-1}} d_{i-1}^{k-1}(u) + \frac{u - u_{i-1}}{u_{i+n-k} - u_{i-1}} d_i^{k-1}(u) \tag{17}$$

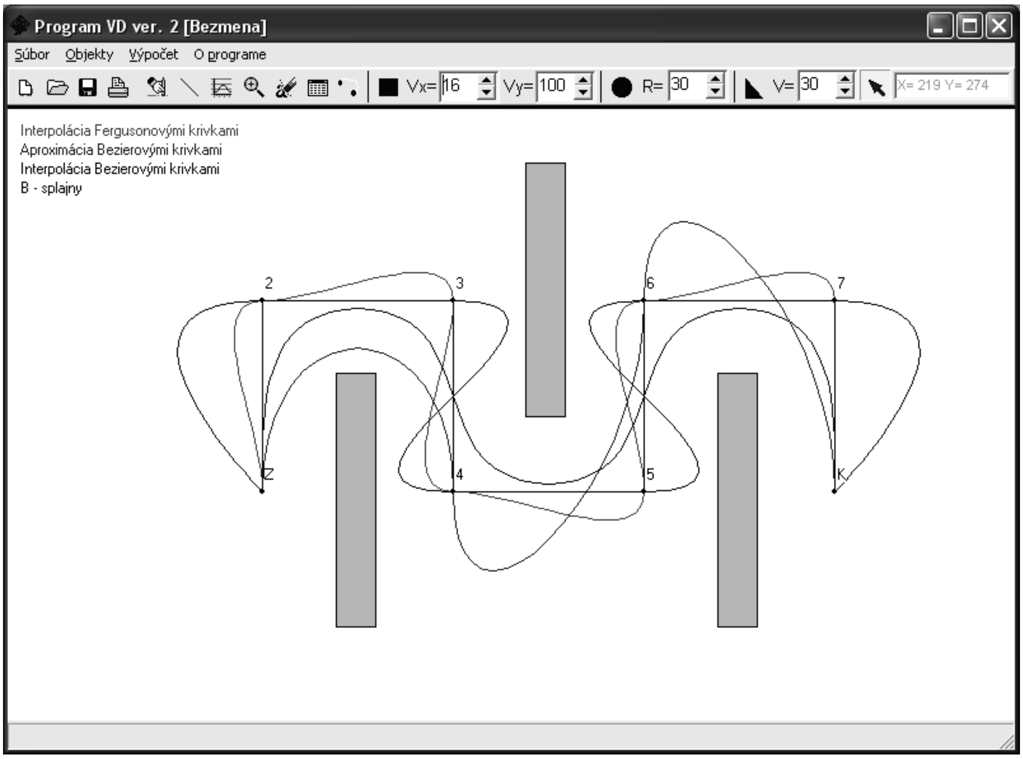
where $k = 1,\ldots,n-r$ and $i = I-n+k-1,\ldots,I+1$
which is the degree of B-Spline given the parametr *u*.

$$s(u) = d_{I+1}^{n-r}(u), \tag{18}$$

while $d_i^0(u) = d_i C.$

RESULTS AND DISCUSSION

To analyze the wide range of described approximation and interpolation curves, we used our self-created computer program 'VD' (abbreviation for 'path draw' in Slovakian language). The graphical interface of this program and a comparison of some generated curves are depicted in Fig. 7. Within this program, we defined the known position of the obstacles as well as the control points of the trajectory. Then we selected a mathematical model of the trajectory by a preset approximation or interpolation curve formula, we obtained a graphical and a numerical result. The results were subsequently analyzed graphically and numerically.



Program VD ver. 2 [Bezmena]

Vx= 16 Vy= 100 R= 30 V= 30 X= 219 Y= 274

Interpolácia Fergusonovými krivkami
Aproximácia Bezierovými krivkami
Interpolácia Bezierovými krivkami
B - splajny

Figure 7. Graphical interface of the program VD with examples of trajectories.

The overall length of the trajectory between points P_0 and P_n is defined by equation (19):

$$d(P_0,P_n)=\sum_{i=0}^{n}\int_{a_i}^{a_{i+1}}\sqrt{1+\left(f_i^{I}(x)\right)^2}\,dx \qquad (19)$$

where: P_0 , P_n – start and end point of the trajectory; a_i – individual section of the trajectory; $f_i(x)$ is the respective mathematical function formula of the curve.

To investigate the radius of turn for a selected point of the trajectory – which may be necessary due the specific limitations of the vehicles, the formula (20) is used:

$$R^2=\frac{\left(1+\left(f^{I}(x)\right)^2\right)^3}{\left(f^{II}(x)\right)^2} \qquad (20)$$

where: R means the radius of curvature of the trajectory at a specicic point and; $f(x)$ is the mathematical function formula of the curve.

Basing on the graphical analysis of the curves and on calculations using the preceding formulas, an overall comparison of the various options optimizing of the vehicles' movement between obstacles gave the best results for the approximation based on Cubic B-Spline. The mathematical model of such trajectory exhibits fluency, both in terms of necessary speed changes, and regarding the smoothness of the change of direction. Important is also the fact that of all the analyzed trajectories this one is the shortest, which yields energy saves. Although the shortening of the trajectory need not be regarded as considerable, compared to other options, the total saving of energy may be high, in particularly over a longer period of time or if the same trajectory repeats regularly several times (stock houses, factories, agricultural activities) (Pavlovkin & Jurišica, 2003b). Finally, it has to be pointed out that the trajectory approximated by Cubic B-Spline exhibits relative high level of safety, as it passes all the obstacles – unlike some other trajectories – with sufficient distance and virtually eliminates any possibility of collision of the vehicle with an obstacle (Kvasnová, 2014).

CONCLUSIONS

We analyzed the trajectory of a vehicle along a defined path between obstacles. We used mathematical simulations of various approximation and interpolation curves by means of our own program 'VD'. Based on the results, we concluded the most convenient method – considering the smoothness of the trajectory, its length, shape and obstacle clearance – using the approximation by Cubic B-Spline curve.

The current development of defining and optimizing vehicles' trajectories prefers a direct control through a system of sensors placed directly on vehicles. Sensors provide information about possible obstacles along the path; the information is evaluated by a computer installed in the vehicle, which controls the vehicle to change flexibly its trajectory (Fu et al., 2013). Although the technical level of sensors has considerably improved and their price has sunk, this method is, however, still technically more complicated, which poses an enhanced risk of malfunctions and increases acquisition and maintenance costs. Therefore, if the layout of obstacles is permanent, the approximation and interpolation methods described in the article to define its trajectory are still useful. In such a layout, vehicles may be equipped only with a simple 'emergency stop sensor' to increase safety and to avoid unpredicted collisions.

REFERENCES

Boonporm, P. 2012. Online path replanning of autonomous mobile robot with Spline based algorithm. In: *IPCSIT*. IACSIT Press, Singapore, Vol. 23. http://www.ipcsit.com/vol23/11-ICSMO2012-W0022.pdf. Accessed 25.3.2015.

Demidov, E. 2003. *An Interactive Introduction to Splines*. http://www.people.nnov.ru/fractal/Splines/Intro.htm. Accessed 8.8.2015.

Elbanhawi, M., Simic, M. & Jazar, R.N. 2015. Continuous Path Smoothing for Car-Like Robots Using B-Spline Curves. *Journal of Intelligent & Robotic Systems* **80**(1), 23–56.

Farin, G. 1993. *Curves and Surfaces for Computer Aided Geometric Design. A Practical Guide*. Academic Press, Inc.

Fu, W., Hadj-Abdelkader, H. & Colle, E. 2013. Reactive collision avoidance using B-spline representation: Application for mobile robot navigation. In: *European Conference on Mobile Robots (ECMR) 2013*, Barcelona, pp. 319–324.

Hwang, J.H., Arkin, R.C. & Kwon, D.S. 2003. Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control. In: *Intelligent Robots and Systems* **2**, 1444–1449.

Kvasnová, P. 2008. Universal loader and the possibilities of its use / Univerzálny nakladač a možnosti jeho využitia. In: *Technika odpadového hospodárstva*. Technická univerzita vo Zvolenu, Zvolen, Slovakia, pp. 70–76 (in Slovak).

Kvasnová P. 2014. Evaluation of risk factors / Posudzovanie rizikových faktorov. In: *Bezpečná práca: dvojmesačník pre teóriu a prax bezpečnosti práce* **45**(5), 3–9 (in Slovak).

Pavlovkin, J. & Jurišica, L. 2003a. Mobile robot navigation (1) / Navigácia mobilného robota (1). In: *AT&P JOURNAL* **2**, 72–73 (in Slovak).

Pavlovkin, J. & Jurišica, L. 2003b. Mobile robot navigation (2) / Navigácia mobilného robota (2). In: *AT&P JOURNAL* **3**, 100 (in Slovak).

Pavlovkin, J. & Sudolský, M. 1999. Aplikácie B-splajnov na plánovanie lokálnych ciest mobilného robota. In: *ROBTEP `99 Automatizacion – Robotics in Theory and Practice*. Strojná fakulta Technickej univerzity v Košiciach, Košice, Slovakia, pp. 215–220 (in Slovak).