MOBILE ROBOT OPTIMAL PATH PLANNING VIA SPLINE APPROXIMATION OF RRT* GENERATED POSITIONS

Yuriy ROMASEVYCH[1*], Viatcheslav LOVEIKIN[2], Viktor KRUSHELNYTSKYI[3]

Nowadays, mobile robots are commonly used in various areas of production, transportation, and domestic applications. During their exploitation, there is a large number of cases when the environment contains static obstacles. The design of smooth and optimal robot paths for such cases is a studied issue of a huge number of academic and practical works. In the current investigation, this problem includes the requirement of path smoothness and its minimal length. To solve it an approach was developed. Its initial stage involves Rapidly-Exploring Random Tree star (RRT) algorithm application and dilution of the obtained points. In the second stage, a cubic spline is used. Its knots are found as shifted points of the diluted path (the first stage output). To ensure a collision-free path, which passes only through the bounded domain two penalty functions were developed. Thus, the initial problem was reduced to the unconstrained optimization problem. It was solved via Varying Cognitive Term – Particle Swarm Optimization (VCT-PSO) method for six different robot environment scenarios. Analysis of the obtained paths showed their high smoothness and shorter (compared to RRT* algorithm) length. Further directions for the approach development were given in the final part of the study.*

Keywords: RRT* algorithm, cubic spline, mobile robot, path planning, obstacles avoidance.

1. Introduction

Mobile robots are commonly used in modern agricultural production [1], logistics [2, 3], domestic [4], and military [5] applications, etc. Planning of mobile robot path is an important issue, which may improve dynamical, energetical features of robot exploitation, its productivity, and a level of safety. There is a huge amount of scientific works dedicated to this problem and it is worth mentioning a variety of involved methods and approaches to solve it. More recent publications in this sphere rely on artificial neural networks approximation

* Corresponding author
[1] Prof., Dept. of Machine and Equipment Design, National University of Life and Environmental Sciences of Ukraine, Ukraine, e-mail: romasevych@nubip.edu.ua
[2] Prof., Dept. of Machine and Equipment Design, National University of Life and Environmental Sciences of Ukraine, Ukraine, e-mail: loveykin@nubip.edu.ua
[3] Assoc. Prof., Dept. of Machine and Equipment Design, National University of Life and Environmental Sciences of Ukraine, Ukraine, e-mail: krushelnytskyiviktor@nubip.edu.ua

features, fuzzy logic, and metaheuristic optimization [6]. However, RRT* algorithm remains the dominant sampling-based planner, with many hybridizations and modifications [7-11]. The reason is connected with the ability of sampling-based planners to generate solutions in high-dimensions spaces and asymptotical optimality of some algorithms (RRT* and its modifications) and quite low computational needs. However, kinematical and dynamical constraints are still difficult issues to handle for sampling-based planners.

The exploitation of metaheuristic optimization techniques – is a notable feature of the path planning problem solutions [2, 6, 12]. Their wide application may be explained by the complex topology of objective functions (most commonly is the length of the path), which are associated with the problem solution, and the need for a global minimum localization. There are a lot of published works addressed the combination of metaheuristic optimized with path planner algorithms (gray wolf optimization and A* [13], enhanced particle swarm optimization (PSO) and basis splines (B-splines) [14]), as well as modification of metaheuristics to enhance their search efficiency for the problem of path planning (bi-population PSO algorithm with a random perturbation strategy [15], variable-length differential evolution [16]) etc. These works have clearly identified the improved computation efficiency of the path planning algorithms compared to the existing ones, as well as the possibility of better solution reaching (lesser path length, reduced number of robot turns, reduced energy consumption, etc.).

To make the path smooth scholars used splines, particularly B-splines [17] and their combination with RRT algorithm [18]. Postprocessing with cubic or B-splines transforms a discrete path, generated with a sampling-based planner, into a smooth, differentiable (given by C^1/C^2-continuity) path, which is easier to track by the controller and safer for the actuators. This general observation is supported by several experimental works and reviews in recent years [19-21].

Similar approach is used in the current study. Despite huge scientific investigations in this field, there is no computationally effective and stable algorithm for the problem solution. The results presented in the scientific works lacked with variety of robot environment scenarios. Some paths are impossible to implement in practice because of a lack of smoothness. These issues were addressed in the current study. The concept of the current study concludes in the appropriate and consistent application of different methods, which include classical ones (RRT*, cubic spline), state-of-the-art optimization techniques, and design of proper objective function.

2. The statement of mobile robot optimal path planning problem

The statement of the problem includes four items: the configuration of the robot movement domain; descriptions of obstacles to avoid; the coordinates of start and goal points; the objective function to minimize. In the current study, we addressed one of the most common problem statement [7-21]:

$$\begin{aligned} &\min_{\mathrm{x}(t)} L(\mathrm{p}(t)) = \int_0^T \|\dot{\mathrm{p}}(t)\| dt; \\ &s.t. \quad \mathrm{p}(0) = \mathrm{p}_s; \ \mathrm{p}(T) = \mathrm{p}_g; \\ &\qquad \mathrm{p}(t) \in D, \ \forall t \in [0, T]; \\ &\qquad \mathrm{p}(t) \notin \mathrm{O}, \ \forall t; \\ &\qquad \mathrm{p}(t) \in C^2, \end{aligned} \tag{1}$$

where $L(\mathrm{p}(t))$ – the length of the path; $\mathrm{p}(t)$ – the path of the mobile robot movement, which is a vector function of time, i.e. $\mathrm{p}(t)=(x(t), y(t))$; $x(t)$ and $y(t)$ – mobile robot coordinates along x and y axes as a functions of time; t – time; T – duration of the robot movement (in the current study a constant value); p_s and p_g – vectors of start (initial) and goal (final) points of the path; D – the domain of the mobile robot movement, which is presented as a square $D = (x, \ y), \ \forall x \in [x_{lb}, x_{lb}], \ \forall y \in [y_{lb}, y_{lb}]$ (in the current study $x_{lb}=y_{lb}=0$, $x_{ub}=y_{ub}=1$); O – the set of obstacles to avoid, which may be presented as follows:

$$\mathrm{O} = O_i, \ \forall i \in [1, \ I], \ i \in \mathrm{Z}^+, \tag{2}$$

where I – the total number of obstacles.

A dot above character means a time derivative.

The problem (1), (2) solution (the optimal path) connects the start p_s and goal p_g points with a smooth curve of minimal length, it does not exceed domain borders and any part of the optimal path does not belong to any obstacles region O.

3. Approach development

The scheme of the problem solving includes two following stages: calculation of spline knots coordinates, and development of a spline with adjusted knots. To design the knots, a common algorithm may be used. In the current, study we used RRT* algorithm – it gives the close to minimal-length path, but it contains too many points for further spline approximation. In addition, RRT* generated path points do not form the proper path in terms of length. First-order approximation does not satisfy the last problem condition (1), while higher-order approximations are not optimal in terms of path length. These circumstances force us to process the RRT* generated path points before approximating them with a cubic spline. The postprocessing means dilution of RRT* generated path points. The number of obtained points must be lesser, than the number of initial path points. However, not all the points may be skipped. The key points, which are close to obstacles corners (where the path suppose to change its direction), start and goal points must be preserved. The coordinates of diluted path points must be adjusted to form the spline knots. Here we must stress, that during such an adjusting procedure one or more conditions (3rd and 4th lines of the problem statement (1)) may be violated. To satisfy them penalty functions must be developed and used in the adjusting procedure.

Let's give strict expressions to explain the essence of the approach. Firstly, we apply RRT* algorithms to obtain a set of points $\mathrm{p}_{\mathrm{RRT*}}$, which belong to the domain D, and do no not belong to obstacles O:

$$\begin{aligned}
&\mathrm{p}_{\mathrm{RRT*}.k} = (x_k,\ y_k),\ \forall k \in [1,\ K];\\
&\mathrm{p}_{\mathrm{RRT*}.1} = \mathrm{p}_s;\ \mathrm{p}_{\mathrm{RRT*}.K} = \mathrm{p}_g;\\
&\mathrm{p}_{\mathrm{RRT*}.k} \in D;\\
&\mathrm{p}_{\mathrm{RRT*}.k} \notin \mathrm{O},
\end{aligned} \tag{3}$$

where x_k and y_k – coordinates of k-th point of the generated with RRT* algorithm path; K – total number of points, generated with RRT* algorithm. The next step is $\mathrm{p}_{\mathrm{RRT*}}$ dilution. The set of points, denoted as $\tilde{\mathrm{p}}_{\mathrm{RRT*}}$, includes the points, that satisfy closeness to the obstacles edges, or each n-th point of $\mathrm{p}_{\mathrm{RRT*}}$:

$$\tilde{\mathrm{p}}_{\mathrm{RRT*}.u} = \mathrm{p}_{\mathrm{RRT*}.k},\ if\ \left\|\mathrm{p}_{\mathrm{RRT*}.k} - O_{i.r}\right\|_2 \le \Delta\ \vee \frac{k}{n} \in \mathrm{Z}^+,\ \forall u \in [1,\ U]; \tag{4}$$

where Δ – the threshold, which helps to split the points according to their closeness to the obstacles' edges (in the current study Δ=0.05); n – the dilution order, which indicates, what number of points from $\mathrm{p}_{\mathrm{RRT*}}$ are mandatory selected for $\tilde{\mathrm{p}}_{\mathrm{RRT*}}$ set; U – total number of diluted path $\tilde{\mathrm{p}}_{\mathrm{RRT*}}$ points. Note, if first order spline is used to approximate the path $\tilde{\mathrm{p}}_{\mathrm{RRT*}}$, then there is no guarantee that obtained path satisfies the last two conditions of the problem (1). Even, if the second- or higher-order spline approximation is used, the condition of obstacles avoidance is still under question. To satisfy them, the spline knots' coordinates should be slightly shifted from the points $\tilde{\mathrm{p}}_{\mathrm{RRT*}}$. Here we mean the shifting along both axes x and y. The values of these shifts are denoted as $\Delta_{u.x}$ and $\Delta_{u.y}$ (subscript indicates the axes). They form a shift-vector $\Delta_u=(\Delta_{u.x},\ \Delta_{u.y})$, its length indicate the overall change of a knot position. Proper selection of vector Δ_u components may satisfy all the problem (1) conditions. 3-dementional case of the path requires three components of the shift-vector Δ_u.

In the following study, we used cubic spline s(p), which satisfies the last condition (1) and includes all the points $\widetilde{\mathrm{p}}_{\mathrm{RRT*}}$. To obtain a discrete path, the spline was discretized, i.e. the discrete path s_j is under consideration. Each point of the path s_j belongs to the spline:

$$\mathrm{s}_j(\widetilde{\mathrm{p}}_{\mathrm{RRT*}.u} + \Delta_u) = (x_j,\ y_j),\ \forall j \in [1,\ J], \tag{5}$$

where J – the total number of the discrete path points; Δ_u – shift-vector of the u-th spline knot coordinate. Note, that such an approach moves spline knots from points $\widetilde{\mathrm{p}}_{\mathrm{RRT*}}$ to slightly different positions along x and y axes.

However, one or more points s_j may violate conditions regarding the domain D and obstacles O avoidance. In order to meet them the following penalty functions were developed:

$$H_D = \delta_D \begin{cases} 0,\ \mathit{if}\ \ \mathrm{s}_j \in D,\ \forall j \in [1,\ J], \\ \sum\limits_{j=1}^{J} \min\left(\left|x_j - x_{lb}\right|,\ \left|x_j - x_{ub}\right|,\ \left|y_j - y_{lb}\right|,\ \left|y_j - y_{ub}\right|\right),\ \mathit{if}\ \exists j\ \mathrm{s}_j \notin D; \end{cases} \tag{6}$$

$$H_i = \delta_O \begin{cases} 0,\ \mathit{if}\ \ \mathrm{s}_j \notin O_i,\ \forall j \in [1,\ J], \\ \sum\limits_{j=1}^{J} \min\left(\left|x_j - O_{i.x_{lb}}\right|,\ \left|x_j - O_{i.x_{ub}}\right|,\ \left|y_j - O_{i.y_{lb}}\right|,\ \left|y_j - O_{i.y_{ub}}\right|\right),\ \mathit{if}\ \exists j\ \mathrm{s}_j \in O_i; \end{cases} \tag{7}$$

where δ_D and δ_O – the weight coefficients, which set the importance of respective condition satisfaction (in the current study δ_D=10^{15} and δ_O=10^{10} – such setting of δ_D and δ_O values stress first of all the need of $\mathrm{s}_j \in D$ condition satisfaction, and secondly obstacle avoidance).

The provided path discretization (5) allows to check every point of the discrete path s_j to make sure that all the problem (1) conditions are satisfied. The sense of presented expressions (6) and (7) is in the following: if one or another condition is violated, then the corresponding penalty function (6) or (7) becomes huge. In this case, an optimizer works to turn penalties to zero, and after that „focus” on minimization of path length. Indeed, if a discrete path s_j point does not belong to D, then the value of the distance from the point s_j to the D closest bound increases the value of H_D penalty. All the D domain violation forms the overall value of H_D (6).

A similar idea is exploited for the handling of obstacle O avoidance (7): if the point s_j coordinate does not satisfy the 4th line of the problem statement (1), then the value of this violation is added to the H_i penalty. Note: the summing must be carried out for each point s_j in relation to each obstacle O_i.

Now we reduced the initial problem (1), (2) to the problem of unconstrained optimization:

$$Cr(\Delta_{u.x}, \Delta_{u.y}) = \frac{\sum_{j=1}^{J-1} \left\| s_j - s_{j+1} \right\|_2 + H_D + \sum_{i=1}^{I} H_i}{\sum_{k=1}^{K-1} \left\| p_{\mathrm{RRT}^*.k} - p_{\mathrm{RRT}^*.k+1} \right\|_2} \to \min, \tag{8}$$

If H_D=0 and H_i=0 ∀ i, then the objective function (8) value shows the ratio of two lengths: obtained via the developed approach and RRT* generated path.

The objective function (8) is a function of a set of arguments $\Delta_{u.x}$, $\Delta_{u.y}$. Their number is twice bigger than the number of spline knots.

In order to illustrate its efficiency, a set of problems were solved.

4. Analysis of the developed approach application

4.1. Description of benchmark mobile robot environment scenarios

In order to estimate the efficiency of the developed approach it should be tested on different scenarios (configurations of obstacles to avoid in the domain). They are presented in Fig. 1. They have similar features in the sizes of the domain – it is a square with a size of a side equals 1 m. The obstacles are modeled as rectangles placed in horizontal and vertical directions in a manner, that forms hard-to-pass configurations. The thickness of the thinnest „walls" equals 0.1 m.

The orange point indicates the start of the mobile robot position, and the black one – the end position (the goal). Each of the scenarios has some difficulties for path planning: for №1 and №3 quit bent path expected to connect start and end points; for №4 and №6 narrow passage between two „walls"; for №2 and №3 quit long sections along the „walls"; for №3 and №5 the need of several time of changing the movement direction to almost opposite ones.

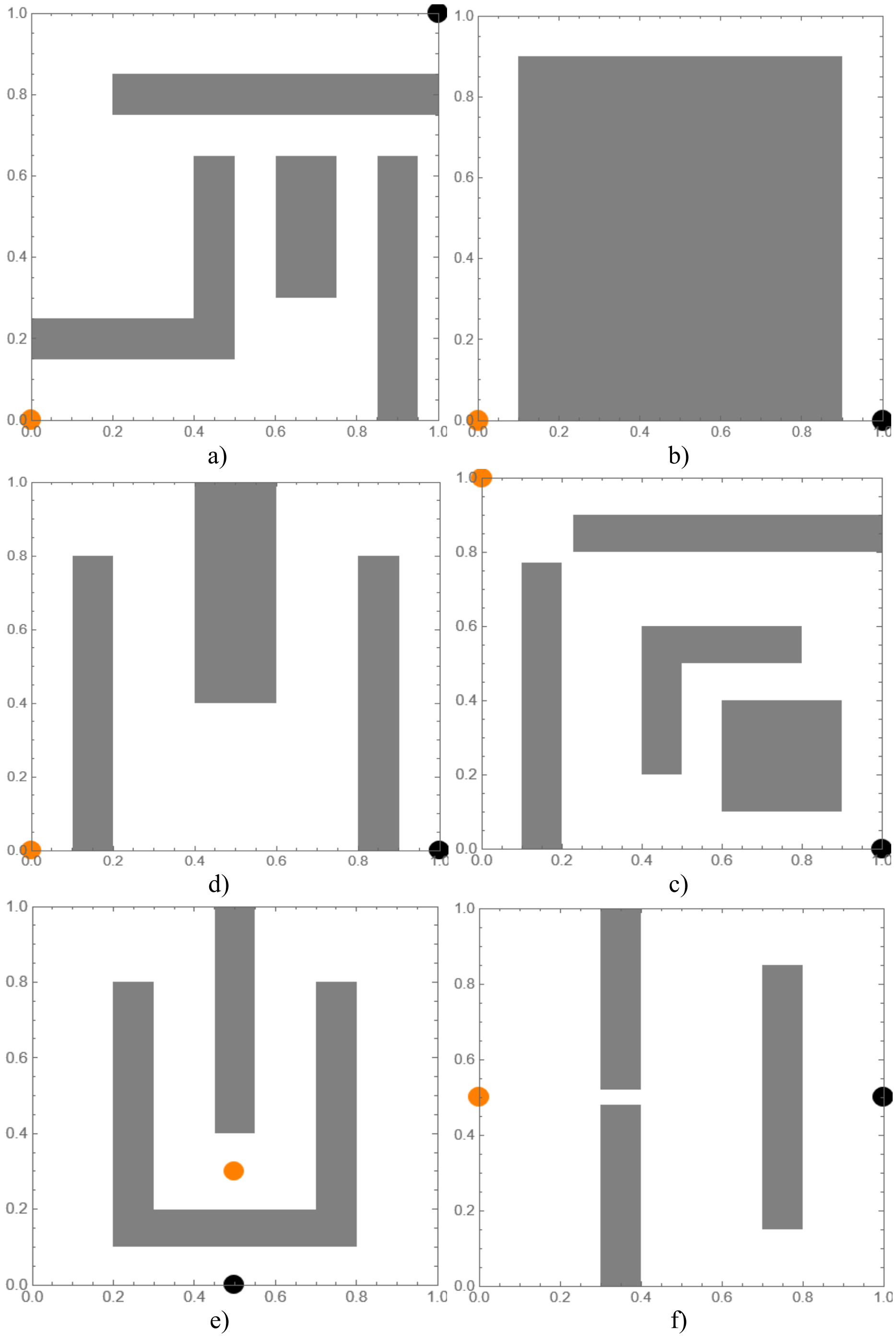


Fig. 1. Scenarios of mobile robot environment to pass: a) №1; b) №2; c) №3; d) №4; e) №5; f) №6

In the next subsections, we give the results of the approach application for described above scenarios. The first thing to focus on is connected with the optimizer convergence because of the fact, that it strongly impacts the appropriate (collision-free and with minimal length) path.

4.2. Analysis of optimization procedure convergence

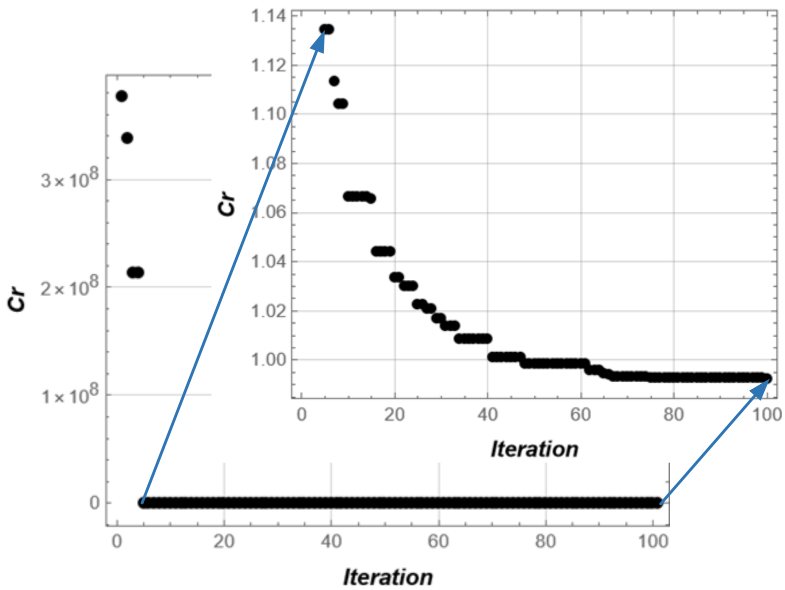
To minimize the objective function (8) a modification of Particle Swarm Optimization (PSO) algorithm – VCT-PSO [22] was applied. The reason why such an algorithm was selected as an optimizer is connected with its strong search features. Indeed, the number of objective function arguments may be quite big, the topology of the objective function (8) is complex, and the penalty functions make the objective function topology non-continuous. All of these require an optimizer, with approved strong features of minimum localization. The parameters of VCT-PSO algorithm are given in Table 1.

Table 1

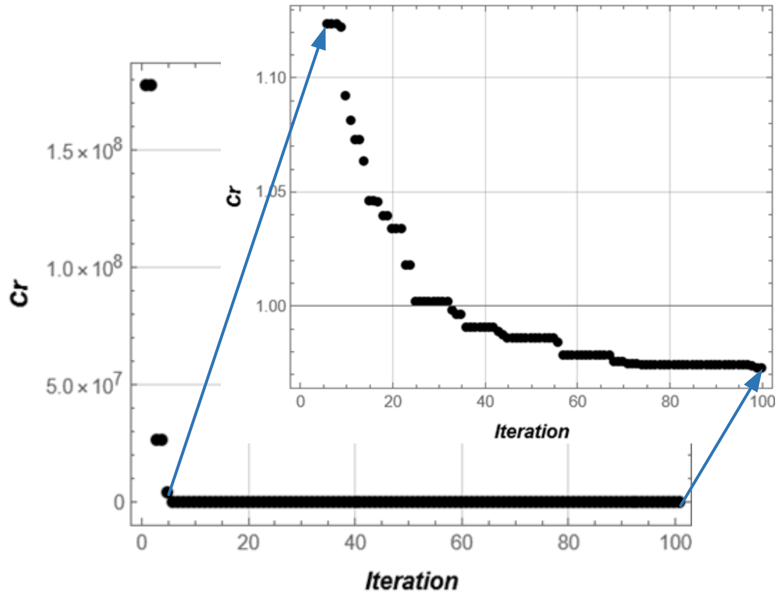
Parameters of VCT-PSO algorithm

Parameter	Parameter value
Number of iterations	100
Swarm population	50
RC	5
w	0.72
c_1 and c_2	1.19

The domain for all the objective function arguments $\Delta_{u.x}$ and $\Delta_{u.x}$ is from -0.1 to 0.1. In Fig. 2 the convergence of VCT-PSO method plots were given for scenarios №1 (Fig. 2, a) and №6 (Fig. 2, b).



a)



b)

Fig. 2. VCT-PSO method convergence for scenarios: a) №1; b) №6

The rest of the optimization procedures indicate similar behaviour. At the early stage of the *Cr* minimization there was a violation of constraints regarding obstacles' collision, and the penalty value (7) was quite big.

After satisfaction of all these constraints, the length of trajectory was the only value to minimize. Note, that for all of the shown cases (Fig. 3) the optimal path is shorter, than that generated with RRT* algorithm – the values of *Cr* at some iteration crossed the value „1" and continued to decrease. A hundred iterations was enough to reach the minimum value of the path length, i.e. the parameters of the VCT-PSO method (Table 1) were reasonably selected.

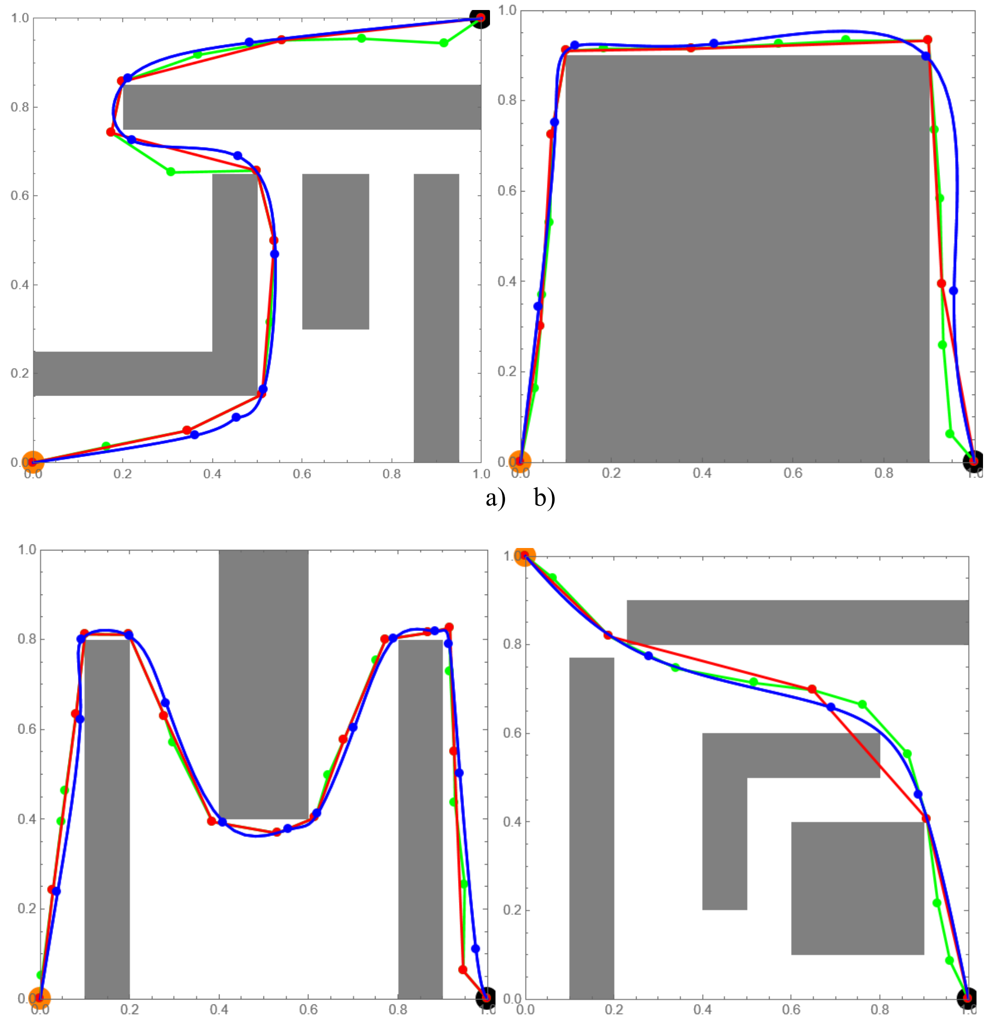
Thus, the proper preparation of points for cubic spline path approximation positively affects the convergence of VCT-PSO method.

4.3. Comparative analysis of obtained paths

All the found paths are illustrated in Fig. 3. The green curves correspond to p_{RRT^*}, the red one – to $\tilde{p}_{RRT^*}$, and blue one – to s_j. The p_{RRT^*}, and $\tilde{p}_{RRT^*}$ trajectory points are presented on Fig. 3 as well; spline knots are shown as blue points.

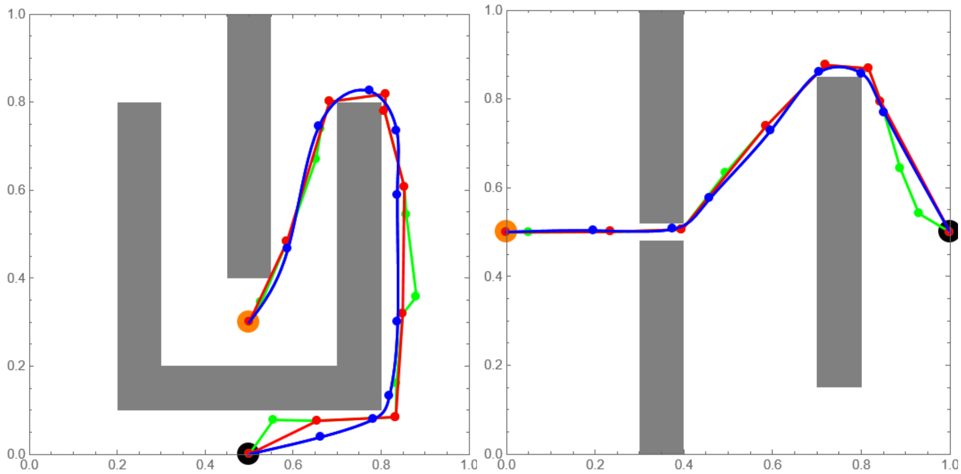
For all cases, the paths are quite close to each other. However, s_j is smooth and this feature makes the corresponding path much more appropriate for mobile robot movement. Indeed, to change the movement direction in points generated by paths p_{RRT*} or $\tilde{p}_{RRT*}$ mobile robot must slow down to complete stop, then yaw during some time.

Opposite, the corresponding to s_j path does not require the mobile robot to stop at knots positions and the average velocity of robot movement may be much higher (remark: in some sections where the path changes direction abruptly only little decrease in robot velocity is needed).



a) b)

d) c)



e) f)

Fig. 3. Mobile robot paths for different scenarios: a) №1; b) №2; c) №3; d) №4; e) №5; f) №6

In order to make the analysis more complete, the numerical indicators of found solutions were calculated (Table 2). Path $\tilde{p}_{RRT*}$ for 4th scenario crosses the obstacle (Fig. 3, d), which makes it completely irrelevant to the problem (1) conditions. This means that naive dilution of points is not applicable for every scenario, and this approach cannot be recommended for application. Analyzing the lengths of the paths one may note, that the corresponded to s_j paths for scenarios № 3 and 5 are slightly shorter, comparing with $\tilde{p}_{RRT*}$ path; for scenarios № 2 and 6 the length of s_j path is the same as $\tilde{p}_{RRT*}$ path; only for № 1 scenario s_j correspond path is 2 mm longer, than the $\tilde{p}_{RRT*}$ corresponded path. Comparing the path s_j and p_{RRT*} corresponded path we note, that the first is 0.77…4.80% shorter.

Table 2

Indicators of found solutions

Scenario number	Number of discrete path points			Path length, m			Violation of constraints for $\tilde{p}_{RRT*}$
	K	U	J	p_{RRT*}	$\tilde{p}_{RRT*}$	s_j	
1	15	10	100	2.348	2.298	2.330	No
2	18	8		2.669	2.647	2.647	No
3	25	16		3.034	3.016	3.003	No
4	12	5		1.584	1.541	1.558	Yes
5	17	11		1.792	1.755	1.710	No
6	12	9		1.414	1.375	1.375	No

In addition, both p_{RRT*} and $\tilde{p}_{RRT*}$ generated paths belong to C^0 function class, while developed in the study – to the C^2 class. This feature may be generalized to higher continuity classes of function (if needed).

Taking everything into account we may state – the problem (1), (2) is solved.

The study has several limitations, which are highlighted below. Firstly, the considered scenarios are static, meaning the obstacles do not change their positions. This is quite a common problem statement, but still, there is some room to enhance the approach developed in the article. Another open issue, which has not been properly studied, is the selection of numbers: *U*, upper and lower bounds of $\Delta_{u.x}$ and $\Delta_{u.x}$ domains. We believe that unreasonable setting of *U* may dramatically deteriorate the quality of the found path, while upper and lower bounds of $\Delta_{u.x}$ and $\Delta_{u.x}$ domains must be set as a ratio of *D* bounds. These issues limit the study's possible generalization.

Further developments of this approach are connected with the exploitation of spline features. The next step in this direction is a calculation of robot movement trajectory – the robot position coordinates as a function of time. The developed above approach may be enhanced with some kinematical or/and dynamical conditions (constraints, criteria to minimize). For instance, the criterion may include two terms: the duration of robot movement, and energy consumed by the robot drives to pass the trajectory. Both terms are influenced by a path, duration of the movement and its velocity profile. Thus, the corresponding objective function will be much more complex in the sense of arguments number and its topology features.

In addition, more complex obstacles must be used in further studies. Real-world obstacles are much complex in terms of their forms and their number. In order to understand the efficiency of the developed approach, the more difficult to handle scenarios will be considered (randomly set balls and walls, multiple narrow tunnels, multiple branching corridors, etc.).

Obtained in the study result (smooth and minimal length robot path) makes the robot movement smooth as well, i.e. without impacts caused by sharp changes in robots' movement direction. This feature is connected with some benefits regarding decreasing of robot drive energy consumption, extension of robot frame and drive lifetime, as well as a lifetime of robot battery. The practical application of the results should be implemented via robot control system (more specific – the planner of robot's movement) for a class of mobile robots.

5. Conclusions

The article proposes an approach to design the mobile robot path in a static bounded environment with multiple obstacles. Initially, it needs RRT* algorithm application and dilution of the obtained points. The outputted points must be slightly shifted from their positions and they are considered as knots of a cubic spline. The values of the shifts are arguments of the objective function to minimize. The latter contains three terms, which correspond to the length of the path, the condition to remain in the domain, and the condition not to collide with the obstacles. To satisfy these two conditions special penalty functions were developed.

The application of the developed approach to six scenarios showed its efficiency. The obtained paths are smooth and 0.77…4.80% shorter, than those generated with RRT* algorithm.

Further directions of the approach development are associated with consideration of robot paths but trajectories, i.e. functions of robot position over time. Supplementing of objective function with kinematical, dynamical, and energetic indicators of robot movement, as well as including the duration of robot movement, and the robot's velocity profile, i.e. are the issues for further consideration.

The fuller class of robot path planners (A*, Probabilistic Roadmap, Potential Field Methods, TrajOpt, Covariant Hamiltonian Optimization for Motion Planning) will be involved in further studies for results comparison.

Acknowledgement

This paper was supported by the HEI-COPILOT project within the EIT HEI Initiative "Innovation Capacity Building for Higher Education", funded by the European Union.

REFERENCES

[1]. *V. Lysenko, I. Bolbot, Y. Romasevych, V. Loveykin, V. Voytiuk.* Algorithms of robotic electrotechnical complex control in agricultural production. Control Systems: Theory and Applications, 2018, pp. 271-289.

[2]. *Z. Haopeng, M. Zhichao, L. Lipeng, W. Yang, Z. Zheyu, L. Hao.* Optimized Path Planning for Logistics Robots Using Ant Colony Algorithm under Multiple Constraints, Computer Science - Robotics, 2025. https://arxiv.org/abs/2504.05339

[3]. *O. Zarivnyi, Y. Romasevych.* Mathematical modeling of the device for transporting small-sized cargo. UPB Scientific Bulletin Series D Mechanical Engineering, **vol. 86(3),** 2024, pp. 87–102.

[4]. C. *Zhang, L. Zhou, Y. Li, Y. Fan.* A Dynamic Path Planning Method for Social Robots in the Home Environment. Electronics. **vol. 9(7):1173**, 2020, pp. 1-18. DOI: 10.3390/electronics9071173

[5]. *J.V.R. Vasconcelos, A.S. Brandão, M. Sarcinelli-Filho.* Real-Time Path Planning for Strategic Missions. Applied Sciences, **vol. 10(21):7773**, 2020, p. 1-13. DOI: 10.3390/app10217773

[6]. *S. Deshpande, A. K. Kashyap, B. K. Patle.* A review on path planning ai techniques for mobile robots. Robotic Systems and Applications, **vol. 3, No. 1**, 2023, pp. 27–46, DOI: 10.21595/rsa.2023.23090

[7]. *L. Zhang, K. Cai, Z. Sun, Z. Bing, C. Wang, L. Figueredo, S. Haddadin, A. Knoll.* Motion Planning for Robotics: A Review for Sampling-based Planners. https://arxiv.org/html/2410.19414v2?utm

[8]. *Z. Huang, H. Chen, J. Pohovey, K. Driggs-Campbell.* Neural Informed RRT*: Learning-based Path Planning with Point Cloud State Representations under Admissible Ellipsoidal Constraints. 2024. Computer Science - Robotics. https://arxiv.org/abs/2309.14595

[9]. *P. Zhang, C. Xiong, W. Li, X. Du, C. Zhao.* Path planning for mobile robot based on modified rapidly exploring random tree method and neural network. International Journal of Advanced Robotic Systems, **vol. 15(3)**, 2018, pp. 1-8, DOI: 10.1177/1729881418784221

[10]. *J. Ding, Y. Zhou, X. Huang, K. Song, S. Lu, L. Wang.* An improved RRT* algorithm for robot path planning based on path expansion heuristic sampling. Journal of Computational Science. **vol. 67**, 2023, 101937. DOI: 10.1016/j.jocs.2022.101937

[11]. *I.-B. Jeong, S.-J. Lee, J.-H. Kim.* Quick-RRT*: Triangular inequality-based implementation of RRT* with improved initial solution and convergence rate. Expert Systems with Applications. **vol. 123**, 2019, pp. 82-90. DOI: 10.1016/j.eswa.2019.01.032

[12]. *S. Lin, A. Liu, J. Wang, X. Kong.* An intelligence-based hybrid PSO-SA for mobile robot path planning in warehouse. Journal of Computational Science, **vol. 67**, 2023, 101938. DOI: 10.1016/j.jocs.2022.101938

[13]. *L. Shangjunnan, L. Shuhai, X. Huaping.* Improved gray wolf optimization algorithm integrating A* algorithm for path planning of mobile charging robots. Robotica. **vol. 42(2)**, 2023, pp. 536-559. DOI:10.1017/S0263574723001625

[14]. *Y.-L. Poy, Z.-Y. Loke, S. Darmaraju, C.-H.Goh, B.-H. Kwan, H. Liu, D.W.K. Ng.* Enhanced Particle Swarm Optimisation for Multi-Robot Path Planning with Bezier Curve Smoothing. Robotics, **vol. 13(141)**, 2024. DOI: 10.3390/robotics13100141

[15]. *T. Bodong, K. Jae-Hoon.* Mobile robot path planning based on bi-population particle swarm optimization with random perturbation strategy. Journal of King Saud University - Computer and Information Sciences. **vol. 36(2)**, 2024. DOI: 10.1016/j.jksuci.2024.10197

[16]. *A. Rodríguez-Molina, J. Solís-Romero, M.G. Villarreal-Cervantes, O. Serrano-Pérez, G. Flores-Caballero.* Path-Planning for Mobile Robots Using a Novel Variable-Length Differential Evolution Variant. Mathematics. **vol 9(4)**, 2021, pp. 357. DOI: 10.3390/math904035

[17]. *I. Noreen.* Collision Free Smooth Path for Mobile Robots in Cluttered Environment Using an Economical Clamped Cubic B-Spline. Symmetry, **vol. 12(9):1567**, 2020, pp. 1-13, DOI: 10.3390/sym12091567

[18]. *Y. Sun, C. Zhang, C. Liu.* Collision-free and dynamically feasible trajectory planning for omnidirectional mobile robots using a novel B-spline based rapidly exploring random tree. International Journal of Advanced Robotic Systems, **vol. 18(3)**, 2021, DOI: 10.1177/17298814211016609

[19]. *A. Orthey, C. Chamzas, L.E. Kavraki.* Sampling-Based Motion Planning: A Comparative Review. https://arxiv.org/abs/2309.13119

[20]. *B. Z.Türkkol, N. Altuntaş, Y. S. Çekirdek.* A Smooth Global Path Planning Method for Unmanned Surface Vehicles Using a Novel Combination of Rapidly Exploring Random Tree and Bézier Curves. Sensors, vol. 24(2024), 8145. DOI: 10.3390/s24248145

[21]. *L. Xingchen, G. Xifeng, Z. Wei, H. Lina.* Smooth and collision-free trajectory generation in cluttered environments using cubic B-spline form. Mechanism and Machine Theory, vol. 169, 2022, 104606. DOI: 10.1016/j.mechmachtheory.2021.104606

[22]. *Y. Romasevych, V. Loveikin Y. Loveikin.* Development of a PSO Modification with Varying Cognitive Term. 2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek), 2022, pp. 1-5, DOI: 10.1109/KhPIWeek57572.2022.9916413.