DOI: http://doi.org/10.25728/cs.2024.3.5

A LOCAL PATH PLANNING ALGORITHM FOR AVOIDING OBSTACLES IN THE FRENET FRAME

M. I. Makarov

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

✉ maxim.i.makarov@gmail.com

Abstract. This paper presents a local path planning algorithm in the coordinate system of the roadbed. The algorithm is based on varying initial trajectory points using the potential field method and ensuring the smooth resulting path in a new coordinate system. This algorithm is executed by minimizing an objective functional. The problem is solved with application to path planning for an unmanned transport platform: it is necessary to change the vehicle's global smooth trajectory points in real time while maintaining smoothness and avoiding emerging obstacles. Compared to the Cartesian coordinate system, the new coordinate system is advantageous in terms of the execution time of the algorithm. The algorithm is implemented in Python. With a planning horizon being specified, this approach can be combined with various path-following algorithms having no obstacle avoidance methods. Computer simulation results are provided to demonstrate the effectiveness of the proposed algorithm.

Keywords: path planning, Frenet coordinate system, unmanned vehicles.

INTRODUCTION

With recent advances in lidar technology, high-performance GPUs, and machine learning, autonomous technologies are undergoing revolutionary changes resulting in new capabilities. Much attention is paid to the development of unmanned transport platforms, which leads to various fundamental and applied problems. Path planning is one problem of this class. Here, it is required to build, using environment data from various sensors, a locally optimal path to avoid obstacles in real time while staying within the roadbed boundaries on the original route.

In robotics, only global path planning approaches are actively adopted: the robot's target route is built in advance. In such cases, onboard control laws are responsible for obstacle avoidance [1]. These approaches become inapplicable due to their computational complexity when dealing with large vehicles (e.g., cars) and new conditions (lane width restrictions, dynamic obstacles, and high speeds). A modern approach to overcoming the drawbacks described above is to decompose the problem into local path planning and motion control along a given path. Local planning considers environment data received from sensors to build a locally optimal obstacle avoidance path with all required constraints. Then the control problem is solved to ensure local path following with minimum displacements.

Local path planning in the Cartesian coordinate system is not very convenient due to difficulties in describing the mutual arrangement of the vehicle, roadbed, and emerging obstacles. The Frenet coordinate system (also called the Frenet frame) was conceptualized in [2]. In this system, the vehicle's position is defined as the path traveled along a predetermined smooth (reference) curve and the transverse displacement relative to it, which better and more clearly describes the vehicle's maneuvering capabilities within the roadbed boundaries. The transverse displacement is varied with a discrete step, which is excellent for solving motion control problems on motor roads with pre-designated lanes. In the cases of no lanes (motion is possible anywhere on the roadbed), the local planning problem becomes computationally intensive as the displacement step is small and many admissible trajectories are generated for each step. Other approaches devoid of this drawback, such as those relying on control with predictive models [3], require reproducing a numerical model of the controlled vehicle as accurately as possible. This is not always easy in practice, especially if different vehicles are used. In addition, the cumbersome recalculation process is permanently restarted to ensure accurate path following, even if no new objects appear on the path.

In this paper, we propose an approach to local path planning in the Frenet frame without the disadvantages mentioned above. Solving the related optimization problem using a penalty function allows avoiding obstacles while maintaining the smooth resulting path and reducing the discretization step of the transverse displacement to the accuracy required for the real-time application of the algorithm.

1. TRANSITIONS BETWEEN COORDINATE SYSTEMS

1.1. The Frenet Frame

The Frenet frame specifies the position of an object in the 2D space relative to a reference curve using two coordinates (s, d), where s is the arc length from the origin of the reference curve (the longitudinal displacement) and d is the transverse displacement relative to the same curve (Fig. 1). In other words, we pass to a coordinate system associated with the roadbed, where s is the roadbed length from the starting point to the current point and d is the displacement relative to its center. In the figure, the car's position in the Frenet frame is given by the coordinates (2, 1). When moving along the path a_1, the coordinate d will remain invariable: $d = 1$.

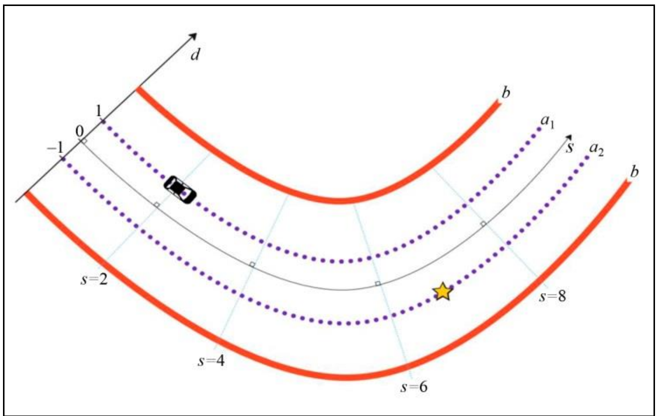

Fig. 1. The object's position on the roadbed in the Frenet frame: s—the reference curve forming one of the axes; d—the axis of transverse displacement relative to the reference curve; b—roadbed boundaries; a_1 and a_2—paths with constant coordinates $d = 1$ and $d = -1$, respectively.

For the transition to the Frenet frame, it is necessary to define a reference curve determining this coordinate system. Consider a smooth and thrice continuously differentiable curve in the Cartesian coordinate system with a natural parameterization $p = p(s)$. There exists a mapping f that defines a transformation of the radius vector of the curve points in the Cartesian coordinate system to the parameter s.

1.2. The Transition from the Cartesian Coordinate System to the Frenet Frame

In the Cartesian coordinate system, the position of any point of the original curve is given by the radius vector $\vec{R}_p$ (Fig. 2). The position of the curve point is reconstructed by the parameter s.

An arbitrary point P with a radius vector $\vec{r}$ is obtained by the composition of vectors $\vec{R}_p$ and $\vec{d}_p$, where $\vec{d}_p$ is perpendicular to the tangent to the reference curve at the point s_p nearest to P.

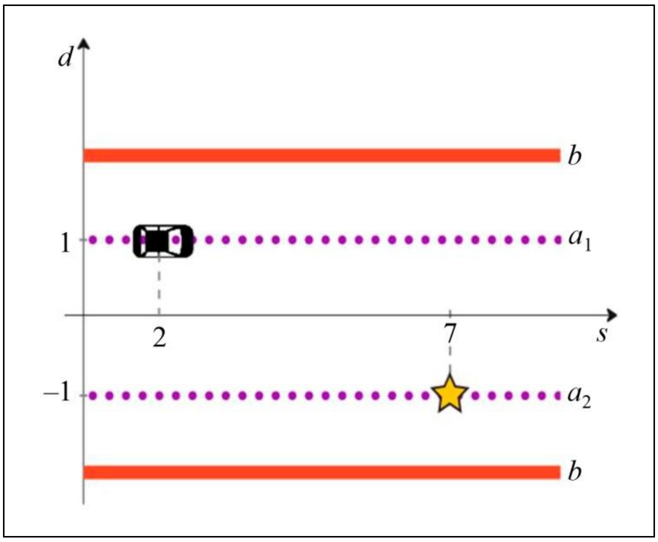

Fig. 2. The roadbed (Fig. 1) represented in the Frenet frame: s—the reference curve defining a new coordinate axis; d—the normal to the reference curve (the second axis); b—roadbed boundaries; a_1 and a_2—the paths with the constant coordinates $d = 1$ and $d = -1$, respectively.

We introduce the following angles: $\theta_s(\vec{R}_p)$ as the inclination of the tangent to the reference curve in the Cartesian coordinate system at the point S_p (Fig. 3) and θ_d as the inclination of the vector $\vec{d}_p$. Then the transition to the Frenet frame is performed by

$$s_p = f\left(\vec{R}_p\right),$$

$$d_p = \operatorname{sign}(\theta_s - \theta_d)\left|\vec{d}\right|.$$

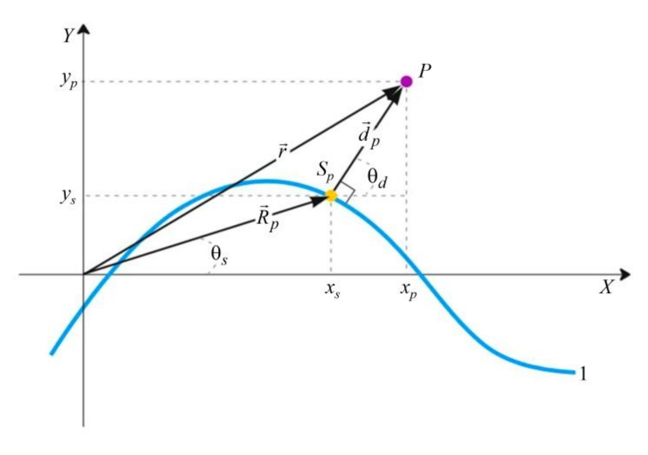

Fig. 3. Transition between the Cartesian coordinate system and the Frenet frame: *P*—an arbitrary point in the space; 1—the reference curve; S_p—the point nearest to *P* on the reference curve.

1.3. Transition from the Frenet Frame to the Cartesian Coordinate System

Since the resulting path must be represented in the Cartesian coordinate system, we consider the inverse transition as well. This transition is performed by determining the coordinates of the reference curve point in the Cartesian coordinate system from the parameterized curve equation and adding the transverse displacement $\vec{d}$:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} x_s \\ y_s \end{bmatrix} + \left|\vec{d}_p\right| \begin{bmatrix} -\sin\theta_d \\ \cos\theta_d \end{bmatrix}.$$

2. HORIZON CONSTRUCTION IN THE FRENET FRAME

A horizon is an ordered fixed-size set of coordinates representing some segment of a global path for which local planning is performed at the current time. The horizon length is determined by the range of external sensors transmitting data about external objects and by the computing power of the computer. The Frenet frame turns out to be convenient for horizon calculation. In this coordinate system, all objects are defined by their position on the roadbed: it is much easier to determine whether an obstacle lies on the current trajectory, including its curved segments. In the new orthogonal coordinate system, the horizon is approximated by cubic B-splines.

It is required to construct a horizon ensuring the avoidance of static obstacles. Obstacle avoidance and B-spline curve smoothing by varying the points in an arbitrary direction in the original Cartesian space were considered in [4]. Utilizing these results, we vary the spline points in the resulting new coordinate system. Moreover, to reduce the number of variables, it suffices to restrict the variation to the axis *d* only. The *i*th point in the Frenet frame has the position $q_i = \left(s_i,\, d_i^0 + d_i\right) = \left(s_i,\, \tilde{d}_i\right)$, where d_i is the variation of the transverse displacement of the *i*th coordinate relative to the reference curve.

According to [4, 5], the curvature of the curve in the new coordinate system is decreased by minimizing the sum of the squared norms of the vectors $(\tilde{d}_i - \tilde{d}_{i-1})$ and $(\tilde{d}_{i-1} - 2\tilde{d}_i + \tilde{d}_{i+1})$, which define the first and second derivatives of the curve function, respectively. This sum can be written as

$$S\left(\tilde{d}_h\right) = \tilde{d}_h^{\mathrm{T}}(H_1 + H_2)\tilde{d}_h, \tag{1}$$

where

$$H_1 = C_1^{\mathrm{T}} C_1, H_1 \in R^{n\times n},\ H_2 = C_2^{\mathrm{T}} C_2, H_2 \in R^{n\times n},$$

$$\tilde{d}_h = \tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_n,$$

$$C_1 = \begin{bmatrix} 1 & -1 & 0 & 0 & \ldots \\ 0 & 1 & -1 & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{bmatrix}, C_1 \in R^{(n-1)\times n},$$

$$C_2 = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & \ldots \\ 0 & 1 & -2 & 1 & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \end{bmatrix}, C_2 \in R^{(n-2)\times n}.$$

To control the mobility of individual points, we introduce the following penalty functional for point variations:

$$P(d_h) = d_h^{\mathrm{T}} D d_h,\ D \in R^{n\times n}, \tag{2}$$

where $d_h = d_1, d_2, \ldots, d_n$ is the transverse variation of the horizon points and *D* is a diagonal matrix with elements proportional to the variation penalty of the corresponding point.

3. REPULSIVE POTENTIAL

This paper considers point obstacles only. They are avoided by creating a potential field around the object in the Frenet frame. Each object O_j in this coordinate system is defined by the two coordinates $\left(s^{(j)}, d^{(j)}\right)$. Each point of the planned horizon *h* is also defined by the two coordinates $\left(s_i, d_i\right)$ $(s_h = s_1, s_2, \ldots, s_n)$. Then the potential over all objects from all points of the horizon is given by

$$U_{\text{hor}}(s_h, \tilde{d}_h) = \frac{1}{2}\eta\sum_{j=1}^{n_{\text{obj}}}\sum_{i=1}^{n}\exp\left(-(s^{(j)}-s_i)^2+(d^{(j)}-\tilde{d}_i)^2\right), \quad (3)$$

where $\eta > 0$ and n_{obj} is the number of obstacles detected.

Thus, the farther the horizon points are from the detected obstacles, the smaller the magnitude of the potential field will be.

4. THE OBJECTIVE FUNCTIONAL

The penalty functions for increasing the curvature and length of the path and approaching an obstacle have been introduced above. In most applications-relevant problems, it is also necessary to consider the roadbed boundaries by limiting the admissible variation range of the parameter *d*. Moreover, individual boundaries can be specified for each point, which is especially important under different roadbed widths. In the case of unknown roadbed boundaries, a penalty is introduced for displacements from the reference curve to ensure the motion near the original global path:

$$M\left(d_h\right) = d_h d_h^{\text{T}}. \quad (4)$$

In view of the expressions (1)–(4), the objective functional takes the form

$$\Phi_{\text{hor}}\left(s_h, d_h\right) = \frac{1}{2}S\left(d_h^0 + d_h\right) + \frac{1}{2}P\left(d_h\right) + U_{\text{hor}}\left(s_h, d_h^0 + d_h\right) + \gamma M\left(d_h^0 + d_h\right), \quad (5)$$

where $\gamma \geq 0$ and $d_h^0 = d_1^0, d_2^0, ..., d_n^0$ are the initial displacements of the selected horizon relative to the reference curve. If the horizon points coincide with those of the reference curve, then $d_j^0 = 0 \ \ \forall j \in 1,...,n$.

The horizon is found by solving the optimization problem

$$d_h^* = \text{argmin}_{\{d_1,...,d_n\}}\Phi_{\text{hor}}\left(s_h, d_h\right), \quad D_i^l \leq \left(d_i^0 + d_i\right) \leq D_i^r, \ d_h^* \in R^n, \quad (6)$$

where $D_i^r \in R^n$ and $D_i^l \in R^n$ are the distance matrices from each point of the reference curve to the nearest point on the right- and left-hand boundary of the roadbed, respectively.

Solving this minimization problem may require knowledge of the gradient of the objective functional (5). Since only the transverse displacements are varied, the corresponding gradients have the form

$$\nabla S\left(\tilde{d}_h\right) = 2\left(H_1 + H_2\right)\tilde{d}_h,$$

$$\nabla \text{P}\left(d_h\right) = 2\text{P}d_h,$$

$$\nabla \text{M}\left(\tilde{d}_h\right) = 2\tilde{d}_h,$$

$$\nabla U_{\text{hor}}\left(s_h, \tilde{d}_h\right) = -\sum_{j=1}^{n_{\text{obj}}}\sum_{i=1}^{n}\eta\left(d^{(j)} - d_i\right) \times \exp(-(s^{(j)} - s_i)^2 + \left(d^{(j)} - \tilde{d}_i)^2\right).$$

5. THE HORIZON CALCULATION ALGORITHM

The main idea of this algorithm is to use the horizon obtained at the previous iteration as an initial approximation, transfer the initial coordinates to the current position, and extend the path to the desired length (Fig. 4). By a horizon *H* we understand an ordered set of vectors $h_i = \left(s_i, \tilde{d}_i\right)$:

$$H = \begin{bmatrix} h_1^{\text{T}} & h_2^{\text{T}} & \ldots & h_N^{\text{T}} \end{bmatrix}^{\text{T}} \in R^{2\times n}.$$

Initialization. Input a given array of *N* points forming a smooth reference curve with an equal distance Δs between neighbor points [4]. For the *i*th point of this array, the displacement along the reference curve is given by $s = i\,\Delta s$.

Step 1. Obtain the vehicle's position in the Cartesian coordinate system.

Step 2. Examine all points of the reference curve to find the number *p* of the point nearest to the vehicle's position and the distance d_p to this point: $\left(s_p,\ d_p\right) = \left(p\,\Delta s,\ d_p\right)$.

Step 3. Initialize the horizon

$$H = \begin{bmatrix} s_p & s_p + \Delta s & \ldots & s_p + \left(n-1\right)\Delta s \\ d_p & d_p & \ldots & d_p \end{bmatrix}^{\text{T}} \in R^{2\times n},$$

which characterizes the motion along the reference curve while maintaining the initial transverse displacement.

Step 4. Repeat Steps 1 and 2.

Step 5. Consider two possible situations:

- The displacement along the reference curve lies outside the current horizon, i.e., the current displacement s_p is not contained in the matrix *H*. Then repeat Step 3.
- The vehicle's position lies inside the current horizon, i.e., the current displacement s_p is contained in the matrix *H*.

Let m be the number of the horizon coordinate s corresponding to the displacement s_p. Remove the first rows of the horizon up to the $(m – 1)$th row inclusive, thereby transferring the array's beginning to the row corresponding to the coordinate s_p. Complement the horizon to the required length by adding the necessary coordinates s to the end, continuing the remaining row with the given step Δs and pairing the last known transverse displacement.

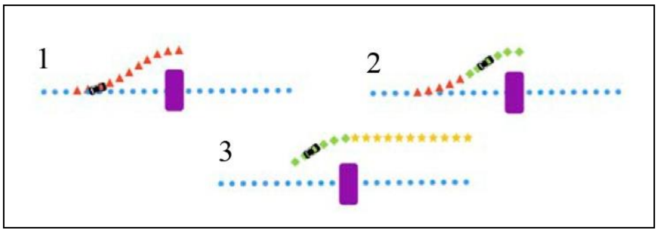

Fig. 4. The initial approximation process for the horizon: circles—reference trajectory; triangles—the horizon obtained at the previous step of the algorithm; rhombuses—the points transferred to the beginning of the horizon; asterisks—the points complementing the horizon to the required length.

Step 6. Solve the minimization problem (6) using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm.

Step 7. Transform the obtained coordinates of the horizon H back to the Cartesian coordinate system. Supply them to the input of the path-following algorithm.

Step 8. Repeat Steps 4–7 of the algorithm until reaching the end of the reference curve. To save computing resources, a new iteration of the algorithm is executed when facing a new obstacle or approaching the end of the current horizon, $0.1n$ points before the end.

6. A NUMERICAL EXAMPLE

For the generated horizon, the path planning algorithm was tested using the path-following algorithm described in [6]. The reference curve was specified by points from a sparse set of GNSS receiver coordinates recorded during the vehicle's motion, which were transformed to the UTM system and smoothed by the method presented in [4]. Here, $D \in R^{n\times n}$ is a diagonal matrix with values {25, 25, 0, 0} on the principal diagonal.

Figure 5 shows the resulting path built for the parameter values $\gamma = 0.5$, $\eta = 2$ and the penalty for displacements from the reference curve [3]. Note that the roadbed width constraints were neglected. Obviously, the trajectory [1] tends to return to the reference curve at all time instants.

The case in Fig. 6 corresponds to a smaller value of the reference curve approach coefficient and the road width constraints $\gamma = 0.001$, $\eta = 2$; here, $D \in R^{n\times n}$ is a diagonal matrix with values {25, 25, 0, 0} on the principal diagonal. In this case, the vehicle realigned to avoid the obstacles and continued its motion at some distance from the reference curve.

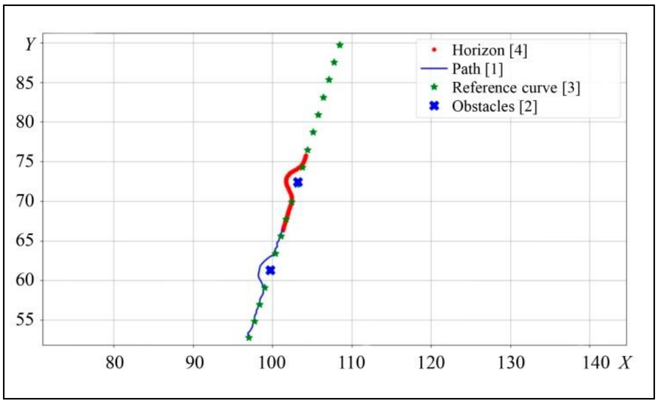

Fig. 5. The vehicle's path [1] along the initial points of the reference curve [3] with bypass of point obstacles [2] and planned horizon [4] in the UTM coordinates.

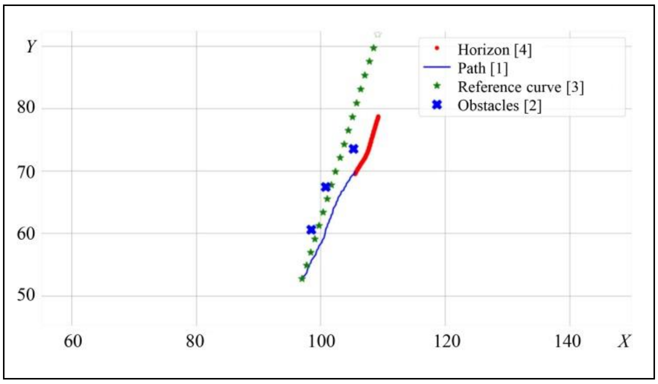

Fig. 6. The vehicle's path [1] along the initial points of the reference curve [3] with bypass of point obstacles [2] and planned horizon [4] in the UTM coordinates.

CONCLUSIONS

This paper has considered local path planning for unmanned vehicles in the Frenet frame with point obstacles. The existing local path planning approaches, such as potential field-based and predictive control model (PCM) methods, either have insufficient flexibility in maneuvers or require significant computing resources for accurate path prediction. Moreover, the existing local planning methods neglect the smoothness of the curve being built.

We have proposed a local horizon planning algorithm in the Frenet frame with the effective avoidance of point obstacles. Unlike traditional methods in the Cartesian coordinate system, this algorithm varies the transverse displacement along the reference curve in the Frenet frame, which improves computing efficiency and simplifies maneuver processing.

According to the numerical examples above, the algorithm builds a smooth path in the presence of multiple obstacles in real time, demonstrating accuracy and computational simplicity. The algorithm can be integrated with various path-following algorithms having no obstacle avoidance methods.

REFERENCES

1. Rapoport, L.B., and Gilimyanov, R.F., Path Deformation Method for Robot Motion Planning Problems in the Presence of Obstacles, *Automation and Remote Control*, 2013, vol. 74, no. 12, pp. 2163–2172.
2. Basavanna, M. and Shivakumar, M., An Overview of Path Planning and Obstacle Avoidance Algorithms in Mobile Robots, *International Journal of Engineering Research & Technology*, 2019, vol. 08, no. 12.
3. Lu, B., Li, G., Yu, H., et al., Adaptive Potential Field-Based Path Planning for Complex Autonomous Driving Scenarios, *IEEE Access*, 2020, vol. 8, pp. 225294–225305. DOI: 10.1109/ACCESS.2020.3044909.
4. Werling, M., Ziegler, J., Kammel, S., and Thrun, S., Optimal Trajectory Generation for Dynamic Street Scenarios in a Frenet Frame, *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, 2010, pp. 987–993. DOI: 10.1109/ROBOT.2010.5509799.
5. 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, *Journal of Computer and Systems Sciences International*, 2008, vol. 47, no. 5, pp. 812–819.
6. Thrun, S., Montemerlo, M., Dahlkamp, H., et al., Stanley: The Robot That Won the DARPA Grand Challenge, in *Springer Tracts in Advanced Robotics*, Buehler, M., Iagnemma, K., and Singh, S., Eds., Berlin–Heidelberg: Springer, 2007, vol. 36, pp. 1–43.

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

Received April 17, 2023,
and revised May 13, 2024.
Accepted May 21, 2024.

Author information

Makarov, Maxim Igorevich. Engineer, Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ maxim.i.makarov@gmail.com
ORCID iD: https://orcid.org/0000-0002-4854-5910

Cite this paper

Makarov, M.I., A Local Path Planning Algorithm for Avoiding Obstacles in the Frenet Frame. *Control Sciences* **3**, 56–61 (2024). http://doi.org/10.25728/cs.2024.3.5

Original Russian Text © Makarov, M.I., 2024, published in *Problemy Upravleniya*, 2024, no. 3, pp. 66–72.

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com