

Path Deformation Method with Constraints on Normal Curvature for Wheeled Robots in Precision Agriculture Based on Second-Order Cone Programming

T. A. Tormagov

Moscow Institute of Physics and Technology, Moscow, Russia

e-mail: tormagov@phystech.edu

Received June 26, 2023

Revised December 30, 2023

Accepted January 20, 2024

Abstract—In precision agriculture, path planning for agricultural robots with complete covering a three-dimensional landscape is an essential task. For robots with front wheels steering the normal curvature of the trajectories should be limited to some value determined by the characteristics of the vehicle. The paper considers a method of deformation of these paths to account for obstacles for trajectories described by homogeneous cubic B-splines. We propose an optimization problem that allows calculating paths with minimizing skips in the coverage. The considered problem is convex and belongs to the class of second-order cone programming, which entails the possibility of its computationally efficient solution. The computational examples are presented.

Keywords: precision farming, precision agriculture, obstacle avoidance, complete coverage path planning, second-order cone programming, SOCP

DOI: 10.31857/S0005117924020037

1. INTRODUCTION

In precision agriculture, navigation measurements and monitoring using various sensors are used to improve crop cultivation technologies. Effective use of the collected data allows, for example, to reduce the work time, the area of the land plots involved in farming and the amount of necessary resources. One of the work planning problems in precision farming is to construct paths for agricultural robots that completely cover a specified area of a three-dimensional landscape.

Planned movement trajectories must be realizable. The normal curvature u of the trajectories of robots with front wheel steering must satisfy the condition

$$\|u\| < u_{\max}, \quad (1)$$

where u_{\max} is the maximum possible normal curvature of the trajectory. The value of u_{\max} is defined by the formula

$$u_{\max} = \frac{\tan \alpha_{\max}}{L}, \quad (2)$$

where L is the distance between the front and rear robot axles (wheelbase), α_{\max} is the maximum efficient front wheels steering angle [1]. For a vehicle with two front wheels, the angles of their rotation may be different. For this reason, we use the effective rotation angle in the formula (2).

A differential wheeled agricultural machine allows can turn on the spot (zero turn), and the condition (1) is not necessary to implement the trajectory. In this paper, we consider the path planning for the machines with front wheel steering for which the normal curvature of the realized trajectories must be limited. It is assumed that during the field processing the movement is carried out by front gear, and the reverse gear can only be used for the turns at the field boundary.

Complete coverage path planning for landscapes is used, for example, in sowing and planting plants, treating them from pests, harvesting and mowing lawns. Parallel or almost parallel paths are often used to cover the field. When the terrain of the field can be neglected, path planning is performed for a flat surface. The features of constructing flat parallel paths in arable farming are considered in [2]. Also the problem of choosing the direction of constructing paths taking into account the shape and length of turns and the problem of decomposing a field of complex shape into sections, for each of which parallel paths can be planned separately, are reviewed in [2]. The path planning on a three-dimensional landscape is noticeably different from the flat case. In [3–5] it is proposed to construct a three-dimensional coating relative to some initial swath that crossing the field. In general, the initial path is curved. In [4], a search for neighboring paths by solving the problem of the intersection of a cylinder with a surface is proposed. For a parametrically defined curved surface an approach to construct strictly parallel paths without restrictions on curvature is proposed in [6]. As shown in the work on the planning of three-dimensional paths in arable farming [3], the target functions that allow estimating field coverage can take into account soil erosion by water, the lengths of turns between swaths, gaps in the coverage due to curvature restrictions. In [5], an complete coverage path planning algorithm for constructing swaths along a gentle landscape with restrictions on the normal curvature and on the overlap width of neighboring swaths is proposed.

The obstacles on the field can be both dynamic and stationary. The dynamic obstacles can change their position over time. For example, it can be animals or other agricultural machines. For autonomous robots, the trajectories of dynamic obstacle avoidance planned during the movement. Information about dynamic obstacles is collected in real time using stereo cameras, lidars and other sensors [7]. Dynamic consideration of obstacles requires the trajectory replanning [8]. Solving trajectory planning problems in real time requires a certain amount of computing resources. For this reason, stationary obstacles must be considered when designing a coverage prior to starting field processing. Note that in both the static and dynamic obstacle cases the condition (1) must be met.

In [2, 9–11] field decomposition methods are discussed. For the different configurations of obstacles, these methods allow to divide the field into separate sections, each of which can be covered by parallel paths. Such separation often requires the allocation of a sufficiently wide area for turns near the obstacle. In the literature, many problems are considered in which obstacle avoidance trajectories are constructed with minimizing the path length. We can be notice the approaches based on the graph search algorithms [12, 13] and path deformation algorithms using penalty functions (artificial potential fields) [14–16]. However, in precision farming, solving the problem of deforming the path while minimizing its length can lead to significant skips in the coverage. Therefore, an approach to reduce the unprocessed area is required.

In this paper, we propose a method that allows for planning field coverage while taking obstacles into consideration by solving the second-order cone programming problem for machines with front wheel steering. Second-order cone programming (SOCP) problem can be formulated in general as follows. [17, 18].

Problem 1.

$$\underset{x}{\text{minimize}} \quad f^T x \tag{3}$$

subject to

$$\|A_i x + b_i\| \leq c_i^T x + d_i, \quad i = 1, \dots, N, \quad (4)$$

where $x \in R^l$ (l is the dimension of the variable), $f \in R^l$, $A_i \in R^{(l_i-1) \times l}$, $b_i \in R^{l_i-1}$, $c_i \in R^l$, $d_i \in R$, N is number of constraints, and l_i are natural numbers.

Denote by $\|\cdot\|$ the Euclidean norms of vector. SOCP problems are convex and have computationally efficient solution methods, see [19, 20].

Note that after coverage path planning, it is necessary to determine the routes of one or more machines along the obtained swaths, which allows processing in the shortest time or with the minimum route length. In general, routing is a computationally time-consuming problem. Its exact solution often cannot be obtained in an acceptable time. Therefore, in practice, heuristic routing algorithms are often used. Examples of such algorithms based on the simulated annealing method are given in [5, 21, 22].

2. PROPOSED METHOD

Let us assume that the field coverage has been constructed without taking into account obstacles. It is required to change the swaths that cross the boundaries of obstacles so that each path is feasible and the area of the untreated sections is as small as possible. For realizability, it is necessary that the path does not cross obstacles and that the curvature condition (1) is met.

For proposed method, it is convenient to use a local right Cartesian coordinate system ENU (x, y, z) (East, North and Up) to determine the coordinates of points in space. To determine it, a certain point on the field is fixed, which is then used as the origin of the coordinate system. The x axis is directed to the east, the y axis is directed to the north, and the z axis complements the axes to a right-handed set of vectors. The x and y axes lie in a tangent plane to the ellipsoid of the Earth model. A digital elevation model (DEM) is used to describe the surface on which robots operate. It is defined by the function $z = (x, y)$, which allows to determine the value of z for the known coordinates x and y . However, in practice, it is often impossible to accurately determine the height of each point on a field. Therefore, height interpolation is used. First, data on the heights of reference points on the field are collected. Then, the height determination function $z = (x, y)$ is constructed based on the data obtained. Examples of methods for constructing such functions on a uniform grid include bilinear interpolation and bicubic interpolation [4]. On an irregular grid bivariate B-splines interpolation [23] and Kriging [24] can be used.

It is convenient to use smooth functions to describe the trajectory of wheeled robots. One example of such a function is the homogeneous cubic B-spline [25]. The spline curve is defined by a set of control points r_1, r_2, \dots, r_n . The set is supplemented with points

$$r_0 = 2r_1 - r_2, \quad r_{n+1} = 2r_n - r_{n-1}. \quad (5)$$

Every four adjacent control points $r_{i-1}, r_i, r_{i+1}, r_{i+2}$ set an elementary B-spline $r^{(i)}(t)$, $i = 1, \dots, n-1$, by parametric formula

$$r^{(i)}(t) = \frac{(1-t)^3}{6} r_{i-1} + \frac{4-6t^2+3t^3}{6} r_i + \frac{1+3t+3t^2-3t^3}{6} r_{i+1} + \frac{t^3}{6} r_{i+2}, \quad (6)$$

where the spline parameter t takes values from the interval $[0, 1]$, see Fig. 1. The curve $r(t)$, $t \in [0, n]$ consists of a set of elementary splines $r^{(i)}(t)$, $i = 1, \dots, n$. It can be defined by formula

$$r(t) = r^{(i)}(t - (i-1)), \quad i = \lfloor t + 1 \rfloor, \quad t \in [0; n), \quad (7)$$

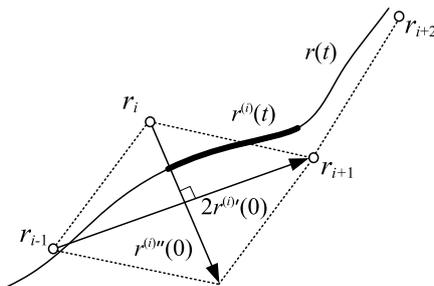


Fig. 1. The spline curve $r(t)$, the elementary spline $r^{(i)}(t)$ (highlighted on the spline curve) determined by four equidistant control points r_{i-1} , r_i , r_{i+1} , r_{i+2} , vectors $r^{(i)'}(0)$ and $r^{(i)''}(0)$.

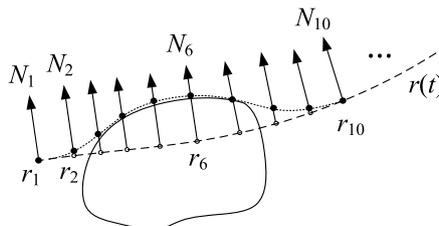


Fig. 2. The contour of the obstacle (solid line), the spline curve $r(t)$ before (dashed line) and after (dotted line) the deformation of the path, the search directions N_i and control points r_i .

$r(n) = r^{(n-1)}(1)$, where $[t]$ is integer part of t . As it can be seen from the formulas (6) and (7), $r(t)$ is continuous with continuous first and second derivatives including at the junctions of elementary splines. Note that both the spline trajectory and the control points may not lie on the surface $z = (x, y)$ defined by the digital elevation model. However, they are usually located near this surface, which is suitable for the practical application of such a description of paths.

In [5, 7] a method for constructing obstacle-free path coverage using limited curvature paths described using homogeneous cubic B-splines is proposed. Due to the presence of obstacles in the field, this method cannot be used directly. In this paper it is proposed to modify the field coverage constructed without taking into account obstacles by solving a set of path deformation problems. Since the spline curve is defined by a set of control points, the definition of a deformed path is reduced to specifying its control points.

Denote the coordinates of control points by x_i, y_i, z_i , such that $r_i = (x_i, y_i, z_i)$. Let us define a set of unit vectors $N_i = (N_i^x, N_i^y, N_i^z) \in R^3, i = 1, \dots, n$, so that they are parallel to the tangent planes to the surface at points $(x_i, y_i, z(x_i, y_i))$ and the direction of the normal of the spline projection onto the tangent plane for $(x_i, y_i, z(x_i, y_i))$ matches N_i . Along the directions N_i we look for new positions of the spline control points $r_i + N_i d_i, i = 1, \dots, n$, where the ranges of values of d_i are determined by the geometric constraints of the problem, see Fig. 2.

Let us consider the issue of constraints on the normal curvature of a trajectory (1).

Statement 1 (sufficient conditions). *For a flat homogeneous cubic B-spline with control points $r_i + N_i d_i, i = 1, \dots, n$, and the complement (5), to be meet the condition (1), it is sufficient that*

$$\left\| \begin{matrix} x_{i-1} + N_{i-1}^x d_{i-1} - 2(x_i + N_i^x d_i) + x_{i+1} + N_{i+1}^x d_{i+1} \\ y_{i-1} + N_{i-1}^y d_{i-1} - 2(y_i + N_i^y d_i) + y_{i+1} + N_{i+1}^y d_{i+1} \end{matrix} \right\| \leq u_{\max} \hat{l}_i(d_{i-1}, d_{i+1}), \tag{8}$$

$$\begin{aligned} 4\hat{l}_i(d_{i-1}, d_{i+1}) = & (x_{i+1} - x_{i-1})^2 + 2(x_{i+1} - x_{i-1})(N_{i+1}^x d_{i+1} - N_{i-1}^x d_{i-1}) + \\ & + (y_{i+1} - y_{i-1})^2 + 2(y_{i+1} - y_{i-1})(N_{i+1}^y d_{i+1} - N_{i-1}^y d_{i-1}), \end{aligned} \tag{9}$$

for $i = 1, 2, \dots, n$.

Note that (8) only implies the fulfillment of condition (1) at the junction points of elementary splines. The suitability of this condition for practical application is due to two factors. The first factor is that the maximum value of $\|r^{(i)''}(t)\|$ for each elementary spline achieves at $t = 0$ or $t = 1$. The proof of this statement is given in [5]. The second factor is that the value of $\|r^{(i)'}(t)\|$ reflects the correspondence between small changes in the trajectory length and corresponding changes in the spline parameter. It is approximately constant for an elementary spline with equidistant control points.

In the case where the surface (x, y) is an inclined plane, the conditions for normal curvature can be obtained from (8) by rotating the coordinate system. For a curved surface, the curvature vector $k(t)$ may not coincide with the vector of normal curvature $u(t)$. Denote the maximum length of the vector between adjacent control points of the spline by D_s . Consider the case of gentle surfaces for which motion over a distance of $2D_s$ can be considered with good accuracy as motion on an inclined plane. In precision agriculture, this assumption is valid for a wide range of applications. In this instance, the magnitude of the vector $u(t)$ can be approximated using the curvature of the trajectory in the tangent plane. Define the vector r'_i as

$$r'_i = \frac{r_{i+1} - r_{i-1}}{\|r_{i+1} - r_{i-1}\|}. \tag{10}$$

For each control point r_i let us define a flat coordinate system (\tilde{x}, \tilde{y}) with the origin at the point r_i with coordinate axes co-directed to the vectors r'_i and N_i . The condition (8) take the form

$$\left\| \begin{matrix} \tilde{x}_{i-1} + \tilde{N}_{i-1}^x d_{i-1} + \tilde{x}_{i+1} + \tilde{N}_{i+1}^x d_{i+1} \\ \tilde{y}_{i-1} + \tilde{N}_{i-1}^y d_{i-1} - 2d_i + \tilde{y}_{i+1} + \tilde{N}_{i+1}^y d_{i+1} \end{matrix} \right\| \leq u_{\max} \tilde{l}_i(d_{i-1}, d_{i+1}), \tag{11}$$

where

$$\begin{aligned} 4\tilde{l}_i(d_{i-1}, d_{i+1}) = & (\tilde{x}_{i+1} - \tilde{x}_{i-1})^2 + 2(\tilde{x}_{i+1} - \tilde{x}_{i-1})(\tilde{N}_{i+1}^x d_{i+1} - \tilde{N}_{i-1}^x d_{i-1}) + \\ & + (\tilde{y}_{i+1} - \tilde{y}_{i-1})^2 + 2(\tilde{y}_{i+1} - \tilde{y}_{i-1})(\tilde{N}_{i+1}^y d_{i+1} - \tilde{N}_{i-1}^y d_{i-1}), \end{aligned} \tag{12}$$

in the coordinate system for the control point r_i . Value pairs $(\tilde{x}_{i-1}, \tilde{y}_{i-1})$ and $(\tilde{x}_{i+1}, \tilde{y}_{i+1})$ are coordinates of projections of neighboring control points in the system (\tilde{x}, \tilde{y}) . $(\tilde{N}_{i-1}^x, \tilde{N}_{i-1}^y)$ and $(\tilde{N}_{i+1}^x, \tilde{N}_{i+1}^y)$ are the coordinates of their normals in the system (\tilde{x}, \tilde{y}) . Note that in general the coordinate system (\tilde{x}, \tilde{y}) is different for each r_i . Therefore the calculated values of $(\tilde{x}_{i-1}, \tilde{y}_{i-1})$, $(\tilde{x}_{i+1}, \tilde{y}_{i+1})$, $(\tilde{N}_{i-1}^x, \tilde{N}_{i-1}^y)$, and $(\tilde{N}_{i+1}^x, \tilde{N}_{i+1}^y)$ for r_i can not be used in the curvature conditions for r_{i-2} and r_{i+2} . If the points r_{i-2} and r_{i+2} exist then a separate calculations of projections in coordinate systems with origins at r_{i-2} and at r_{i+2} are required.

The curvature conditions (11) have the form of the second-order cone (4). The right side of (11) depends linearly on the variables $d_i, i = 1, \dots, n$. It allows us to formulate the problem of path deformation with the curvature constraints in the form of the following second-order cone programming problem.

Problem 2.

$$\underset{q}{\text{minimize}} \quad q_{n+1} \tag{13}$$

subject to (11) for $i = 1, \dots, n$,

$$\|w^T q\| \leq q_{n+1}, \tag{14}$$

$$\underline{d}_i \leq q_i \leq \overline{d}_i, \quad i = 1, \dots, n, \tag{15}$$

where $q \in R^{n+1}$, q_i is the i th component of the vector q .

The first n components of the optimal vector q are equal to shifts of the control points that need to be carried out, $d_i = q_i, i = 1, \dots, n$. The values of \overline{d}_i and $\overline{\overline{d}}_i, i = 1, \dots, n$ define geometric constraints on the shift of control points in the direction N_i caused by obstacles and field boundaries. A bypass direction is selected for each obstacle. The first n components of the vector $w \in R^{n+1}$ are fixed weights, which can be chosen as equal or depending on the location of the control points relative to the obstacles. The last component of the vector w is zero, $w_{n+1} = 0$. The optimization criterion (13) and the condition (14) allow to minimize the shift of the control points and reduce gaps in the coverage formed when bypassing obstacles due to restrictions on the path normal curvature.

In practice, we can use the representation of obstacle boundaries and the field contour as closed polylines to determine the values of \overline{d}_i and $\overline{\overline{d}}_i, i = 1, \dots, n$. The projections of these closed polylines onto the local horizon form polygons. In the local horizontal plane, the search for \overline{d}_i and $\overline{\overline{d}}_i$ represents the selection of the traversing direction for each obstacle and solving the intersection search problem for a straight line and a polygon.

The proposed optimization problem is convex and admits computationally efficient solution methods. Another computational feature of this problem is that a conclusion about its feasibility or infeasibility can be reached during the solution process. Note that the proposed approach is not aimed at finding directions to bypass obstacles. It serves as a tool for constructing paths within a given convex area defined by the constraints (15). If the path deformation problem is infeasible and an acceptable path exists within the specified boundaries, then this may be due to the inability to meet the conditions for curvature when shifting points along the selected directions. The inability to satisfy the curvature constraints may be due to the fact that the uniformity of the control point locations is violated for large displacements. In addition, the projections on the plane (x, y) of the search directions N_i can intersect. This fact leads to solutions with self-intersecting trajectories for which the curvature conditions are often violated. These drawbacks can be eliminated using the following path-deformation algorithm based on the relaxation of the problem's constraints and representing the path with equidistant control points.

Algorithm 1 (path deformation).

1. Solve the Problem 2 with the initial value of u_{\max} . If a solution exists, parameterize the trajectory using evenly spaced control points. The Problem 2 is repeatedly solved using the current positions of the control points. Return the solution as an answer, if it exists.
2. Solve the Problem 2 for the current location of the control points and the maximum value of normal curvature $u_{\max} + \Delta, \Delta > 0$. If a solution exists, parameterize the trajectory using evenly spaced control points and go back a step 1. If a solution does not exist, increase the value of Δ and repeat the calculations.

The algorithm at each stage of its operation involves choosing a change in the curvature of Δ . The number of iterations in the algorithm depends on the value of Δ that is chosen.

3. THE COMPUTATIONAL EXAMPLES

Figure 3 shows an example of solving the path deformation problem for three obstacles. The value of u_{\max} is 0.4 m^{-1} and the swath width is 1 m. Firstly, the field coverage was constructed without taking into account obstacles using the method given in [5]. Then the path was deformed with unit weights ($w_i = 1$) for all control points. The parameter Δ was selected equal to

$$\Delta = (u_{\max}^{-1} - 0.09m)^{-1} - u_{\max}, \tag{16}$$

where m is the iteration number of the path deformation algorithm in step 2. This relaxation of the curvature constraint corresponds to a decrease in the allowed radius of curvature of the

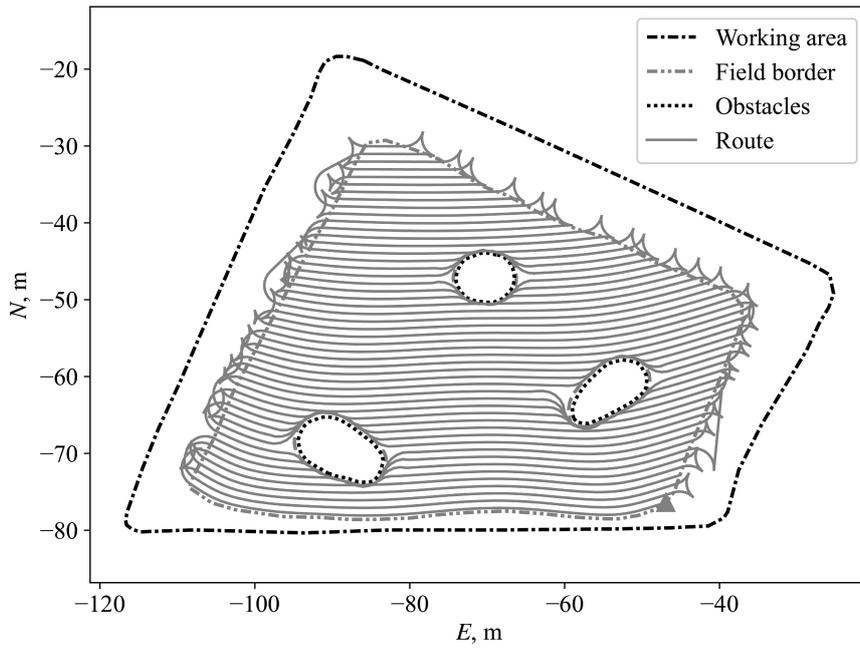


Fig. 3. The example of the path deformation for three obstacles on the field.

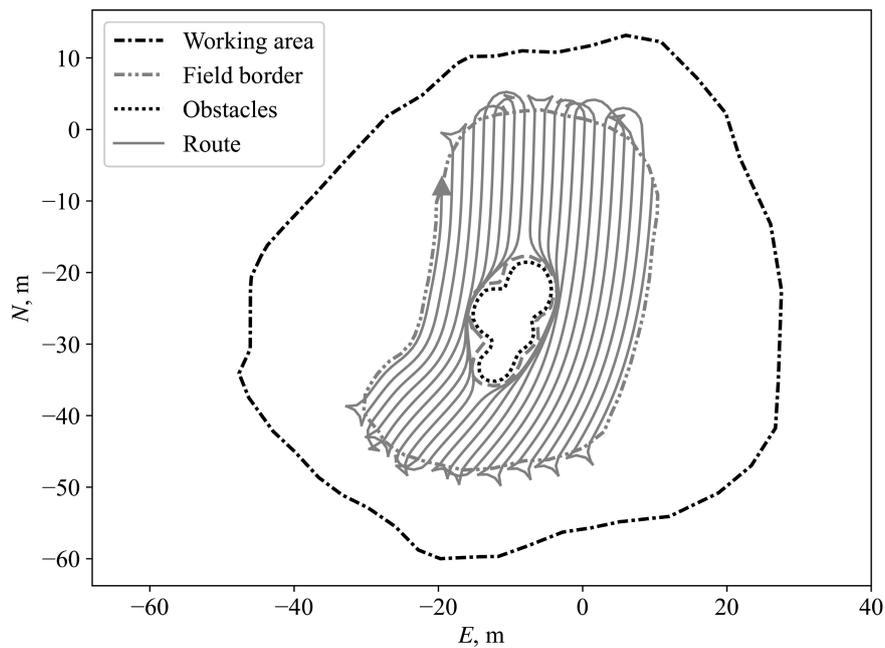


Fig. 4. The example of path deformation using the penalty functions (artificial potential fields).

trajectory by 0.09 meters for each iteration. Obviously, the number of iterations is limited because it is necessary to provide a non-negative value for the expression $u_{\max}^{-1} - 0.09m$.

Figure 4 shows an example of solving the path deformation problem using penalty functions (artificial potential fields [15, 16]). The result of the application of the path deformation method based on the solution to problem 2 (SOCP) is shown in Fig. 5. It can be seen that the optimization criteria (13) for the problem 2, together with the conditions (14), made it possible to reduce the untreated area around a non-convex obstacle, which was formed due to restrictions on the normal curvature.

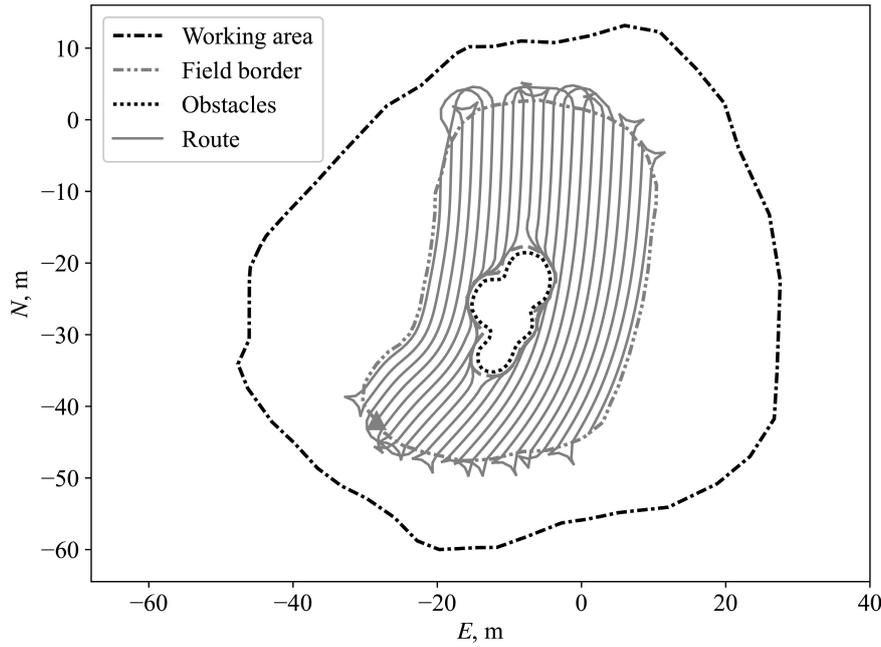


Fig. 5. The example of path deformation based on solving the SOCP problem.

In all examples, the algorithm described in [19] (with CVXPY [26, 27]) was used for solving second-order cone programming problems. The routes for bypassing swaths are designed for a single robot with forward and reverse gears in headland turns. The ant colony algorithm (ACO) was used to optimize the length of the route. The features of its application to routing problems are discussed in [28]. It should be noted that minimizing the route length for this example does not necessarily translate into minimizing the working time, at least due to the requirement for stops when changing directions on headland turns.

4. CONCLUSION

The paper proposes a method for deforming trajectories to account for obstacles, which is applicable in precision agriculture when planning the field coverage with swaths. The results are obtained for the case of describing the motion trajectory by the homogeneous cubic B-spline. The proposed method allows to obtain paths for front wheel steering robots, for which the maximum acceptable normal curvature of the path is limited. The trajectory is calculated by minimizing the Euclidean distance between control points of the initial and final splines, which reduces the area that remains unprocessed. The novelty of the approach is the formulation of the path deformation with a curvature constraint in the form of the second-order cone programming problem, which is convex and has computationally efficient solution methods. The limitation of the approach is the need for a sufficiently flat field relief, which often meets in practical applications of precision agriculture.

APPENDIX

Proof of Statement 1. In the case of motion in the plane (x, y) , the norm of the curvature vector $k(t)$ of the spline curve at the point $r^{(i)}(t)$ is determined by the formula

$$\|k(t)\| = \frac{\|r^{(i)''}(t)\| \sin \varphi(t)}{\|r^{(i)'}(t)\|^2}, \tag{A.1}$$

where $r^{(i)'}(t)$ and $r^{(i)''}(t)$ are vectors of the first and second derivatives of the spline by parameter, $\varphi(t)$ is the angle between the vectors $r^{(i)'}(t)$ and $r^{(i)''}(t)$. In this case, the curvature vector $k(t)$ coincides with the normal curvature vector $u(t)$. The vectors of the first and second derivatives at the junction of elementary splines have the form

$$r^{(i)'}(0) = \frac{1}{2}(r_{i+1} + N_{i+1}d_{i+1} - r_{i-1} - N_{i-1}d_{i-1}), \quad (\text{A.2})$$

$$r^{(i)''}(0) = r_{i-1} + N_{i-1}^r d_{i-1} - 2(r_i + N_i^r d_i) + r_{i+1} + N_{i+1}^r d_{i+1}. \quad (\text{A.3})$$

Note that the function $\hat{l}_i(d_{i-1}, d_{i+1})$, defined by the formula (9), is a lower bound for $\|r^{(i)'}(0)\|^2$, which does not take into account the quadratic terms of displacement.

$$\|r^{(i)'}(0)\|^2 = \hat{l}_i(d_{i-1}, d_{i+1}) + \frac{1}{4}(N_{i+1}^x d_{i+1} - N_{i-1}^x d_{i-1})^2 + \frac{1}{4}(N_{i+1}^y d_{i+1} - N_{i-1}^y d_{i-1})^2. \quad (\text{A.4})$$

Using (8), we obtain

$$\|r^{(i)''}(0)\| \sin \varphi(0) \leq u_{\max} \|r^{(i)'}(0)\|^2. \quad (\text{A.5})$$

This concludes the proof of (1) for the junctions of elementary splines.

REFERENCES

1. Gilimyanov, R.F., Pesterev, A.V., and Rapoport, L.B., Smoothing Curvature of Trajectories Constructed by Noisy Measurements in Path Planning Problems for Wheeled Robots, *J. Comput. Syst. Sci. Int.*, 2008, vol. 47, no. 5, pp. 812–819.
2. Jin, J. and Tang, L., Optimal Coverage Path Planning for Arable Farming on 2D Surfaces, *Trans. ASABE. St. Joseph, MI: ASABE*, 2010, vol. 53, no. 1, pp. 283–295.
3. Jin, J. and Tang, L., Coverage Path Planning on Three-Dimensional Terrain for Arable Farming, *J. F. Robot.*, 2011, vol. 28, no. 3, pp. 424–440.
4. Hameed, I.A., La Cour-Harbo, A., and Osen, O.L., Side-to-Side 3D Coverage Path Planning Approach for Agricultural Robots to Minimize Skip/Overlap Areas between Swaths, *Rob. Auton. Syst.*, 2016, vol. 76, pp. 36–45.
5. Tormagov, T. and Rapoport, L., Coverage Path Planning for 3D Terrain with Constraints on Trajectory Curvature Based on Second-Order Cone Programming, in *Advances in Optimization and Applications*, Olenov, N.N. et al., Eds., Cham: Springer International Publishing, 2021, pp. 258–272.
6. Gálvez, A., Iglesias, A., and Puig-Pey, J., Computing Parallel Curves on Parametric Surfaces, *Appl. Math. Model.*, 2014, vol. 38, no. 9–10, pp. 2398–2413.
7. Tormagov, T.A., Generalov, A.A., Shavin, M.Y., and Rapoport, L.B., Motion Control of Autonomous Wheeled Robots in Precision Agriculture, *Gyroscopy Navig.*, 2022, vol. 13, no. 1, pp. 23–35.
8. Chichkanov, I. and Shavin, M., Algorithm for Finding the Optimal Obstacle Avoidance Maneuver for Wheeled Robot Moving Along Trajectory, *2022 16th International Conference on Stability and Oscillations of Nonlinear Control Systems (Pyatnitskiy's Conference)*, Moscow: IEEE, 2022, pp. 1–3.
9. Latombe, J.-C., *Robot Motion Planning*, Boston, MA: Springer US, 1991.
10. Choset, H. and Pignon, P., Coverage Path Planning: The Boustrophedon Cellular Decomposition, *Field and Service Robotics*, Zelinsky, A., Ed., London: Springer London, 1998, pp. 203–209.
11. Acar, E.U., Choset, H., Rizzi, A.A., et al., Morse Decompositions for Coverage Tasks, *Int. J. Rob. Res.*, 2002, vol. 21, no. 4, pp. 331–344.
12. Hart, P., Nilsson, N., and Raphael, B., A Formal Basis for the Heuristic Determination of Minimum Cost Paths, *IEEE Trans. Syst. Sci. Cybern.*, 1968, vol. 4, no. 2, pp. 100–107.

13. Stentz, A., Optimal and Efficient Path Planning for Unknown and Dynamic Environments, *Int. J. Robot. Autom.*, Int. Association of Science and Technology for Development, 1995, vol. 10, no. 3, pp. 89–100.
14. Chuang, J.-H., Potential-Based Modeling of Three-Dimensional Workspace for Obstacle Avoidance, *Proceedings IEEE International Conference on Robotics and Automation. IEEE Comput. Soc. Press*, 1993, pp. 19–24.
15. Gilimyanov, R.F. and Rapoport, L.B., Path Deformation Method in Robot Motion Planning Problems in the Presence of Obstacles, *Problemy Upravleniya*, 2012, no. 1, pp. 70–76.
16. Gilimyanov, R.F. and Rapoport, L.B., Path Deformation Method for Robot Motion Planning Problems in the Presence of Obstacles, *Autom. Remote Control*, 2013, vol. 74, no. 12, pp. 70–76.
17. Lobo, M.S., Vandenberghe, L., Boyd, S., et al., Applications of Second-Order Cone Programming, *Linear Algebra Appl.*, 1998, vol. 284, no. 1–3, pp. 193–228.
18. Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge: Cambridge University Press, 2004.
19. O’Donoghue, B., Chu, E., Parikh, N., et al., Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding, *J. Optim. Theory Appl.*, 2016, vol. 169, no. 3, pp. 1042–1068.
20. Domahidi, A., Chu, E., and Boyd, S., ECOS: An SOCP Solver for Embedded Systems, *2013 European Control Conference*, 2013, pp. 3071–3076.
21. Vahdanjoo, M., Zhou, K., and Sørensen, C.A.G., Route Planning for Agricultural Machines with Multiple Depots: Manure Application Case Study, *Agronomy*, 2020, vol. 10, no. 10, p. 1608.
22. Conesa-Muñoz, J., Bengochea-Guevara, J., Andújar, D., et al., Route Planning for Agricultural Tasks: A General Approach for Fleets of Autonomous Vehicles in Site-Specific Herbicide Applications, *Comput. Electron. Agric.*, 2016, vol. 127, pp. 204–220.
23. Dierckx, P., An Algorithm for Surface-Fitting with Spline Functions, *IMA J. Numer. Anal.*, 1981, vol. 1, no. 3, pp. 267–283.
24. Cressie, N., The Origins of Kriging, *Math. Geol.*, 1990, vol. 22, no. 3, pp. 239–252.
25. Pesterev, A.V. and Gilimyanov, R.F., Path Planning for a Wheeled Robot, *Trudy ISA RAS*, 2006, no. 25, pp. 205–212.
26. Diamond, S. and Boyd, S., CVXPY: A Python-Embedded Modeling Language for Convex Optimization, *J. Mach. Learn. Res.*, 2016, vol. 17, no. 83, pp. 1–5.
27. Agrawal, A., Verschueren, R., Diamond, S., et al., A Rewriting System for Convex Optimization Problems, *J. Control Decis.*, 2018, vol. 5, no. 1, pp. 42–60.
28. Junjie, P. and Dingwei, W., An Ant Colony Optimization Algorithm for Multiple Travelling Salesman Problem, *First International Conference on Innovative Computing, Information and Control – Volume I (ICICIC’06). IEEE*, 2006, vol. 1, pp. 210–213.

This paper was recommended for publication by L.B. Rapoport, a member of the Editorial Board