The task of pursuing objects moving on different surfaces

A A Dubanov¹, T V Ausheev² and D A Semenov¹

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

alandubanov@mail.ru, atv62@mail.ru, semenov.da17@gmail.com

Abstract. The purpose of this article is to describe a mathematical model of the pursuit problem in the case when the pursued and pursuing objects move along different surfaces one above the other. Orthonormal dynamic bases, which are determined by the vectors of velocities and normals to the surfaces, are introduced on the surfaces under consideration, at the points where objects are located, and tangent planes are constructed. The coordinates of our model's opponent are projected onto each specified plane for analysis and decision making. The velocities of the objects in the model are constant in magnitude. The inertness of objects is modeled using the angular velocity of rotation, which is essential in the described model. As a result of research, an iterative algorithm was obtained, which leads to the achievement of a threshold value in the horizontal plane of projections. According to the research results, a dynamic visualization of the pursuit task was made, which gave clarity to the results obtained.

1. Introduction
The task of pursuit as a field of the theory of differential games arose as the need to develop applications in military affairs. The first works in this area are connected with the works of R. Isaacs, L.S. Pontryagin, N.N. Krasovsky, L.A. Petrosyan. Further development of the founders of the theory of differential games to the development of applications on the plane (“boat torpedo”) and in space (airplanes) with various modifications of the tasks. In our work we conducted a computer simulation of the pursuit problem, when the objects are on surfaces located one above the other. The current state of IT technology allows surface modeling for this task, the power and miniaturization of modern microprocessors allow real-time calculation of the coordinates of drones and mobile robots using the developed algorithms. Modern means of communication allow the required data exchange in the task. The state of computer graphics allows you to visualize the process of prosecution. In this regard, we believe that such a statement of the problem may be new and relevant.

2. Formulation of the problem
This article describes the pursuit task when the pursued object (“Rabbit”) moves across the surface. The pursuing object moves on the surface \( \mathbf{P}_r(u, v) \). The pursuing object moves on the surface \( \mathbf{P}_f(u, v) \). Let \( R_r(T) \) – be the position point of the Rabbit on the surface \( \mathbf{P}_r(u, v) \), at time \( T \), and \( \bar{R}_f(T) \) – be the point of the position of Fox on the surface \( \mathbf{P}_f(u, v) \), at same time. In points \( \bar{R}_r(T) \) and
\( \vec{R}_f(T) \) surfaces \( \vec{P}_r(u,v) \) and \( \vec{P}_f(u,v) \), respectively, we construct dynamic local bases \( [\vec{E}_1 \quad \vec{E}_2 \quad \vec{E}_3] \) and \( [\vec{Q}_1 \quad \vec{Q}_2 \quad \vec{Q}_3] \) (Figure 1). Where \( \vec{E}_1 = \frac{1}{v_r} \frac{d\vec{R}_r(T)}{dT} \) and \( \vec{Q}_1 = \frac{1}{v_f} \frac{d\vec{R}_f(T)}{dT} \). Where \( V_r \) and \( V_f \) Constant on module of speed "Rabbit" and "Foxes" \( \vec{E}_3 \) and \( \vec{Q}_3 \) – single normals \( \vec{n}_r, \vec{n}_f \) in points \( R_r(T) \) and \( R_f(T) \). And \( \vec{E}_2 = \vec{E}_3 \times \vec{E}_1, \vec{Q}_2 = \vec{Q}_3 \times \vec{Q}_1 \).

Further, we will assume that the plan \( \Sigma \) is formed by the vectors \( \vec{E}_1 \) and \( \vec{E}_2 \), and the plan \( \Pi \) by vectors \( \vec{Q}_1 \) and \( \vec{Q}_2 \). To choose the solution of a "Rabbit" we need to project a point \( \vec{R}_r \) on the plan \( \Sigma \) and get its projection \( \vec{R}'_f \) for analysis, on which quarter of the plan \( \Sigma \) the projection of "Foxes" is approaching. Similarly, to make a decision, “Fox” needs to project a point \( \vec{R}_f \) on the plan \( \Pi \), get her projection \( \vec{R}'_f \), in order to analyze what quarter of the plan \( \Pi \) it is located.

So, the purpose of the article is to develop the algorithm in detail and write a program in the computer mathematics system “MathCAD” for the above pursuit task.

3. Surface modeling

The modeling of the surface \( \vec{P}_r(u,v) \), on which the Rabbit moves is made by specifying points of horizontals line in the «AutoCAD» environment, followed by polynomial regression in the «MathCAD» environment using the built-in functions of the mathematical package. On the surface obtained after polynomial regression, a uniform grid is introduced with the calculation of partial derivatives at the grid nodes.

With the model of the surface on which the Fox is moving, we have a situation like this. To the surface of the "Rabbit" \( \vec{P}_r(u,v) \) equidistant surface is constructed (Figure 2) \( \vec{P}_e(u,v) = \vec{P}_r(u,v) + \Delta \vec{R} \cdot \vec{n}_r \), where \( \Delta \vec{R} \) – the distance that the equidistant surface is separated from the original.

Then a horizontal plane is entered \( z = r_0 \) and the surface \( \vec{P}_f(u,v) \) is composed as follows:

\[
\begin{align*}
\vec{P}_f(u,v) = \begin{cases} 
z, \text{если } \vec{P}_e(u,v)_z \leq r_0 \\
\vec{P}_e(u,v), \text{если } \vec{P}_e(u,v)_z > r_0
\end{cases}
\end{align*}
\]

where \( r_0 \) – some threshold height value (Figure 3).
After that, a uniform grid is introduced on the surface $\vec{P}_f(u,v)$ with the calculation of partial derivatives at the nodes. In Figure 3, the surface $\vec{P}_f(u,v)$ is represented in a translucent form, and the original surface of the "Rabbit" is represented with the designation of horizontals.

4. Construction of projections in local bases

If "Rabbit", moving along the surface $\vec{P}_r(u,v)$, forms a dynamic basis $[\vec{E}_1 \vec{E}_2 \vec{E}_3]$, then the plane $\Sigma$, formed by the vectors $\vec{E}_1$ and $\vec{E}_2$, is a tangent plane to the surface $\vec{P}_r(u,v)$ in point $\vec{R}_r$. The projection of the point $\vec{R}_f$ on the plane $\Sigma$ will be as follows: $\vec{R}_f’ = \vec{R}_f + \frac{\vec{E}_3 (\vec{R}_f - \vec{R}_r)}{\vec{E}_3 \cdot \vec{E}_3}$ (Figure 4) "Fox", moving on the surface $\vec{P}_f(u,v)$, forms a dynamic basis $[\vec{Q}_1 \vec{Q}_2 \vec{Q}_3]$. The projection of the point $\vec{R}_r$ on the plane $\Pi$, formed by the vectors $\vec{Q}_1$ and $\vec{Q}_2$ will be as follows: $\vec{R}_r’ = \vec{R}_r + \frac{\vec{Q}_3 (\vec{R}_r - \vec{R}_f)}{\vec{Q}_3 \cdot \vec{Q}_3}$ (Figure 5).

![Figure 4. "Fox" Projection](image)

![Figure 5. "Rabbit" Projection](image)

Then, in the basis $[\vec{E}_1 \vec{E}_2 \vec{E}_3]$ of the local “Rabbit” coordinate system, the coordinates of the "Fox" projection $\vec{R}_f’$ will be as follows: $\vec{R}_f’ = [(\vec{R}_f’ - \vec{R}_r) \cdot \vec{E}_1]$. In the local “Fox” coordinate system $\vec{R}_r’ = [(\vec{R}_r’ - \vec{R}_f) \cdot \vec{Q}_1]$. The coordinates of the “Rabbit” projection $\vec{R}_r’$ will be as follows: $\vec{R}_r’ = [(\vec{R}_r’ - \vec{R}_r) \cdot \vec{Q}_2]$. It should be noted that $(\vec{R}_r’ - \vec{R}_r) \cdot \vec{Q}_3 = \vec{0}$. 

The basis of the world coordinate system $[\vec{H}_1 \vec{H}_2 \vec{H}_3]$ in the “Rabbit” coordinate system will look like this: $\vec{H}_\Sigma = \frac{\vec{H}_1 \cdot \vec{E}_1}{\vec{H}_1 \cdot \vec{E}_1}$, where $\vec{H}_1 = \frac{\vec{H}_1 \cdot \vec{E}_1}{\vec{H}_1 \cdot \vec{E}_1}$, $\vec{H}_2 = \frac{\vec{H}_2 \cdot \vec{E}_2}{\vec{H}_2 \cdot \vec{E}_2}$, $\vec{H}_3 = \frac{\vec{H}_3 \cdot \vec{E}_3}{\vec{H}_3 \cdot \vec{E}_3}$. And in the
coordinate system of "Fox" will be like this: \( \bar{H}_{\Pi} = \begin{bmatrix} \bar{H}_{1\Pi} \\ \bar{H}_{2\Pi} \\ \bar{H}_{3\Pi} \end{bmatrix} \), where \( \bar{H}_{1\Pi} = \frac{\bar{H}_1 \cdot \bar{E}_1}{\bar{H}_1 \cdot \bar{E}_2} \), \( \bar{H}_{2\Pi} = \frac{\bar{H}_2 \cdot \bar{E}_1}{\bar{H}_3 \cdot \bar{E}_2} \), \( \bar{H}_{3\Pi} = \frac{\bar{H}_3 \cdot \bar{E}_1}{\bar{H}_3 \cdot \bar{E}_3} \).

5. Direction movement selection

We have written a program that, in the calculation cycle at each step, makes a choice in which direction “Rabbit” and “Fox” should move. Let \( \omega_r \) and \( \omega_f \) be the angular velocities of the “Rabbit” and “Foxes”. For example, consider the choice of the direction of movement of the “Rabbit”.

On plane \( \Sigma \), in the “Rabbit” coordinate system, the “Rabbit” has zero coordinates. The analysis of the coordinates of the “Fox” by the “Rabbit” is that if the “Fox” is in the lower half-plane \( \Sigma \), then the “Rabbit” takes a counterclockwise step (Figure 6). If the "Fox" is in the upper half-plane, then the "Rabbit" makes a clockwise step. That is, the new coordinates of the Rabbit over the time interval \( \Delta T \) are equal in the coordinate system of the plane \( \Sigma \):

\[
\vec{R}'_{\Sigma} = \begin{cases} 
V_r \cdot \Delta T \cdot \begin{bmatrix} \cos(\omega_r \cdot \Delta T) \\ \sin(\omega_r \cdot \Delta T) \\ 0 \end{bmatrix}, & \text{if } (\vec{R}_f - \vec{R}_r) \cdot \vec{E}_2 < 0 \\
V_r \cdot \Delta T \cdot \begin{bmatrix} -\sin(\omega_r \cdot \Delta T) \\ \cos(\omega_r \cdot \Delta T) \\ 0 \end{bmatrix}, & \text{if } (\vec{R}_f - \vec{R}_r) \cdot \vec{E}_2 \geq 0
\end{cases}.
\]

Similar can be said about the choice of the direction of movement "Fox". If “Rabbit” is in the upper half-plane \( \Pi \), then “Fox” takes a counterclockwise step (Figure 7). If in the lower half-plane, then clockwise:

\[
\vec{R}'_{\Pi} = \begin{cases} 
V_f \cdot \Delta T \cdot \begin{bmatrix} \cos(\omega_f \cdot \Delta T) \\ -\sin(\omega_f \cdot \Delta T) \\ 0 \end{bmatrix}, & \text{if } (\vec{R}_r - \vec{R}_f) \cdot \vec{Q}_2 < 0 \\
V_f \cdot \Delta T \cdot \begin{bmatrix} \sin(\omega_f \cdot \Delta T) \\ \cos(\omega_f \cdot \Delta T) \\ 0 \end{bmatrix}, & \text{if } (\vec{R}_r - \vec{R}_f) \cdot \vec{Q}_2 \geq 0
\end{cases}.
\]
6. Calculation of coordinates in the next stages of iterations

Earlier, we considered how the basis \( [\vec{H}_1 \ \vec{H}_2 \ \vec{H}_3] \) of the world coordinate system in local dynamic bases \( [\vec{E}_1 \ \vec{E}_2 \ \vec{E}_3] \) and \( [\vec{Q}_1 \ \vec{Q}_2 \ \vec{Q}_3] \) would look. Then, in the world coordinate system \( [\vec{H}_1 \ \vec{H}_2 \ \vec{H}_3] \), the new coordinates of the “Rabbit” \( \vec{R}_{r\Sigma} \) on plane \( \Sigma \) will be as follows:

\[
\vec{R}_{r\Sigma} = \vec{R}_r + \begin{bmatrix}
\vec{R}'_{r\Sigma} \cdot \vec{H}_1\Sigma \\
\vec{R}'_{r\Sigma} \cdot \vec{H}_2\Sigma \\
\vec{R}'_{r\Sigma} \cdot \vec{H}_3\Sigma
\end{bmatrix}
\]

And the new coordinates of "Fox" \( \vec{R}_{f\Pi} \) on the plane \( \Pi \), at the same stage of the iterations, will look like this in the world coordinate system:

\[
\vec{R}_{f\Pi} = \vec{R}_f + \begin{bmatrix}
\vec{R}'_{f\Pi} \cdot \vec{H}_1\Pi \\
\vec{R}'_{f\Pi} \cdot \vec{H}_2\Pi \\
\vec{R}'_{f\Pi} \cdot \vec{H}_3\Pi
\end{bmatrix}
\]

We need vertical projections of points \( \vec{R}_{r\Sigma} \), \( \vec{R}_{f\Pi} \) on the surface \( \vec{P}_r(u, v) \) and \( \vec{P}_f(u, v) \), respectively. The surfaces \( \vec{P}_r(u, v) \) and \( \vec{P}_f(u, v) \), on which Rabbit and Fox move can always be converted to: \( \vec{P}_r(x, y) = \begin{bmatrix} x & y \\ S_r(x, y) \end{bmatrix} \) and \( \vec{P}_f(x, y) = \begin{bmatrix} x & y \\ S_f(x, y) \end{bmatrix} \). If the coordinates of the points \( \vec{R}_{r\Sigma} \) and \( \vec{R}_{f\Pi} \), \( \vec{R}_{fox\_new} \), we will be presented as:

\[
\vec{R}_{r\Sigma} = \begin{bmatrix} x_{r\Sigma} \\
y_{r\Sigma} \\
S_r(x_{r\Sigma}, y_{r\Sigma}) \end{bmatrix}, \quad \vec{R}_{f\Pi} = \begin{bmatrix} x_{f\Pi} \\
y_{f\Pi} \\
S_f(x_{f\Pi}, y_{f\Pi}) \end{bmatrix}
\]

Thus, we considered the iteration process in the pursuit task we set, is formed. The criterion for achieving the goal in the program developed by us is the distance between the Rabbit and the Fox in the horizontal plane of the projections: \( (x_{r\Sigma} - x_{f\Pi})^2 + (y_{r\Sigma} - y_{f\Pi})^2 = R_0 \), where \( R_0 \) - some threshold value, which we have installed.

7. Results

On the basis of [1], [2], [3], [4], we proposed a mathematical model of the pursuit problem, when task participants are on different surfaces. Approximately can be considered an analogue of pursuit from the air along a trajectory that takes into account the terrain. An algorithm for constructing trajectories, depending on the spatial location of opponents. A method for analyzing the coordinates of the opponent and making decisions about the choice of direction of movement is proposed.

Figure 8. “Quasi equidistant” chase

In the system of computer mathematics "MathCAD" a program has been developed that implements this model of the pursuit problem. The full text of the program with detailed comments...
can be found on the website of the author [5]. Also, there is an archive file. According to the results of
the program, an animated image was made, which you can view on the author's channel [6]. Figure 8
shows the first frame of the pursuit process. In the left part of the picture, the surface of the “Fox” is
not displayed, only the paths of the “Fox”, the “Rabbit” and the surface of the “Rabbit” are visible. On
the right side of the picture, the surface of the “Fox” is displayed in a translucent form.

8. Conclusion

This article addressed the challenge of pursuit, in which participants are on different surfaces. The
theoretical aspects of the “driver-killer” problem (“boat torpedo”) were considered in the works of
Rufus Isaacs [1]. In the works of L.S. Pontryagin considered the theoretical issues of the pursuit
problem in space (airplanes) [2]. In [3], the linearized pursuit problem on a plane with constant
velocities and a bounded curvature of the trajectories was considered. In [4], algorithmic issues of
controlling mobile robots in the pursuit problem on the plane were considered. The authors proposed a
technique for modeling the behavior of objects, participants in the pursuit task, in which there are
elements of artificial intelligence. The statement of the problem in this article suggests further
development in terms of predicting trajectories, obtaining information about the opponent’s intended
actions in the pursuit task and in terms of taking into account the physical characteristics of the
environment in which the pursuit takes place.

We believe that the results of the study outlined in this article can be claimed by the developers of
robotic complexes with elements of artificial intelligence.

9. Acknowledgement

The article was funded by innovative grant of the Buryat State University in 2019 "Hardware and
software complex of persecution"

References
[1] Isaacs R 1999 Differential games (New York: John Wiley and Sons)
[2] Pontryagin L S 1971 Linear differential runaway game Trudy MIAN USSR 112 pp 30-63.
[3] Zhelnin Yu N 1977 Linearized task of pursuing and evading on a plane Scholarly notes TsAGI 3
    pp 88-98
[4] Burdakov S V and Sizov P A 2014 Algorithms for controlling the movement of a mobile robot
    in the pursuit problem Scientific and technical statements of the St. Petersburg State
    Polytechnic University. Computer science. Telecommunications. Management 6 (210) pp 49-58
[5] http://dubanov.exponenta.ru Section "Movement" Foxes" by "quasi-equidistant"
[6] Video "Movement" Foxes "on the quasi-equalist" https://www.youtube.com/watch?v=w2HSwKpB63w