

Application of Piecewise Linear Approximation of the UAV Trajectory for Adaptive Routing in FANET

Kuzichkin Oleg R.^{1,*}, Vasilyev Gleb S.¹, Surzhik Dmitry I.², Kurilov Igor A.²

¹Belgorod State Research University, Belgorod, 308015, Russia

²Vladimir State University named after Alexander Grigoryevich and Nikolai Grigorievich Stoletovs, Vladimir, 600000, Russia

ARTICLE INFO

Article history:

Received: 03 February, 2021

Accepted: 04 March, 2021

Online: 20 March, 2021

Keywords:

Unmanned aerial vehicle

UAVs

Flying Ad Hoc Network

FANET

Routing

Trajectory approximation

Piecewise linear function

ABSTRACT

A significant problem of routing protocols in the Flying Ad Hoc Networks (FANET) is a significant overhead cost due to the high mobility of networking nodes. The problem is caused by a need to send information messages about locations of unmanned aerial vehicles (UAVs). In order to reduce the amount of service information, the following trajectory approximation algorithms have been investigated: an algorithm for conjugating courses and an algorithm based on continuous piecewise-linear functions (CPLF). Four modifications of the CPLF-based algorithm are considered, which differ in the type of piecewise linear function used: basic CPLF, generalized CPLF, generalized CPLF with a compact notation form, and adaptive CPLF. The disadvantages of each algorithm are analyzed. The CPLF approximation of a fragment of an aircraft trajectory consisting of two straight sections and a curved section with variable steepness between them is performed. It is established that adaptive CPLF with variable step reduces the error of trajectory approximation due to the location of most points on the curved sections of the aircraft maneuvering. The modified version of ADV routing protocol has shown a lower overhead value (the gain for small pause time values reaches 23 %). Thus, the effectiveness of the proposed approximation-based routing in FANET is shown.

1. Introduction

This paper is an extension of work originally presented in the 7th International Conference on Control, Decision and Information Technologies (CoDIT' 20) [1].

Networks built on the basis of unmanned aerial vehicles (UAVs) (Flying Ad Hoc Network, FANET) [2] are characterized by an unstable network topology due to the movement of network nodes. This aspect should be taken into account when developing and applying routing algorithms. The well-known network layer protocols do not fully meet the requirements of the FANET, so the development of new routing protocols for flying networks remains relevant.

Proactive protocols calculate routes between all network nodes beforehand and maintain a relevant routing table at each node [3]. An integral part of such routing protocols is a connection management mechanism that establishes logical connections between network nodes that are within earshot of each other, and also closes logical connections. All stations distribute information

about the established logical connections over the network. Thus, each of the stations decentralizes the logical topology of the network and fills in its own routing table. In addition, proactive routing protocols provide minimal delay in the delivery of data packets, and when the network topology changes, broadcast messages about these changes are initiated [4].

The author proposed a reactive protocol Dynamic Source Routing (DSR) [5]. In reactive routing protocols, routes exist only when they are needed, that is, when data is being transmitted over them. When the route is initially determined, packets are sent in all possible directions, and information about the passed node is added to the header. As a result, when the goal is reached, the packet header contains a fully formed route between the specified nodes. In the event of loops, i.e. re-reception of the first packet, the node destroys this packet. The high mobility of FANET nodes when using proactive protocols requires frequent updates of the routing table, which is their disadvantage.

Hybrid routing protocols combine the advantages of reactive and proactive routing protocols, but require a relatively complex implementation. Their use is associated with the need to divide the

*Corresponding Author: Kuzichkin O.R., Contact No: 910-171-3945, Email: oldolkuz@yandex.ru

network structure into segments, which reduces the efficiency of routing (for example, HWMP-Hybrid Wireless Mesh Protocol).

Geographic routing protocols use the spatial coordinates of network nodes and are more efficient for FANET, especially when the network is large. An example is the Greedy Perimeter Stateless Routing protocol (GPSR) [6]. Directional antennas-based protocols reduce the number of transit nodes and, as a result, lower the delay time (DOLSR-Directional Optimized Link State Routing Protocol [7]).

A significant problem of routing protocols in the Flying Ad Hoc Networks (FANET) is a significant overhead cost due to the high mobility of networking nodes. The problem is caused by a need to send information messages between nodes about locations of separate unmanned aerial vehicles (UAVs) [8]. Also the use of directional antennas on board the UAV [9-11] necessitates additional transmission of the angular coordinates of each node, since when the airplane turns, the receiving antenna may deviate from the maximum of the radiation pattern, which will reduce the useful signal.

In [1] the application of UAV trajectory approximation for adaptive routing in FANET networks is justified. However, the expressions are given in a fairly brief form. In addition, the approximation example of a trajectory fragment in the form of a sinusoidal function does not allow us to fully evaluate the advantages of the proposed approach, for which it is necessary to model a trajectory consisting of a set of straight sections and curved sections with variable steepness.

2. The algorithm for conjugated courses

Approximation of the UAV trajectory will reduce the overhead caused by the transmission of additional information about the position of network nodes. The use of trajectory approximation algorithms will allow the vehicle coordinates to be transmitted only at the beginning or end of each maneuver. With this approach, it is not necessary to transmit data about intermediate points of the trajectory between maneuvers, which reduces overhead costs.

The creation of the trajectory of an flying vehicle consists of several stages [12]. The first is to set the motion kinematics, which determines the spatial position of the object at an arbitrary time t. For this purpose, the object is assigned a trajectory that can be set by a set of reference vertices with a velocity vector defined at each vertex. Next, the trajectory of the air object is plotted along the reference verteces. The main disadvantage of the method of conjugating courses is a rather large amount of calculations and, accordingly, a decrease in the speed of the algorithm.

The course matching algorithm is based on dividing the trajectory into segments, and each segment is described as a mathematical expression [13]. The input data of the algorithm are the coordinates of the node points of the trajectory. The rectilinear motion of the aircraft is described by parametric equations:

$$x = x_0 + \mathcal{G}_{0x}t + \frac{a_x t^2}{2}, y = y_0 + \mathcal{G}_{0y}t + \frac{a_y t^2}{2}, \quad (1)$$

where x and y are current spacial coordinates, x₀ and y₀ are initial values of spacial coordinates, G_{0x,y} are projections of velocity on x

and y axis, a_x and a_y are projections of acceleration on x and y axis, accordingly.

When calculating straight segments of the trajectory, the minimum trajectory curvature radius with a nominal (acceptable) level of overload, it is necessary to calculate the coefficient K(φ_n)

$$K(\varphi_n) = \begin{cases} \frac{1,017456}{\sqrt{tg \frac{\varphi_n}{2}}} \cdot \left(1 + \frac{1}{3 \cdot tg^2 \frac{\varphi_n}{2}} \right) & \text{at } \varphi_n \geq \frac{\pi}{2} \\ \left(0,272166 + \frac{1,014303}{tg \frac{\varphi_n}{2}} \right) & \text{at } \varphi_n < \frac{\pi}{2} \end{cases} \quad (2)$$

where φ_n is course at the segment n of a trajectory.

For circular segments of the trajectory, when calculating speed, one might need to calculate the sine, cosine:

$$\mathcal{G}(t) = \frac{(\varepsilon \cdot (t - t_0) + \omega_0)^2}{\sqrt{x^2 + y^2 + z^2}} \cdot \begin{bmatrix} \cos(\psi(t)) \cdot (x \cdot A_x + y \cdot A_y + z \cdot A_z) \\ -\sin(\psi(t)) \cdot (x \cdot B_x + y \cdot B_y + z \cdot B_z) \end{bmatrix} \quad (3)$$

where G(t) is current radial velocity of an aircraft, ε is circular acceleration, t and t₀ is current and initial time, ω₀ is initial circular velocity, x, y, and z are spatial coordinates, ψ is current current maneuvering angle, A_{x,y,z} and B_{x,y,z} are coefficients of the movement.

For parabolic segments when calculating the maximum move in a parabola, there is a need also calculate the arc sine:

$$\psi_n = 2 \arcsin \left[3tg \frac{\varphi_n}{2} \sqrt{\frac{tg^2 \frac{\varphi_n}{2} + \frac{1}{9}}{\left(tg^2 \frac{\varphi_n}{2} - \frac{7}{72} \right)^2 + 36tg^2 \frac{\varphi_n}{2} \cdot \left(tg^2 \frac{\varphi_n}{2} + \frac{31}{144} \right)^2}} \right], \quad (4)$$

where ψ_n and φ_n are current maneuvering angle and course at the segment n of a trajectory.

Expressions (1-4) describe a smooth trajectory with high accuracy, which is necessary in automatic control systems [14] and simulation systems of full-scale modeling of aircraft [12, 15]. However, routing algorithms do not require such high accuracy in describing the path. The main requirement for the routing algorithm is the choice of optimal transit nodes for a specific network topology, which can be successfully performed with small errors in the representation of the trajectory. Thus, for routing algorithms the very detailed trajectory model is redundant, computationaly complex and not required.

3. Algorithms based on continuous piecewise linear functions

Studies have shown that continuous piecewise linear functions (CPLF) based on the modulus of a linear function are effective for approximating nonlinear characteristics, trajectories [16-18]. Calculating the approximation coefficients and the function itself

requires less computational resources compared to the course conjugation algorithm.

A smaller value of the error is achieved at the expense of equality of error in every area of approximation. The use of this approach when approximating straight lines leads to a decrease in the error, since the approximating function "adjusts" (adapts) to the change in the nature of the nonlinearity of the original characteristic.

The approximation of the trajectory using the adaptive CPLF is schematically shown in Figure 1. The function section is divided into segments. The approximation nodes are calculated as follows: a fixed error value is set, then the number of segments (approximation nodes) is determined by the algorithm. The length of segment xi varies depending on the value of approximation error ϵ .

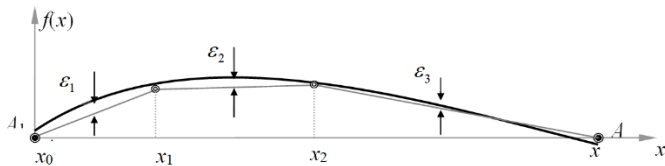


Figure 1: The principle of trajectory approximation using CPLF

The CPLF approximation assumes the replacement of the object's trajectory along an arc with the trajectory of movement along a straight line. The time and velocity modules are selected according to the reference algorithm [12]. As a result, in both cases (trajectories $f(x)$ and F_i) start and end speed is the same and time spent is also the same. This means that each point of the reference segment is projected onto the corresponding point of the CPLF segment. For example, consider a section of the trajectory of Figure 1 in the interval $[x_0, x_1]$. It is shown in Figure 2. It can be seen that the error in determining the position of the object on the segment is equal to the maximum length of the projection line. In the adaptive CPLF, the errors ϵ_i are aligned with each other. The geometric meaning is shown in Figure 3.

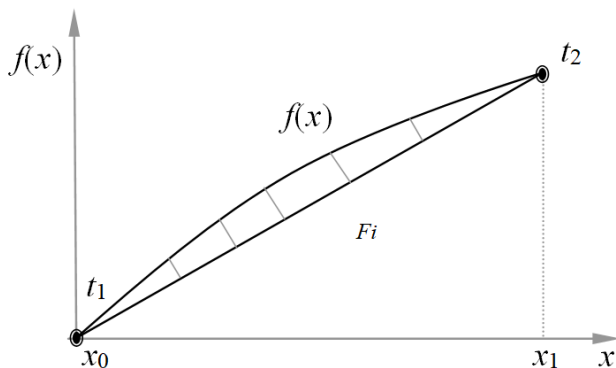


Figure 2: Fragment of the trajectory

The actual error is the distance between the position of the object on the reference curve and its position on the approximating curve. Figure 3 shows the discrepancy between the actual error and the approximation error. Knowing that $\epsilon_1 = \epsilon_2 = \dots = \epsilon_n = \dots = \epsilon_{N-1}$ we estimate the range of changes in the error value δ_i . From the right triangle ABC in Figure 4, we obtain $\sin(\pi/2 - \alpha_i) = \delta_i / \epsilon_i$,

where α_i is the angle between the CPLF and the reference trajectory. Thus

$$\delta_i = \epsilon_i \sin\left(\frac{\pi}{2} - \alpha_i\right). \tag{5}$$

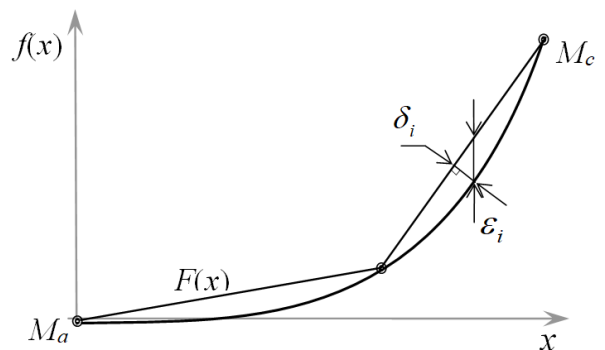


Figure 3: Determination of the approximation error

To estimate the error δ_i we need to know the error α_i . The minimum value $\alpha_0 = 0^\circ$ (at the origin in Figure 4), $\delta_{i\min} = \epsilon_i \sin\left(\frac{\pi}{2}\right)$. The maximum value $\alpha_i = \frac{\pi}{2} - \frac{\varphi}{2}$ and $\delta_i = \epsilon_i \sin(\varphi/2)$, therefore the actual error δ_i is determined by the inequalities $\delta_{i\min} < \delta_i < \delta_{i\max}$ and $\sin \frac{\pi}{2} < \delta_i < \epsilon_i \sin\left(\frac{\varphi}{2}\right)$. It follows from the inequality that the greater the value of φ , the smaller the error δ_i , and δ_i (the side of the triangle ABC) does not exceed ϵ_i . This means that ϵ_i is the maximum error of the approximation of the reference trajectory using the CPLF.

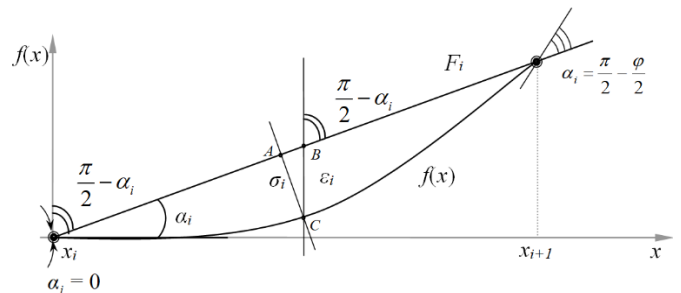


Figure 4: Determination of the linear and angular approximation error

3.1. Basic CPLF

It has a compact form and contains only one approximation coefficient. The basic CPLF is convenient for representing functions and nonlinear characteristics approximated with a fixed argument step, with a small number of approximation nodes. The base CPLF has the form:

$$f_M(t) = b_M |t - t_0 - \Delta t \cdot M|, \tag{6}$$

where t_0 is an initial shift of the CPLF relative to the origin;

Δt is a shift of the modules relatively to each other;

$M=0,1,2,3$ is a count of the number of modules;

b_M is approximation coefficient.

Summing up the four functions shifted relative to each other by a fixed value, we get the basic CPLF (Figure 5):

$$b_0 \cdot |t - t_0| + b_1 \cdot |t - t_0 - \Delta t| + b_2 \cdot |t - t_0 - 2 \cdot \Delta t| + b_3 \cdot |t - t_0 - 3 \cdot \Delta t| = f(t) \quad (7)$$

We determine the coefficients b_M . To do this, using the four points $(t_0; 0)$; $(t_0 + \Delta t; y_1)$; $(t_0 + 2 \cdot \Delta t; y_2)$; $(t_0 + 3 \cdot \Delta t; 0)$ shown in the figure, we make a system of four equations:

$$\begin{cases} b_0 \cdot |0| + b_1 \cdot |-\Delta t| + b_2 \cdot |-2 \cdot \Delta t| + b_3 \cdot |-3 \cdot \Delta t| = 0 \\ b_0 \cdot |\Delta t| + b_1 \cdot |0| + b_2 \cdot |-\Delta t| + b_3 \cdot |-2 \cdot \Delta t| = y_1 \\ b_0 \cdot |2\Delta t| + b_1 \cdot |\Delta t| + b_2 \cdot |0| + b_3 \cdot |-\Delta t| = y_2 \\ b_0 \cdot |3\Delta t| + b_1 \cdot |2\Delta t| + b_2 \cdot |\Delta t| + b_3 \cdot |0| = 0. \end{cases} \quad (8)$$

After the transformations, the system takes the form:

$$\begin{cases} 0 + \Delta t \cdot b_1 + 2 \cdot \Delta t \cdot b_2 + 3 \cdot \Delta t \cdot b_3 = 0 \\ \Delta t \cdot b_0 + 0 + \Delta t \cdot b_2 + 2 \cdot \Delta t \cdot b_3 = y_1 \\ 2 \cdot \Delta t \cdot b_0 + \Delta t \cdot b_1 + 0 + \Delta t \cdot b_3 = y_2 \\ 3 \cdot \Delta t \cdot b_1 + 2 \cdot \Delta t \cdot b_2 + \Delta t \cdot b_3 + 0 = 0. \end{cases} \quad (9)$$

Solving system (9) by the Gauss method [20], we obtain:

$$b_0 = \frac{y_1}{2 \cdot \Delta t}; \quad b_1 = \frac{y_2 - 2 \cdot y_1}{2 \cdot \Delta t}; \quad b_2 = \frac{y_1 - 2 \cdot y_2}{2 \cdot \Delta t}; \quad b_3 = \frac{y_2}{2 \cdot \Delta t}, \quad (10)$$

where y_1, y_2 are the function values in the approximation nodes.

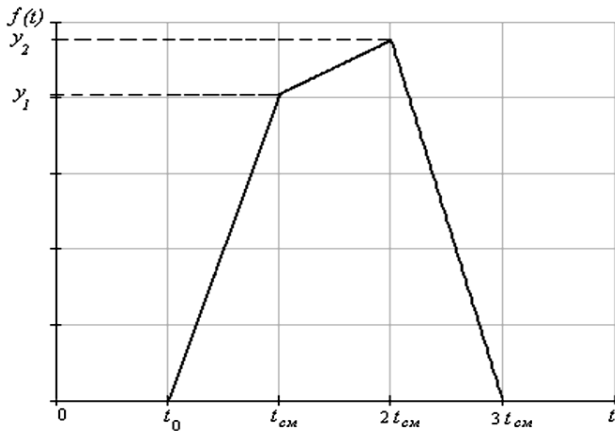


Figure 5: Basic CPLF

In order not to use separate expressions for each coefficient b_0, b_1, b_2, b_3 , we present them as a single coefficient B_M :

$$\begin{aligned} B_M = & \frac{(M-1)(M-2)(M-3)}{-3} \cdot \frac{y_1}{2 \cdot \Delta t} + \\ & + \frac{M(M-2)(M-3)}{1} \cdot \left(\frac{y_2 - 2 \cdot y_1}{2 \cdot \Delta t} \right) + \\ & + \frac{M(M-1)(M-3)}{-1} \cdot \left(\frac{y_1 - 2 \cdot y_2}{2 \cdot \Delta t} \right) + \\ & + \frac{M(M-1)(M-2)}{3} \cdot \frac{y_2}{2 \cdot \Delta t}. \end{aligned} \quad (11)$$

We transform equation (11):

$$B_M = y_2 \frac{5M^3 - 21M^2 + 19M}{6 \cdot \Delta t} - y_1 \frac{5M^3 - 24M^2 + 28M - 3}{6 \cdot \Delta t}. \quad (12)$$

Substituting the coefficient B_M , we finally get:

$$f(t) = \sum_{M=0}^3 B_M |t - t_0 - \Delta t \cdot M|. \quad (13)$$

The disadvantages of the basic CPLF include a cumbersome expression for determining the approximation coefficients B_M . In addition, to approximate functions over a large number of points (more than three), it is necessary to add up individual base CPLFS shifted between each other.

3.2. Generalized CPLF

The generalized CPLF reduces the computational cost of approximation using the basic CPLF. This function is the sum of a few basic CPLF. Using the generalized CPLF, we can approximate functions of any kind based on a single expression for the approximating function, rather than a large number of expressions:

$$F(t) = \sum_{n=1}^N \sum_{M=0}^3 B_{M,n} |t - [t_0 + 2 \cdot \Delta t(n-1)] - \Delta t \cdot M|, \quad (14)$$

where

$$B_{M,n} = y_{n2} \frac{5M^3 - 21M^2 + 19M}{6 \cdot \Delta t} - y_{n1} \frac{5M^3 - 24M^2 + 28M - 3}{6 \cdot \Delta t}$$

is the approximation coefficient, n is the number of basic CPLF for the approximation; $t_0 + 2 \cdot \Delta t$ is the initial offset of the second approximating function; $y_{n1} = y(t_0 + \Delta t \cdot (2n-1))$ are values of the initial characteristic in odd approximation nodes;

$y_{n2} = y(t_0 + \Delta t \cdot 2n)$ are values of the initial characteristic in even approximation nodes.

The disadvantages of the generalized CPLF include additional terms under the module and two sum counters, which complicates the computational process. A generalized CPLF can have a compact notation form and include only one sign of the sum.

3.3. Generalized CPLF with compact notation

We design a function that has a compact expression and allows us to perform an approximation with an arbitrary step. It is the sum of the modules of linear functions with different coefficients in front of them. Expressions under modules represent shifts of linear functions in the time domain. These shifts determine the approximation step. Thus, shifting the individual linear functions by a different value t_n we get the CPLF:

$$F(t) = \sum_{n=1}^N K_n \cdot |t - t_n|, \quad (15)$$

where n is the sum counter; t_n is the approximation step; K_n is the approximation coefficient.

We define an expression for finding the coefficient K_n . To do this, we approximate the function $y = f(t)$ $n = N$ points. Let the first approximation node correspond to the value $t = t_1$,

and the value of the approximated function $y_1 = f(t_1)$. The expression for the CPLF at this point:

$$F(t_1) = \sum_{n=1}^N K_n \cdot |t_1 - t_n| \quad (16)$$

or

$$\sum_{n=1}^N K_n \cdot |t_1 - t_n| = y_1. \quad (17)$$

The law of motion along each straight segment can be given by a function of the parameter t :

$$x(t) = v_{0x}t + \frac{a_x t^2}{2}, \quad y(t) = v_{0y}t + \frac{a_y t^2}{2}, \quad z(t) = v_{0z}t + \frac{a_z t^2}{2}. \quad (18)$$

By decomposing the expression for the CPLF into separate modules we get:

$$K_1 \cdot |t_1 - t_1| + K_2 \cdot |t_1 - t_2| + \dots + K_{N-1} \cdot |t_1 - t_{N-1}| + K_N \cdot |t_1 - t_N| = y_1. \quad (19)$$

We perform a similar operation for the second node $t = t_2$, which corresponds to the value of the approximated function $y_2 = f(t_2)$ and the expression for the adaptive CPLF at this node:

$$F(t_2) = \sum_{n=1}^N K_n \cdot |t_2 - t_n| \quad (20)$$

or

$$\sum_{n=1}^N K_n \cdot |t_2 - t_n| = y_2. \quad (21)$$

As a result, for N approximation nodes we obtain a system of N equations:

$$\begin{cases} K_1|t_1 - t_1| + K_2|t_1 - t_2| + \dots + K_{N-1}|t_1 - t_{N-1}| + K_N|t_1 - t_N| = y_1 \\ K_1|t_2 - t_1| + K_2|t_2 - t_2| + \dots + K_{N-1}|t_2 - t_{N-1}| + K_N|t_2 - t_N| = y_2 \\ \vdots \\ K_1|t_{N-1} - t_1| + K_2|t_{N-1} - t_2| + \dots + K_{N-1}|t_{N-1} - t_{N-1}| + K_N|t_{N-1} - t_N| = y_{N-1} \\ K_1 \cdot |t_N - t_1| + K_2|t_N - t_2| + \dots + K_{N-1}|t_N - t_{N-1}| + K_N|t_N - t_N| = y_N. \end{cases} \quad (22)$$

We find the values of the coefficients $K_1 - K_N$ by the inverse matrix method. In matrix form, the system will take the form:

$$[K] \cdot [T] = [y]. \quad (23)$$

Hence, the desired coefficients $K_1 - K_N$ have the form:

$$[K] = [y] \cdot [T]^{-1}. \quad (24)$$

3.4. Adaptive CPLF

Approximating trajectories with a fixed step is impractical if the trajectories contain long straight sections. To represent each linear fragment, two approximation points are sufficient, and it is advisable to place most of the approximation points on curved sections. Thus, it is promising to apply an adaptive CPLF with a variable approximation step.

As a criterion for the quality of the approximation, we use the minimum mean square error (MSE) criterion. The expression for the approximation MSE in the range (t_0, t_N) has the form

$$\sigma^2 = \int_{t_0}^{t_N} [y(t) - F(t)]^2 dt. \quad (25)$$

The approximation is carried out based on finding the vector $\mathbf{T} = (t_1 \dots t_{N-1})^T$ which fits to the condition of the minimum MSE value:

$$\sigma^2(\mathbf{T}) \rightarrow \min. \quad (26)$$

Adaptive CPLF allows to approximate nonlinear characteristics and functions with a lower error than CPLF with a fixed approximation step. This is achieved by obtaining an equal error in each approximation section. The use of this approach when approximating with lines leads to a decrease in the error, since the approximating function "adjusts" (adapts) to the change in the nature of the nonlinearity of the original characteristic.

Equal approximation error in each section will reduce the total error. To obtain an equal approximation error in each section, the following condition must be met:

$$\delta_1 = \delta_2 = \dots = \delta_n. \quad (27)$$

Based on the MSE expressed by Equation (25), we write:

$$\int_{t_0}^{t_1} [y(t) - F(t)]^2 dt = \int_{t_1}^{t_2} [y(t) - F(t)]^2 dt = \dots = \int_{t_{N-1}}^{t_N} [y(t) - F(t)]^2 dt. \quad (28)$$

Let's represent the integrand expressed in Equation (28) as follows:

$$g(t) = [y(t) - F(t)]^2. \quad (29)$$

$$\delta = \int_{t_i}^{t_k} g(t) dt = G \Big|_{t_i}^{t_k} = G(t_k) - G(t_i), \quad (30)$$

where G is antiderivative of the function $g(t)$.

We find the points for the approximation of an arbitrary function based on the adaptive CPLF with a lower error. To find these points, we define a system of equations:

$$\begin{cases} G(t_1) - G(t_0) = G(t_2) - G(t_1) \\ G(t_1) - G(t_0) = G(t_3) - G(t_2) \\ \vdots \\ G(t_1) - G(t_0) = G(t_N) - G(t_{N-1}). \end{cases} \quad (31)$$

From the system of equations (31), the values of t_n are determined, which approximate the studied characteristic with an equal error in each section. The values of the function $y(t_n)$ at the points t_n are determined from the known expression for the original function $y(t)$.

4. Approximation of the trajectory fragment of the aircraft based on the CPLF

The solution of the minimization problem (26) was performed on the basis of the nonlinear conjugate gradient method [19,20],

the value of convergence tolerance parameter of 10^{-6} was accepted. Evaluation of the approximation MSE based on Equation (25) is shown in Fig. 3.

The approximation result of the aircraft trajectory fragment $\tilde{F}_{1,2}(t)$ by $N=16$ nodes and the absolute approximation error $\Delta_{F1,2}(t)$ are shown in Figures 6 and 7. Here, the lower index 1 corresponds to the uniform law for the approximation nodes, and the index 2 corresponds to the optimal law. The dots in the figures show the approximation nodes. The values of the source function and the approximating function at the nodes are equal, so the error at these points is zero. The MSE values are $\sigma^2_{F1}=1,187 \cdot 10^{-3}$; $\sigma^2_{F2}=1,982 \cdot 10^{-5}$. With the optimal nodes, most of them are located on a curved section, where the trajectory changes most quickly, which gives a significant decrease of MSE (by 59.9 times).

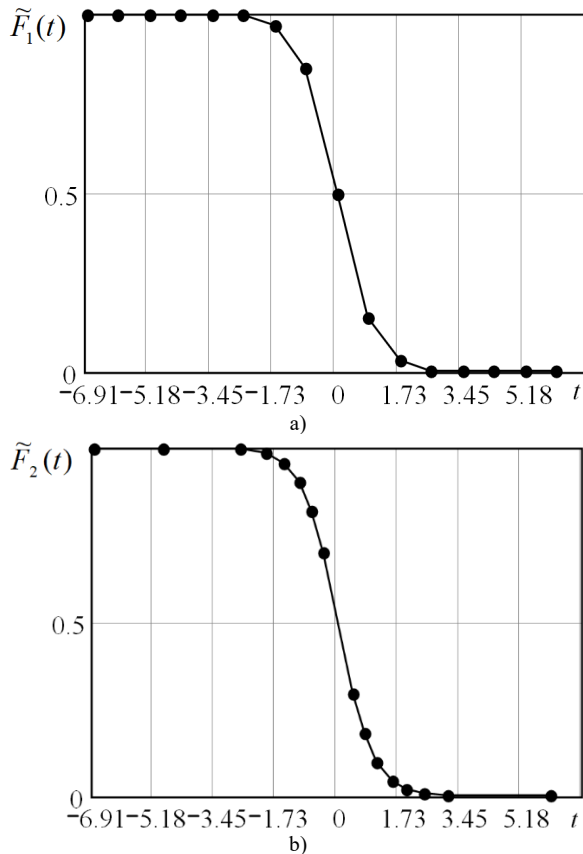


Figure 6: Approximation of the aircraft trajectory fragment based on the CPLF: a) with a uniform step; b) with an adaptive step

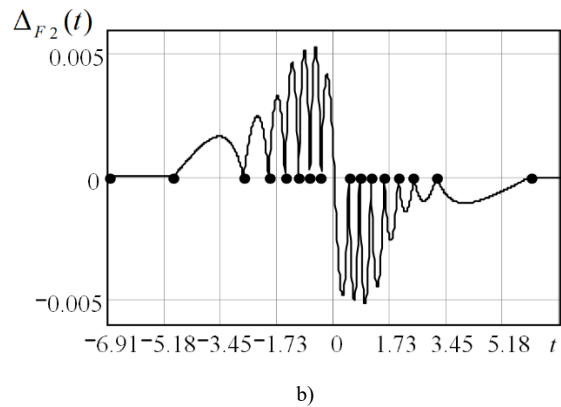
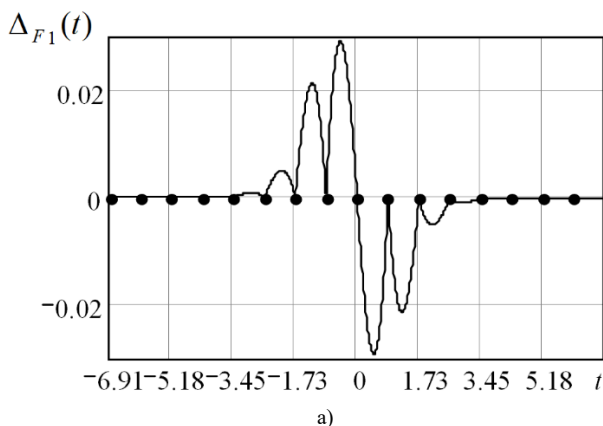


Figure 7: Approximation error of the aircraft trajectory fragment based on the CPLF: a) with a uniform step; b) with an adaptive step

5. Modeling of approximation-based routing in FANET

A comparative simulation of the UAV network with two routing protocols was performed: the standard ad-hoc on-demand vector (AODV) and the modified one. In the modified version of the protocol, Route Request packets were not transmitted at a fixed time interval, as in the standard version, but only when the course changed by more than 200. Otherwise, Route Request packets were not transmitted, and the coordinates of the nodes obtained from the past coordinate values by the piecewise linear approximation were used to update the route table. Changes in the course of individual nodes were taken into account in the node mobility model (the OPNET modeler 14.5 simulator was used). Thus, the modified routing protocol is a hybrid of the standard AODV protocol and geographic-based routing. When modeling the network, the following parameters were obtained:

- the number of network nodes - 20;
- channel bandwidth - 2 Mbit/s;
- the power of the UAV receiving and transmitting device is 10 mW;
- threshold receiving power-minus 90 dB;
- the maximum data packet size is 1024 bytes;
- MAC layer protocol - CSMA/CA (carrier sense multiple access with collision avoidance)
- network size 10x10 km;
- average speed of mobile nodes is 10 m/s.

Mobility model - Random-Waypoint [21]

During the simulation, the Pause time parameter of a mobility model (time between changes in direction and/or speed) was varied and the routing overhead in percent was determined for the standard and modified AODV. The simulation results are shown in Figure 8. The overhead value of the standard AODV is in good agreement with the results in [22]. The modified version of AODV routing protocol shows a lower overhead value (the gain for small pause time values reaches 23 %). Thus, the effectiveness of the proposed approximation-based routing in FANET is shown.

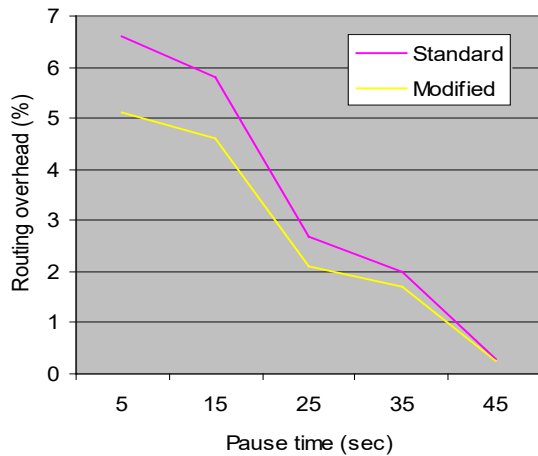


Figure 8: Routing overhead vs Pause time for a standard and modified AODV

6. Conclusion

A serious problem of routing protocols in FANET is the significant overhead caused by the high mobility of network nodes. The problem is caused by a need to send information messages about locations of unmanned aerial vehicles (UAVs). The perspective of algorithms for constructing the UAV trajectory to reduce the amount of service information about the UAV position transmitted via the FANET routing protocols is substantiated. When using such algorithms, information about the coordinates of the nodes is directly transmitted only in the areas of maneuvering of the aircraft, and in the areas of rectilinear motion is calculated based on the algorithms for approximating the trajectory. In order to reduce the amount of service information, the following trajectory approximation algorithms have been investigated: an algorithm for conjugating courses and an algorithm based on continuous piecewise-linear functions (CPLF). Four modifications of the CPLF algorithm are considered which differ in the type of function used: basic CPLF, generalized CPLF, generalized CPLF with a compact notation form, and adaptive CPLF. The advantages and disadvantages of each algorithm are analyzed. Adaptive CPLF with variable approximation step allows lower computational resources compared to the rate-matching algorithm, as well as more accurate compared to the CPLF with a fixed step.

An approximation of a trajectory fragment consisting of two straight sections and a curved section with a variable steepness between them (approximately described by the arctangensoidal function) was performed. The computational experiment has shown the effectiveness of the direct approach: in the considered example with long straight sections, the approximation MSE by the adaptive CPLF is 59.9 times less compared to the fixed step CPLF. The modified version of ADV routing protocol shows a lower overhead value (the gain for small pause time values reaches 23 %). Thus, the effectiveness of the proposed approximation-based routing in FANET is shown.

Acknowledgment

The work was supported by the RFBR grant 19-29-06030-mk "Research and development of a wireless ad-hoc network technology between UAVs and smart city control centers based on the adaptation of transmission mode parameters at various levels of network interaction". The theory was prepared within the framework of the state task of the Russian Federation FZGW-

2020-0029 "Development of theoretical foundations for building information and analytical support for telecommunications systems for geoecological monitoring of natural resources in agriculture".

References

- [1] D.I. Surzhik, G.S. Vasilyev, O.R. Kuzichkin, "Development of UAV trajectory approximation techniques for adaptive routing in FANET networks," 7th International Conference on Control, Decision and Information Technologies (CoDIT'2020), June 29-July 2, 2020, 1226-1230, doi: 10.1109/CoDIT49905.2020.9263944.
- [2] K. Palan, P. Sharma, "FANET Communication Protocols: A Survey," International Journal of Computer Science & Communications (IJCS), 7(1), 219-223, 2016, doi: 10.090592/IJCS.2016.034.
- [3] A. Kiryanov, "Analysis of mechanisms for constructing logical topology in MANET networks," Information Processes, 15(2), 183-197, 2015.
- [4] D.E. Prozorov, "Protocol of hierarchical routing of a ad-hoc mobile network," Radio Engineering and telecommunications systems, 15(3), 74-80, 2014.
- [5] T. Brown, S. Doshi, S. Jadhav, D. Henkel, R. Thekkekkunnel, "A full scale wireless ad hoc network test bed," Proc. of International Symposium on Advanced Radio Technologies, 50-60, 2005.
- [6] K. Brad, K. Hsiang, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM), 2020, doi:10.1145/345910.345953.
- [7] A.I. Alshabtat, L. Dong, J. Li, F. Yang, "Low latency routing algorithm for unmanned aerial vehicles ad-hoc networks," International Journal of Electrical and Computer Engineering, 6(1), 48-54, 2010.
- [8] R.V. Kirichek, "Development and research of a complex of models and methods for flying sensor networks", Doct. thesis of technical sciences, St. Petersburg, 2017.
- [9] A.I. Alshabtat, Cross-Layer Design For Mobile Ad-Hoc Unmanned Aerial Vehicle Communication Networks, PhD Thesis, Western Michigan University, 2011.
- [10] N. Ahmed, S. Kanhere, S. Jha, "Link characterization for aerial wireless sensor networks," GLOBECOM Wi-UAV Workshop, 1274-1279, 2011, doi: 10.1109/JSAC.2013.130825.
- [11] T. Samil, B. Ilker, "LODMAC: Location Oriented Directional MAC protocol for FANETs," Computer Networks, 83(4), 76-84, 2015, doi: 10.1016/j.comnet.2015.03.001.
- [12] M.S. Bobrov, A.M. Averyanov, G.G. Piskunov, V.V. Chekushkin, "Implementation of air object movement routes in simulator-modeling systems," Questions of radio electronics, 4, 157-166, 2009.
- [13] V.N. Burkov, "Adaptive predictive flight control systems", Moscow, Nauka. Phys.-mat. lit., 1987.
- [14] A. Trewar, "Advanced Control of Aircraft, Spacecraft and Rockets", John Wiley & Sons, 2011.
- [15] I.A. Kurilov, D.V. Pavelev, D.N. Romanov, "Review of methods for constructing the trajectory of an air object in a simulator-modeling complex," Radio Engineering and Telecommunications Systems, 1, 42-45, 2011.
- [16] G.S. Vasilyev, O.R. Kuzichkin, D.I. Surzhik, I.A. Kurilov, "Algorithms for analysis of stability and dynamic characteristics of signal generators at the physical level in FANET networks," MATEC Web of Conferences 309, 03019 (2020), doi: https://doi.org/10.1051/mateconf/202030903019.
- [17] I.A. Kurilov, G.S. Vasiliev, S.M. Kharchuk, "Analysis of dynamic characteristics of signal converters based on continuous piecewise linear functions," Scientific and technical Bulletin of the Volga region, 1, 100-104, 2010.
- [18] I.A. Kurilov, G.S. Vasilyev, S.M. Kharchuk, D.I. Surzhik, "Research of static characteristics of converters of signals with a nonlinear control device," 2011 International Siberian Conference on Control and Communications (SIBCON), 93 - 96, 2011.
- [19] Yu.A. Maksimov, E.A. Fillipovskaya, "Algorithms for solving problems of nonlinear programming", Moscow: MEPhI, 1982.
- [20] Y.-H. Dai, Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," SIAM J. Optim, 1, 177-182, 1999.
- [21] J. Yoon, M. Liu, B. Noble, "Random Waypoint Considered Harmful," 0-7803-7753-2/03, IEEE INFOCOM, 2003, doi: 10.1109/INFCOM.2003.1208967.
- [22] M. Sheetal, "Performance Comparison of Ad-hoc Routing Protocols," International Journal of Advance research, Ideas and Innovations in Technology, 2(5), 1-8, 2016, doi: 10.3923/ij.2005.278.283.