Trajectory modelling when bypassing obstacles

A A Dubanov[1], T V Ausheev[2]

[1]Buryat State University, 24 A, Smolin-street, Ulan-Ude, Russia
[2]East Sibirian State University of Technology and Management, 40V, Klyuchevskaya-street, Ulan-Ude, Russia

alandubanov@mail.ru, atv62@mail.ru

Abstract. This article deals with the modeling of trajectories on the plane and on the surface represented by a point basis. The obstacle is modeled on the parameter plane and consists of two parts. The first part is the area where access is denied. The second part is the area of trajectory distortion, if there were no obstacle. The article deals with the cases of modeling trajectories to a moving target, with moving obstacles and from a moving object of pursuit. All proposed algorithms are implemented in the system of computer mathematics MathCAD.

1. Introduction

There are two objects, participants of a task of prosecution moving on a plane or on a surface. The object of the prosecution may move along a certain trajectory and can move depending on the movement patterns of the pursuing object. Objects that should be avoided by the pursuing object may be located on a piece of the plane or surface. However, it is possible aim object should avoid obstacles too. In this regard, the problem of computer-aided design of the preliminary trajectory between the pursuing object and the aim object becomes urgent. Such a trajectory is supposed to be built at every moment of time and the pursuing object must make a decision to adhere to such a trajectory at any given time.

2. Formulation of the problem

The research task is to create a mathematical model of obstacles, consisting of two areas, the area where an object can be destroyed and the area where the trajectory is distorted to get around an obstacle. The purpose of the solution is to develop an algorithm for automated construction of a trajectory between two points such that it bends (evades) from the region, the object to be destroyed. The points between which the trajectory is built can move along their trajectories, and the obstacle can make a movement. In other words, the situation in some area of space can be time-varying and the behavior of objects is independent of each other. The implementation of the test programs will be done in the MathCAD computer mathematics system. Since we are dealing with dynamic objects, all results will be provided with links to animated images.

3. Obstacle modelling

In this paper, we assume the obstacle can be enclosed in a circle of a certain radius. Then we will need to exclude the values from the plane of the parameters that fall into the circle of the obstacle.

We form piecewise smooth continuous parametric functions on the parameter plane (X, Y) [1]. Since we will model in the MathCAD computer mathematics system that we need to define a plane segment (X, Y). Let it be a square of size $[(0..100)] \times [(0..100)]$.

$$L_{11}(u, C, r) = \begin{bmatrix} u \\ 0 \end{bmatrix}$$

$$L_{12}(u, C, r) = \left| \begin{array}{l} \begin{bmatrix} u \\ C_y \end{bmatrix} if\ 0 \le u \le C_x - r \\ \begin{bmatrix} u \\ -\sqrt{r^2 - (u - C_x)^2} + C_y \end{bmatrix} if\ C_x - r < u < C_x - r \\ \begin{bmatrix} u \\ C_y \end{bmatrix} if\, C_x + r \le u \le 100 \end{array} \right.$$

$$L_{21}(u, C, r) = \left| \begin{array}{l} \begin{bmatrix} u \\ C_y \end{bmatrix} if\ 0 \le u \le C_x - r \\ \left[\begin{bmatrix} u \\ \sqrt{r^2 - (u - C_x)^2} + C_y \end{bmatrix}\right] \text{если } C_x - r < u < C_x - r \\ \begin{bmatrix} u \\ C_y \end{bmatrix} if\ C_x + r \le u \le 100 \end{array} \right.$$

$$L_{22}(u, C, r) = \begin{bmatrix} u \\ 100 \end{bmatrix}$$

In the above formulas, $C = \begin{bmatrix} C_x \\ C_y \end{bmatrix}$ is the center of the circle, r is the radius of the circle, u is the parameter along the X Axis from 0 to 100.

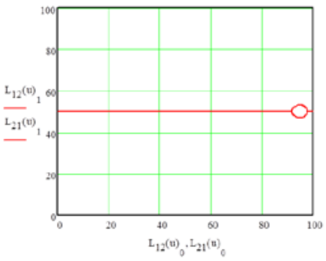

Figure 1. Modeling of piecewise smooth continuous lines

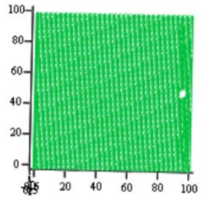

Figure 2. A segment of a plane with a hole

The result of modeling lines is shown in Figure 1. This figure is supplemented by a link to an animated image [3], where the dynamic movement of lines can be seen. Now let's form two half-planes.

$$S_1(u, v, C, r) = \frac{C_y - v}{C_y} \cdot L_{11}(u, C, r) + \frac{v}{C_y} \cdot L_{12}(u, C, r)$$

$$S_2(u, v, C, r) = \frac{100 - v}{100 - C_y} \cdot L_{21}(u, C, r) + \frac{v - C_y}{100 - C_y} \cdot L_{22}(u, C, r)$$

Now we bring together two half-planes:

$$S_f(u, v, C, r) = \left| \begin{array}{l} S_1(u, v, C, r) \text{ если } 0 \le u \le C_y \\ S_2(u, v, C, r) \text{ если } C_y < u \le 100 \end{array} \right. .$$

In Figure 2 you can see the result of the formation of a plane without points falling into a given circle. Figure 2 supplemented by a link to an animated image [4]. You can see the dynamics of the hole movement on the plane segment.

For example, we take the surface of a paraboloid and place a hole on it.

$$P(u, v) = \begin{bmatrix} S_f(u, v, C, r)_x \\ S_f(u, v, C, r)_y \\ \frac{(S_f(u,v,C,r)_x - 50)^2 + (S_f(u,v,C,r)_y - 50)^2}{1000} + 5 \end{bmatrix}.$$

In Figure 3 you can see the hole moves on the surface of the paraboloid. Fig. 3 is supplemented with a link to the animated image [5]. In addition, the resource [2] posted a full listing of the program with comments. This obstacle model, as an area of definition, uses all the space in which the participants of the pursuit task will move. To place several dynamic obstacles, this model is of course possible, but if the linear dimensions of the space are much larger than the linear ones of the obstacles, then it makes sense to model the obstacles themselves as local inclusions, and use the presented model as a visualization tool. The modeling method for holes is taken from the source [1].

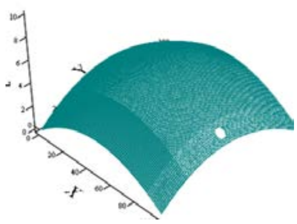

Figure 3. The paraboloid surface with a hole

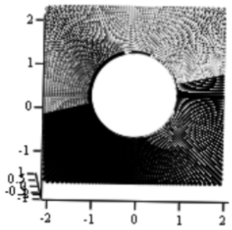

Figure 4. Local model of the obstacles

4. Local obstacle modeling

We will consider the obstacle as an object consisting of two parts. Areas where access is denied and the area of distortion of trajectories. If the scope of destruction on the plane to represent a circle of a certain radius (Figure 4), the distortion can be represented in the form of a square of a certain size.

In Figure 4 the scope of the destruction is presented in the form of a circle of radius 1 with center at the point of origin. The circle is enclosed in a square with side 4, the center of the square coincides with the center of the circle. Parameterization of the plane segment, where the trajectory is distorted, in Figure 4 fully corresponds to the statement in the second paragraph of this article but we need to get a different parameterization to calculate the trajectory.

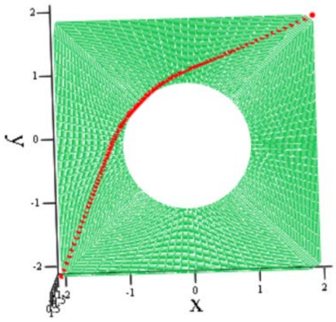

Figure 5. Parameterization of a plane segment

If the point of the unit circle (Figure 5) is identified by the parameter t $L_1(t) = \begin{bmatrix} cos(t) \\ sin(t) \end{bmatrix}$, then it should be put in accordance with the point of the square, which is the intersection of the line connecting the center of the circle and the point $L_1(t)$, and the line of the square. In other words, it is a point:

$$L_2(t) = \left| \begin{array}{c} \begin{bmatrix} 2 \\ 2 \cdot tg(t) \end{bmatrix} \; if \; 0 \le t \le \frac{\pi}{4} \\ \begin{bmatrix} \frac{2}{tg(t)} \\ 2 \end{bmatrix} \; if \; \frac{\pi}{4} < t \le \frac{3\pi}{4} \\ \begin{bmatrix} -2 \\ -2 \cdot tg(t) \end{bmatrix} \; if \; \frac{3\pi}{4} < t \le \frac{5\pi}{4} \\ \begin{bmatrix} -\frac{2}{tg(t)} \\ -2 \end{bmatrix} \; if \; \frac{5\pi}{4} < t \le \frac{7\pi}{4} \\ \begin{bmatrix} 2 \\ 2 \cdot tg(t) \end{bmatrix} \; if \; \frac{7\pi}{4} < t \le \frac{8\pi}{4} \end{array} \right. .$$

Now we can define a segment of the plane (Figure 5):

$$S(t,h) = (1-h) \cdot L_1(t) + h \cdot L_2\,(t).$$

Figure 5 is supplemented by a link to an animated image [6], where you can see the point moving on the local segment of the trajectory. The segment that we got in Figure 5 can be inserted into any surface of the form $Z = f(X,Y)$, after making a new parameterization. The method proposed in this paragraph allows you to perform not only the calculation of the trajectory, but also to visualize the obstacle.

5. Modelling of local segment of the trajectory

Consider carefully the Figure. 5. We set the task of constructing a trajectory from the lower corner of the square to the upper right corner of the square. This is the maximum time that our trajectory will affect, but will not be included in the area of defeat. Although this point can be adjusted by setting a larger radius area of defeat. The entry and exit points in the square correspond to the values of the parameter t:

$$t_1 = \frac{\pi}{4}, t_2 = \frac{5}{4}\pi.$$

Consider the domain of the function:

$$S(t,h) = (1-h)\cdot L_1(t) + h\cdot L_2(t).$$

It will be a rectangle on the plane (t,h) with dimensions $[0 \quad 2\pi] \times [0 \quad 1]$.

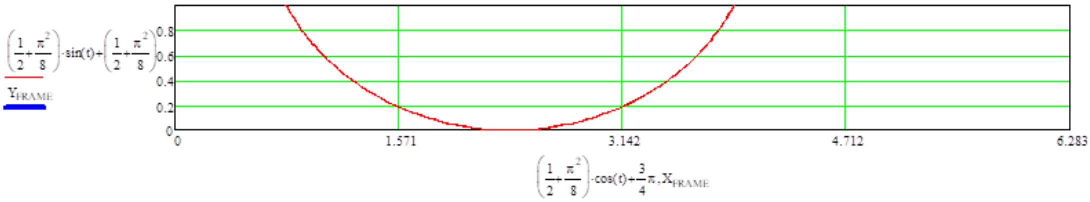

Figure 6. Simulation of a path segment

Mark on this area the definition of the entry point to our square (Figure 6):

$$P_1 = \begin{bmatrix} t_1 \\ 1 \end{bmatrix}, P_2 = \begin{bmatrix} t_2 \\ 1 \end{bmatrix}.$$

Mark the straight line:

$$p(t) = \begin{bmatrix} t \\ 0 \end{bmatrix}.$$

Construct a circle passing through points P_1, P_2 and touching the line $p(t)$. This will be a circle of radius:

$$r = \frac{1}{2} + \frac{\pi^2}{8},$$

and centered at point:

$$C = \begin{bmatrix} \frac{3}{4}\pi \\ \frac{1}{2} + \frac{\pi^2}{8} \end{bmatrix}.$$

So we implement the mechanism of building a circle as in AutoCAD: "Point, Point, Touching." Thus, the circle arc in Fig. 6 is determined by the formula:

$$(\varphi) = \left(\frac{1}{2} + \frac{\pi^2}{8}\right)\cdot cos(\varphi) + \frac{3}{4}\pi, h(\varphi) = \left(\frac{1}{2} + \frac{\pi^2}{8}\right)\cdot sin(\varphi) + \frac{1}{2} + \frac{\pi^2}{8},$$

where the parameter φ belongs to the range

$$\varphi \in \left[arcsin\left(\frac{\frac{1}{2} - \frac{\pi^2}{8}}{\frac{1}{2} + \frac{\pi^2}{8}}\right); \quad \pi - arcsin\left(\frac{\frac{1}{2} - \frac{\pi^2}{8}}{\frac{1}{2} + \frac{\pi^2}{8}}\right)\right].$$

Figure 6 is supplemented by a reference to the animated image [13], where the arc point of the circle runs from the value of P_1 to the value of P_2 .

Substitution of the circular arc equations in the equation of the plane segment reduces to the equation of the local segment of the trajectory in the region of distortion:

$$S\big(t(\varphi), h(\varphi)\big).$$

The trajectory in the distortion region is shown in Figure 5.

Figure 5 supplemented by a reference to the animated image [6], where the trajectory point moves along the area of distortion. By the way, setting a straight line:

$$(t) = \begin{bmatrix} t \\ 0 \end{bmatrix},$$

we set the touch to the area of defeat. Ordinate

$$p(t) = \begin{bmatrix} t \\ 0 \end{bmatrix}$$

we can choose from the range [0 1]. The full text of this program can be downloaded from [2].

6. Modelling of local segment of the trajectory

When calculating the trajectory in the world coordinate system, you can resort to such a model of the distortion zone. This is a circle of radius r, centered at point (Figure 7):

$C = \begin{bmatrix} C_x \\ C_y \end{bmatrix}$.

Our task is to calculate the trajectory from R_1 to R_2 .

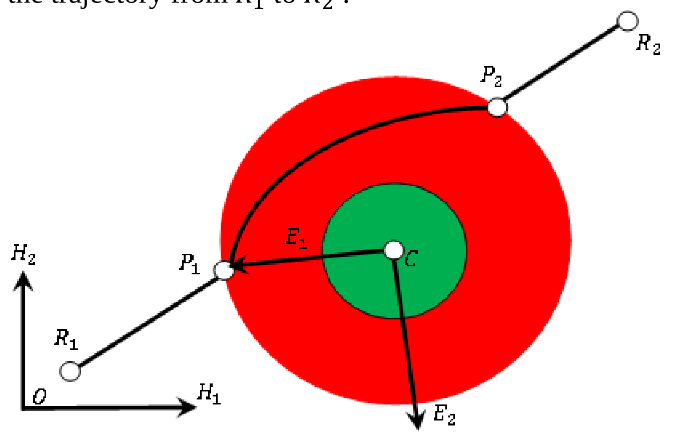

Figure 7. Modeling the trajectory in the world coordinate system

Finding the intersection points of P_1, P_2 line $(R_1 \quad R_2)$ with a circle (C, r) – will cause the solution of the square equation:

$$A_{eq} \cdot t^2 + B_{eq} \cdot t + C_{eq} = 0.$$

Where the coefficients are:

$$A_{eq} = (R_2 - R_1) \cdot (R_2 - R_1)$$
$$B_{eq} = (R_2 - R_1) \cdot (R_1 - C)$$
$$C_{eq} = (R_1 - C) \cdot (R_1 - C) - r^2$$

This square equation is obtained from the solution of a system of equations with respect to the parameter t.

$$R(t) = (1 - t) \cdot R_1 + t \cdot R_2$$
$$(R(t) - C) \cdot (R(t) - C) = r^2$$

If the discriminant of the square equation:

$D_{eq} = B_{eq}^2 - 4 \cdot A_{eq} \cdot C_{eq}$.

Is less than or equal to 0, then the trajectory is a straight line $(R_1 \quad R_2)$. If the discriminant is greater than 0, the trajectory falls into the distortion zone. The solution of our equation will be two values t_1 and t_2 corresponding to points P_1 and P_2 in Fig. 7. Next, go to the local basis $(E_1 \quad C \quad E_2)$ formed by vectors $E_1 = (P_1 - C)/|P_1 - C|$ and $E\_2 = \begin{bmatrix} -E_{1y} \\ E_{1x} \end{bmatrix}$ (Figure 7). In the coordinate system $(E_1 \quad C \quad E_2)$, the point P_1 has the coordinates:

$$\begin{bmatrix} r \\ 0 \end{bmatrix}.$$

And the point P_2 looks like:

$$\begin{bmatrix}(P_2 - C)\cdot E_1\\(P_2 - C)\cdot E_2\end{bmatrix}.$$

In the local coordinate system, the tilt angle to the E_1 axis is 0 for point P_1 , and for point P_2 will be:

$$\alpha = \begin{vmatrix} arccos\left(\frac{(P_2-C)\cdot E_1}{|P_2-C|}\right) \; if \; \frac{(P_2-C)\cdot E_2}{|P_2-C|} \geq 0 \\ -arccos\left(\frac{(P_2-C)\cdot E_1}{|P_2-C|}\right) \; if \; \frac{(P_2-C)\cdot E_2}{|P_2-C|} < 0 \end{vmatrix}.$$

The segment plane in the coordinate system of $(E_1 \quad C \quad E_2)$ will be presented in the form:

$$S(t,h) = (1-h)\cdot\begin{bmatrix}r_0\cdot cos(t)\\r_0\cdot sin(t)\end{bmatrix} + h\cdot\begin{bmatrix}r\cdot cos(t)\\r\cdot sin(t)\end{bmatrix},$$

where r_0 is the radius of the field of area of defeat, and r is the radius of the area of distortion.

As in the previous paragraph on the parameter plane (t,h) with dimensions $[0 \quad 2\pi]\times[0 \quad 1]$ let's postpone the point images P_1, P_2. Point images will be:

$\begin{bmatrix}0\\1\end{bmatrix}$ and $\begin{bmatrix}\propto\\1\end{bmatrix}$, respectively.

Now, using the mechanism of constructing a circle "Point, Point, Radius" (as in AutoCAD), draw an arc of a circle of radius $\frac{1}{2}+\frac{\pi^2}{8}$. The center of such a circle will be at the point:

$$\begin{bmatrix}\frac{\propto}{2}\\1+\sqrt{\left(\frac{1}{2}+\frac{\pi^2}{8}\right)^2-\left(\frac{\propto}{2}\right)^2}\end{bmatrix}.$$

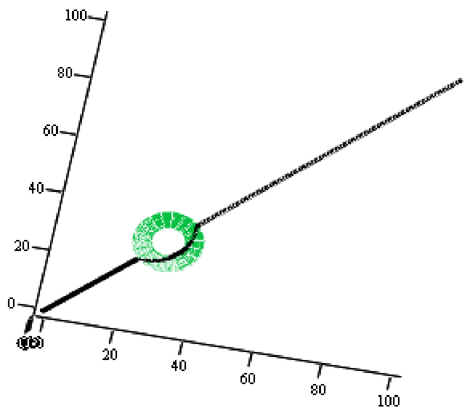

Figure 8. Trajectory in the world coordinate system

The point will run on the plane (t,h) along the trajectory:

$$\begin{bmatrix}t(\gamma)\\h(\gamma)\end{bmatrix} = \left(\frac{1}{2}+\frac{\pi^2}{8}\right)\cdot\begin{bmatrix}cos(\gamma)\\sin(\gamma)\end{bmatrix},$$

Where

$$\gamma \in \left[-arccos\left(\frac{\sqrt{\left(\frac{1}{2}+\frac{\pi^2}{8}\right)^2-\left(\frac{\propto}{2}\right)^2}}{\frac{1}{2}+\frac{\pi^2}{8}}\right) \quad arccos\left(\frac{\sqrt{\left(\frac{1}{2}+\frac{\pi^2}{8}\right)^2-\left(\frac{\propto}{2}\right)^2}}{\frac{1}{2}+\frac{\pi^2}{8}}\right)\right].$$

The resulting arc segment:

$$S\big(t(\gamma),h(\gamma)\big) = \big(1-h(\gamma)\big)\cdot\begin{bmatrix}r_0\cdot cos\big(t(\gamma)\big)\\r_0\cdot sin\big(t(\gamma)\big)\end{bmatrix} + h(\gamma)\cdot\begin{bmatrix}r\cdot cos\big(t(\gamma)\big)\\r\cdot sin\big(t(\gamma)\big)\end{bmatrix},$$

remains to be translated into the world coordinate system $[H_1 \quad O \quad H_2]$ (Figure 7, Figure 8).

That was done by the formulas:

$$L(\gamma) = \begin{bmatrix}S\big(t(\gamma),h(\gamma)\big)\cdot q_1\\S\big(t(\gamma),h(\gamma)\big)\cdot q_2\end{bmatrix} + C\,,$$

where

$$q_1 = \begin{bmatrix} H_1 \cdot E_1 \\ H_1 \cdot E_2 \end{bmatrix}, q\_2 = \begin{bmatrix} H_2 \cdot E_1 \\ H_2 \cdot E_2 \end{bmatrix}, H_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, H_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Figure 8 supplemented by a link to an animated image [8], which move and the obstacle itself, and the objects between which the trajectory is built. The full text of the program can be viewed on the resource [2].

Conclusion

According to the results of the conducted researches an algorithm is proposed for automated generation of the dynamic trajectories of the dynamic obstacles bypass. It is proposed to depict the obstacle model in the form of two areas, the area where an object can be destroyed and the area where the trajectory is distorted. And the area of distortion can be both a circle and a square.

The model of trajectories construction proposed in this article considers the cases when the areas of defeat and distortion of several obstacles do not intersect. This can be considered as a separate topic for research. Also, further studies will introduce a separate area of smoothing where the segments of the trajectories will be joined with a certain degree of smoothness.

This paper proposes models and algorithms for obstacle avoidance. According to the proposed models and algorithms, test programs for calculating trajectories in the system of computer mathematics MathCAD are written. It is assumed that these models and algorithms for constructing trajectories can be used in the development of robotic systems, in the models of which there are elements of the pursuit problem.

References

[1] A. A. Dubanov 2003 Design of basic elements of the form and parametrization of composite surfaces *Information technologies* Vol **10** pp 42–51
[2] Simulation of trajectories in obstacle avoidance. URL: http://dubanov.exponenta.ru (date accessed: 22.07.2019)
[3] *Modeling of intermediate lines* URL: https://www.youtube.com/watch?v=ixNhGHaBXPU&feature=youtu.be (date accessed: 19.03.2019)
[4] *The obstacle in the plane definition* URL: https://youtu.be/Uc4slgitOVw (date accessed: 19.03.2019)
[5] *Obstacle on the paraboloid* URL: https://youtu.be/7xZAzvy0wGc (date accessed: 19.03.2019)
[6] *Movement along the local segment of the trajectory* URL: https://youtu.be/gE7XZWIlGjE (date accessed: 19.03.2019)
[7] *Simulation of a path segment* URL: https://www.youtube.com/watch?v=lxKR3-tnIs4 (date accessed: 19.03.2019)
[8] *Trajectory in the world coordinate system* URL: https://youtu.be/l1iZQaBxN1c (date accessed: 19.03.2019)
[9] Simoson A J 2007 *Pursuit Curves for the Man in the Moone* vol 38 (Washington: The College Mathematics Journal) pp 330–338
[10] Nahin P J 2007 *Chases and Escapes: The Mathematics of Pursuit and Evasion* (Princeton University Press) p 270
[11] Barton J C and Eliezer C J 2000 On Pursuit Curves *The Journal of the Australian Mathematical Society*) pp 358–371
[12]. Petrosyan L A 1993 *Differential Games of Pursuit* (Singapore: World Scientific) p 326
[13] Kuzmina L I and Osipov Yu V 2013 Calculation of the Pursuit Curve Length *Journal for Computational Civil and Structural Engineering* Vol **9** no 3 pp 31–39

Acknowledgments

The work was supported by the project of the Buryat State University innovation grants competition 2019 for the creation of hardware and software support complex.

Scientific supervisor – A A Dubanov.

The author thanks the Director of the Institute of mathematics and Informatics of Buryat State University Antonova Larisa Vasilievna for the assistance and support in the execution of the grant of BSU.