The Concept of Collision-Free Path Planning of UAV Objects

Artur Babiarz and Krzysztof Jaskot

Abstract. The paper presents the concept of path planning of unmanned flight the helicopter type of flying object. The presented method is based on probabilistic robot motion planning method called B-PPT. Method B-PPT is built upon the known methods of PRM (Probabilistic Roadmap) and RRT (Rapidly-exploring Random Trees). Using the method of B-PPT can plan the path of mobile robots and manipulation. Planning the path of a flying object was carried out in Cartesian space, taking into account only the position and altitude. The speed of the method depends on two input variables, ie the configuration space discretization step and set number of randomly generated configurations. Effect of size on the results obtained was presented in this work.

Keywords: UAV object, path planning, B-PPT method.

1 Introduction

In recent years, it can be seen very strong interest in collision-free motion planning of robots of various types: stationary, wheeled, biped and flying [15, 16]. This is due to increasing automation and robotics, not only industry but also in everyday life. Robots currently found in almost every area of life from basic household tasks, by medicine, industry, and ending with the tasks in space. On the other hand, seeks to create a robot manipulative, which could fully simulate the movements and behavior [11]. In all these cases the main problem is intended to move all or part of its initial position to final position. To solve this task in question are used, inter alia, probabilistic methods.

Artur Babiarz · Krzysztof Jaskot
Silesian University of Technology, Institute of Automatic Control,
Akademicka 16, 44-101 Gliwice, Poland
e-mail: {artur.babiarz,krzysztof.jaskot}@polsl.pl

A. Nawrat et al. (Eds.): Advanced Technologies for Intelligent Systems, SCI 440, pp. 81–94.
springerlink.com

1.1 PRM Method

PRM method consists of two main phases [5, 7, 8]:

- the pre-processing phase,
- the phase of the inquiry.

During pre-processing is carried out random generation of locations in the obstacle-free space and create a graph G. In general, the whole environment is marked C, in which there are obstacles to occupy the O. Then the obstacle free space is defined as $F = \frac{C}{O}$ [5]. The number generated in this way is very high positions in the order of several thousand. Arrangement of neighboring points relative to the vertex x is dictated by the criterion [1, 5, 7, 8]:

$$N_x = \{y \in V : D(x,y) \leq maxdistance\} \quad (1)$$

After completion of the initial phase of processing phase starts queries, which is looking for a road linking the starting point P_s to the end point P_k. During the phase of inquiry be verified if you can connect these two points with vertices graph G.

1.2 RRT Method

An alternative to the PRM method has become a way of planning movement called the Rapidly - exploring random Teres. LaValle is the originator of that in [10, 12, 13] presented the first principles and algorithm of the method. RRT method is to intend of removing the disadvantages of PRM, which include primarily the problems of narrow passages in the planning and generated a very large number of points in space to obtain a satisfactory result [6]. Furthermore, due to the use of two-phase algorithm was accused of PRM method can not be applied in the planning of on - line (due to time-consuming graph search performed during the pre-processing). Therefore, the RRT method it is intended to generate additional points to keep checking the condition:

$$\rho(x_i, x_{i+1}) \leq \varepsilon \quad (2)$$

where:
ρ - metrics,
x_i, x_{i+1} - successively generated random space points in time i and i +1,
ε - very small positive number.

In the basic RRT method is generating random, from a starting point, successive vertices in obstacle-free space. Used space is called the space of states. For this purpose, a state equation $\dot{x} = f(x,u)$, where x is the state of the system, and u in the entry system. Allowable set of inputs ensures minimizing the distance between the current state x_i and another one x_{i+1} by using the metrics described by the equation (2). This way of the planning movements can be used for holonomic and nonholonomic robots.

2 B-PPT Method

The presented probabilistic methods PRM i RRT and their modifications have one thing in common. All are based on the so-called local scheduler, which consists of connecting the neighboring vertices, which are the points of the space, straight lines. Assuming that an object can only move in straight lines to be rather simplistic (the exception is the work of [14], which uses the functions of class C^1). In addition, a description of the motion as a linear function eliminates the possibility to obtain velocity and acceleration of the object by calculating the derivatives of these functions at points of connection lines. A major problem may also designate an area which is free and occupied by potential obstacles workspace. For manipulation objects process area is often very complicated and difficult to describe with the mathematical formulas. The name of the method of B-PPT is derived from the abbreviation of Probabilistic trajectory planning using B - splines curves. The proposed method is designed mainly for the manipulators of industrial robots. It is based on task forward kinematics and planning in natural coordinates. The genesis of the method B-PTT can be analysis of how the PRM and RRT, which consequently led to the use of certain features of both methods and connections together with algorithms design curves B - splines.

Algorithm B-PPT method can be presented in the following steps:

- Step 1: load vectors of the initial and final position,
- Step 2: detection collisions for the initial and final position, when a collision is detected, go to Step 9,
- Step 3: generate random vectors L setpoints configuration determining the robot,
- Step 4: The collision detection for L vectors of the position of step 3,
- Step 5: Select the location of vectors that meet the selection criterion,
- Step 6: calculation of the B-splines curve,
- Step 7: The collision detection position vectors of coordinates corresponding to the step 6, when a collision is detected go to step 8, otherwise go to Step 9,
- Step 8: remove the checkpoints of the curve and return to step 6,
- Step 9: The end.

The calculation result obtained by changing the coordinates described by polynomials of any degree, depending on the demands made by the user. Moreover, this method is quick and easy to use because of the effective methods of generation of B - splines curves and the advantages of probabilistic methods. Description and a detailed description of the method of B-PPT can be found in [3].

2.1 Selection Criterion

Selection of successive configurations is based on the distance between a given, and the new location in the workspace. They can be described as a selection criterion by formulas [2]:

$$N_c = \{\tilde{c} \in N \mid D(c, \tilde{c}) = mindist\} \quad (3)$$

$$N_c = \{\{\tilde{c_1}, \tilde{c_2}\} \in N \mid D(c_k, \tilde{c_1}) < D(c_k, \tilde{c_2})\} \quad (4)$$

where:

c_k - final position,
$\tilde{c}, \tilde{c_1}, \tilde{c_2}$ - tested at a time position,
$\{\tilde{c_i}, \tilde{c_j}\}$ - selected two neighboring positions, i,j=1, 2, ..., m,
m - number of selected positions,
N_c - a set of selected points,
N - the set of all randomly generated points,
D - value of the Euclidean metric.

2.2 B-Spline Curves

Used in the experiments simulated curves are called open uniform B-Spline curves of third degree [9]. B-spline functions given by the formula:

$$B_i^3(t) = \begin{cases} \frac{(2+t)^3}{6} & \text{if } -2 \le t < -1; \\ \frac{(4-6t^2-3t^3)}{6} & \text{if } -1 < t \le 0 \\ \frac{(4-6t^2+3t^3)}{6} & \text{if } 0 < t \le 1 \\ \frac{(2-t)^3}{6} & \text{if } 1 \le t < 2; \\ 0 & \text{if } t \ge 2; \end{cases} \quad (5)$$

While the B-spline curves can be described using the equation:

$$s(t) = \sum_{i=0}^{N-n-1} d_i B_i^n(t) \quad (6)$$

where:

N - number of checkpoints,
n - the degree of function of B-spline,
D_i - i-th control point of the curve.
Rewriting the above equation for the coordinate position is obtained:

$$q^i = \sum_{j=0}^{m} p^j B_j^k(u) \quad (7)$$

where:

m - the number of checkpoints,
k - the degree of B-spline functions,

q^i - i-th coordinate of the position, i = 1-n
n - number of the coordinate position of the robot,
u - fixed increasing sequence of numbers. p^j - drawn at random in the j-th iteration of the coordinate position, which is the control point B-spline curve, satisfying the condition $q^i_{min} \leq p^j \leq q^i_{max}$

In order to ensure the interpolation values of the initial and final coordinates of the position shall be a multiple of the value that is equal to the degree of B-spline function Fig. 1.

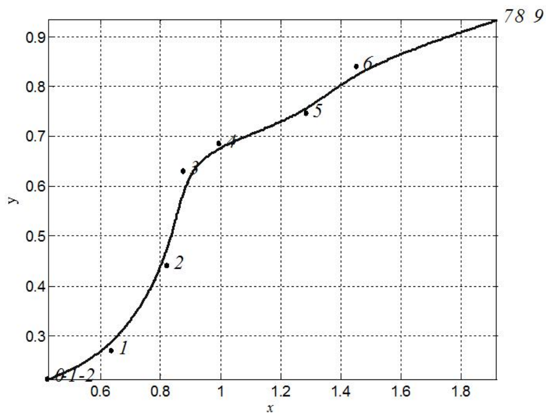


Fig. 1 Example of B-spline curve

3 UAV Mathematical Model

The proposed flight path generation method is carried out on several assumptions:

1. flight takes place at a constant height,
2. flight control system has a photo area where the road is generated, eg from a digital camera,
3. the control system is supplied only item in the i-th moment (x_i, y_i),
4. path is represented by spline curves, degree 3, which allows you to calculate air speed along the path, which is a preset speed, according to the formula (10).
5. input data are vectors of initial position, final altitude and the area representing obstacles and free space.

Determined path is supplied to the control system (exactly the autopilot), shown in the Fig. 2. The control system based on any given location is calculated in the appropriate control signals (in this case the angles responsible for the flight UAVs). Vector appropriate setting angle is applied to the controller, which using information from sensors maintains commanded the flight parameters. In addition, assume that the UAV is equipped with an autopilot and a digital camera and high speed allow communication with the base station. Due to the required relatively high speed of calculation also assumes that the mathematical operations are carried out on computing unit located in the base station.

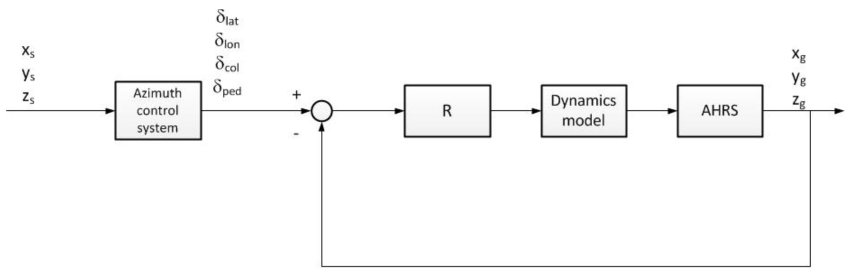


Fig. 2 The system control, where: (x_s, y_s, z_s) - start position, (x_g, y_g, z_g) - goal position, R - regulator, AHRS - Attitude and Heading Reference System, δ_{lat} - aileron servo input, δ_{lon} - ellevator servo input, δ_{col} - collective pitch servo input, δ_{ped} - rudder servo input

Typical UAV dynamics model consists of 12 variables developed in [18]. This model is very accurate, which can cause problems during generation of flight paths. Therefore, in order to facilitate the calculations used a model with six variables (when the UAV has a preset position, altitude and speed):

$$\begin{aligned} \dot{z}_x &= v\cos(\psi) \\ \dot{z}_y &= v\sin(\psi) \\ \dot{\psi} &= \alpha_\psi(\psi^c - \psi) \\ \dot{v} &= \alpha_v(v^c - v) \\ \ddot{h} &= -\alpha_{\dot{h}}\dot{h} + \alpha_h(h^c - h) \end{aligned} \tag{8}$$

where: ψ^c,v^c and h^c are specified course, speed and height passed to the autopilot and α_* are positive constants [4]. This model captures the fundamental dynamics of the system [4, 17]. Further simplification can be assumed that the road will be flown at a constant speed (v^c) and a constant height. UAV dynamics simplifies to:

$$\begin{aligned} \dot{z}_x &= v\cos(\psi) \\ \dot{z}_y &= v\sin(\psi) \\ \dot{\psi} &= \alpha_\psi(\psi^c - \psi) \end{aligned} \tag{9}$$

where: v^c means a constant speed along the path, and ψ^c is the position is sent to the autopilot. Using the dynamics described by equation 9, allows the generator to provide a path similar form. Specifically, the path generator is:

$$\begin{aligned} \dot{x}_r &= v^c\cos(\psi_r) \\ \dot{y}_r &= v^c\sin(\psi_r) \\ \dot{\psi}_r &= u \end{aligned} \tag{10}$$

where: (x_r, y_r) is the current position, ψ_r is a course, v^c is the set linear velocity along the path, and u is a given course.

4 Examples

At this point, will be presented examples of two cases of generation of UAV flight path. They differ in the placement of obstacles in the helicopter flight. In addition, assume that the UAV is equipped with an autopilot and a digital camera and high speed allow communication with the base station. Due to the required relatively high speed of calculation also assumes that the mathematical operations are executed on computing unit located in the base station. Examples used in view of a specific area of the city. Simulations performed by changing the discretization step-flight area and the number of randomly generated vectors of position in space flight. The dimensions given on the graphs are symbolic and are not actual size of the area in which you want to move the UAV. Obstacles are represented as rectangles. It allows to easily check whether a given point lies in the flight path area free from obstacles. The figure 3 shows the first example of the area used in simulations. Simulation used to approximate the objects on the photograph in the form of cuboids, which facilitates collision detection issue.

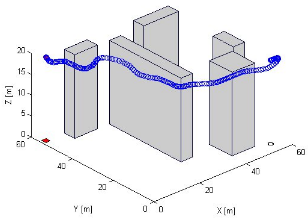
Fig. 3 The first area

In the first example, set the following parameters:

- Number of vectors generated position 200,
- Discretization step 5 [m].

Figure 4 shows the results obtained from simulation.

Graphs represent the flight path of the generated vectors and the distribution of positions used to plan the flight. The result was that in order to solve the problem, the algorithm needed 159 positions.



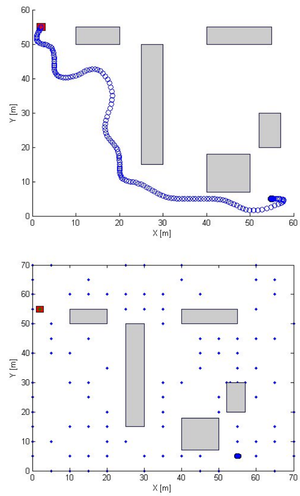
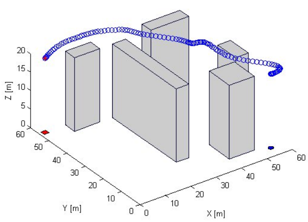


Fig. 4 The simulation: number of position 200, discretization step 5 [m]



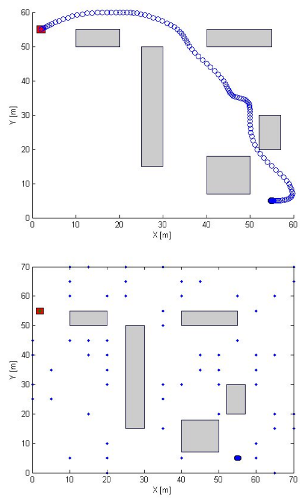
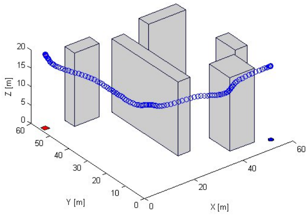


Fig. 5 The simulation: number of position 100, discretization step 5 [m]

In the next simulation step is digitizing the same, but reduced the number of generated positions to 100. Algorithm without any major problems generated collision-free path and the need for this less than half of the vectors of position, exactly 77 (Figure 5). It was therefore a further attempt to reduce the generated vectors. Reduced it to 70. The algorithm in this case also advised and has generated a path with 59 positions (Figure 6). Discretization step is unchanged.

In another example, the increased number of randomly generated positions, in order to check the influence of this number, for example, the shape of the path. For 1000 generated positions the algorithm needed 840 vectors to compute collision-free path of the discretization step 5 [m]. The results present figure 7.



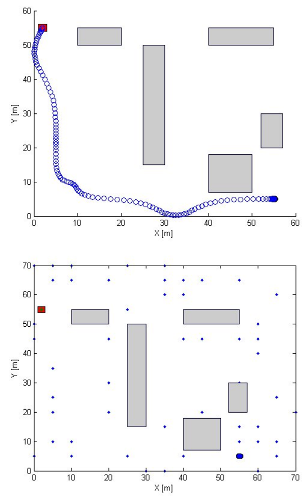
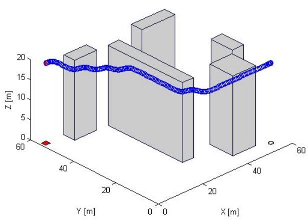


Fig. 6 The simulation: number of position 70, discretization step 5 [m]



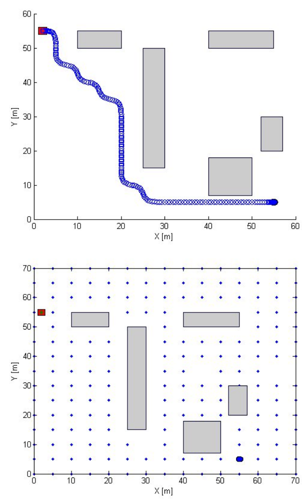


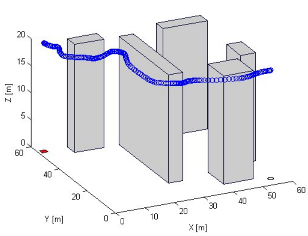
Fig. 7 The simulation: number of position 1000, discretization step 5 [m]

In the following simulations (figure 8, figure 9), the discretization step was reduced to 0.5 [m] and the calculations performed respectively for 100 and 1000 random positions.

The algorithm needed respectively 79 and 815 positions in order to generate the path.

After obtaining satisfactory results of simulation experiments performed for the area shown in the figure 10. The aim is to investigate the effect of density of obstacles in the flight on the effectiveness of planning collision-free path.

As a result of increased obstacles to first-generation collision-free path attempts have failed. Only for 10 [m] discretization step gives a goal. The figure 11 shows



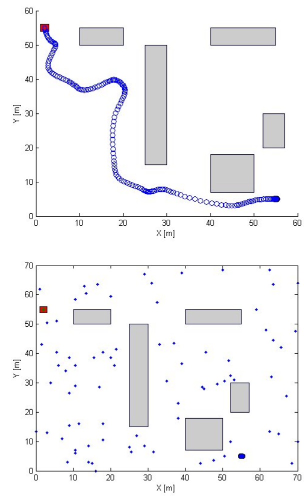
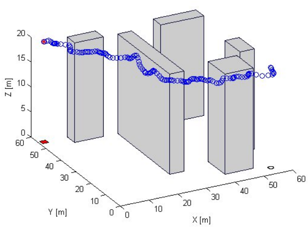


Fig. 8 The simulation: number of position 100, discretization step 0,5 [m]



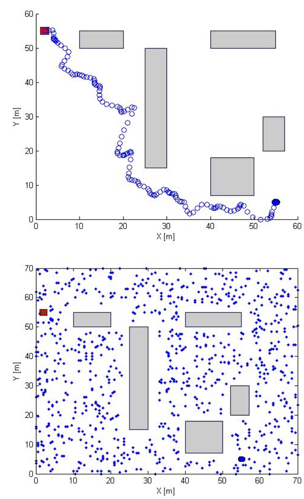


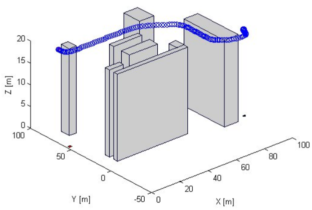
Fig. 9 The simulation: number of position 1000, discretization step 0,5 [m]

the results for 100 sample in which the algorithm calculated the collision-free path generated for 200 positions. The algorithm needed 165 positions.

Just as in the first example, assume that increasing the number of generated positions should improve the algorithm. The results for 1000 random positions illustrated Fig. 12. To calculate the flight path algorithm needed 840 positions.

Despite obtaining collision-free path, shown in the example above, the generation time and number of attempts, after which the obtained results are not fully acceptable for the work control system on-line. In order to improve performance assumes a modification of the proposed algorithm. Precisely in the area of the obstacles created the sub-areas in which additional randomly generated position. This is the density

Fig. 10 The second area



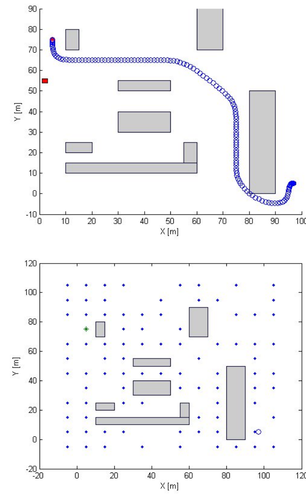
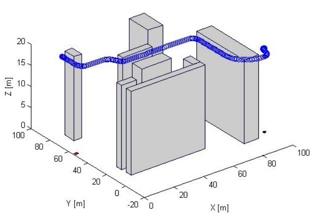


Fig. 11 The simulation: number of position 200, discretization step 10 [m]



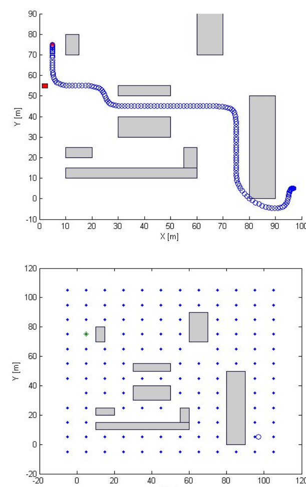


Fig. 12 The simulation: number of position 1000, discretization step 10 [m]

of free space, then already in the standard way to search for collision-free path. Below is an example of executing the path generation in the manner described above. The number of randomized positions in the whole space was set at 200 and the same reference number of positions was generated in each sub-area associated with each obstacle. Whole space discretization step and all sub is 0.5 [m]. The results are shown in the figure 13. The algorithm needed 1250 positions to find a collision-free of the flight path of UAV object.

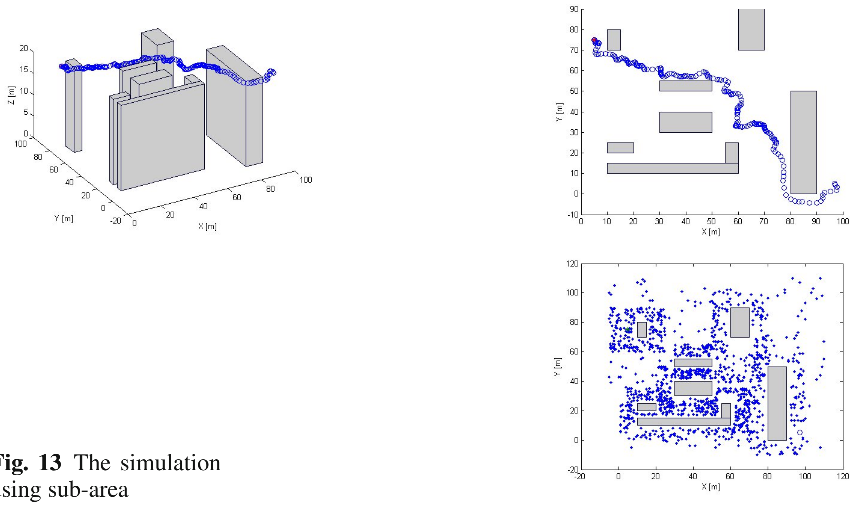


Fig. 13 The simulation using sub-area

5 Summary

The above method is a proposal to solve the problem of planning collision-free flight path for the object type of UAV. In the simulation experiments carried out given that the control system has full information about the area where the helicopter is moving and covering the area obstacles. It is also important to assume that the flying object itself is equipped with fully autonomous autopilot and very good quality digital camera. But the problem is the issue of the calculation. In the above examples assume that the calculations are executed off-line or control system each time waiting for the results of the algorithm. However the main result of the analysis is that it is difficult to find any correlation between the number of preset positions, and the number of selected locations of the collection. The examples are not given the time of calculation, since it also depends on the parameters of the equipment on which the calculations were performed. One can only mention that in the case of calculations for 100 preset setup time was 1 [s], for 1000 of about 2 [s], and for 2000 configurations of about 3 [s]. In addition, simulation experiments carried out do not take into account the criterion for choosing the next location. A very important conclusion after observation of the results is the shape of the flight path. We see that a small number of randomized positions obtained path, which is modeled curves is also spline smooth curve. In some examples, received flight paths are ideal, ie no need to continuously change the direction of flight, the object passes along the obstacles without unnecessary movements, which can be seen in other simulations. Serious problem is the so-called passage through narrow corridors. Well this was illustrated in Example 2, where he concentrated in the area of flight obstacles. Results are presented here only the results of simulation experiments. The correctness of the exact same method can be confirmed through verification of the real object. The main issues that arise are likely to be trouble with the speed of microprocessor computing unit flying object, and image processing to obtain the position of any obstacles. You should also pay attention to the problems of processing the location signals to real signals relevant regulations of UAV systems. This issue is a completely separate issue, and this work has not been touched. But as regards the implementation of the method of B-PPT is characterized by the simplicity of the algorithm and has a very simplified mathematical tools, which uses only the necessary information. The problem is the lack of repeatability of results. This method can be used only if we are to achieve a final position or configuration, since the shape of the path between these points we do not have flow. This is due to base B-PPT method, the use of probability calculus.

References

1. Aarno, D., Kragic, D., Christensen, H.: Artificial potential biased probabilistic roadmap method. In: Proc. IEEE International Conference on Robotics and Automation, New Orleans, pp. 461–466 (2004)
2. Babiarz, A.: Nowe kryterium przeszukiwania w metodzie bazujacej na algorytmie PRM. In: XV Krajowa Konferencja Automatyki, Warszawa, pp. 289–294 (2005)
3. Babiarz, A.: Planowanie trajektorii manipulatorów z zastosowaniem krzywych B - sklejanych. Rozprawa doktorska, Gliwice (2005)
4. Beard, R.W., McLain, T.W., Goodrich, M.A., Anderson, E.P.: Coordinated target assignment and intercept for unmanned air vehicles. IEEE Transactions on Robotics and Automation 18(6), 911–922 (2002)
5. Burns, B., Brock, O.: Information Theoretic Construction of Probabilistic Roadmaps. In: Proc. of the International Conference on Robotics and Systems, Las Vegas, Nevada (2003)
6. Hsu, D., Jiang, T., Reif, J., Sun, Z.: The Bridge Test for Sampling Narrow Passages with Probabilistic Roadmap Planners. In: IEEE International Conference on Robotics and Automation, Taipei, Taiwan, pp. 4420–4426 (2003)
7. Hsu, D., Latombe, J.C., Kurniawati, H.: On the Probabilistic Foundations of Probabilistic Roadmap Planning. In: 12th Int. Symp. on Robotics Research, San Francisco (2005)
8. Kavraki, L.E., Latombe, J.C.: Probabilistic Roadmaps for Robot Path Planning. In: Gupta, K., del Pobil, A. (eds.) Practical Motion Planning in Robotics: Current Approaches and Future Directions, pp. 33–53. John Wiley (1998)
9. Kiciak, P.: Podstawy modelowania krzywych i powierzchni. WNT, Warszawa (2000)
10. Kuffner, J.J., LaValle, S.M.: RRT-Connect: An efficient approach to single-query path planning. In: Proc. IEEE Int'l Conf. on Robotics and Automation, San Francisco, CA, pp. 995–1001 (2000)

11. Kuffner, J.J., Nishiwaki, K., Kagami, S., Inaba, M., Inoue, H.: Motion planning for humanoid robots under obstacle and dynamic balance constraints. In: Proc. IEEE Int'l Conf. on Robotics and Automation, Seoul, Korea, pp. 692–698 (2001)
12. LaValle, S.M.: Rapidly-exploring random trees: A new tool for path planning. Computer Science Dept. (1998)
13. LaValle, S.M., Kuffner, J.J.: Randomized Kinodynamic Planning. Int'l Journal or Robotics Research 5(8) (2001)
14. Nieuwenhuisen, D., Kamphuis, A., Mooijekind, M., Overmars, M.H.: Automatic Construction of High Quality Roadmaps for Path Planning. Utrecht University, technical report UU-CS-2004-068 (2004)
15. Pettersson, P.O.: Helicopter Path Planning using Probabilistic Roadmaps, Master's thesis, Dept. of Computer Science at Linkopings Universitet (2003)
16. Pettersson, P.O.: Sampling-based Path Planning for an Autonomous Helicopter. Linkoping Studies in Science and Technology Thesis No. 1229 (2006)
17. Proud, A.W., Pachter, M., D'Azzo, J.J.: Close formation flight control. In: Proceedings of the AIAA Guidance, Navigation and Control Conference, Portland, OR, AIAA Paper No. 99-4207 (1999)
18. Roskam, J.: Airplane Flight Dynamics and Automatic Flight Controls, Design. Analysis and Research Corporation, Lawrence, KS (2001)