JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 7, ISSUE 08, 2020

OPTIMIZING CLASSIFICATION THRESHOLDS FOR THE STATE OF TRANSIONOSPHERIC COMMUNICATION CHANNELS DESCRIBED BY THE NAKAGAMI DISTRIBUTION TO ENSURE UAS POSITIONING ACCURACY

Gennadiy I. Linets[1], Sergey V. Melnikov[2], Alexander M. Isaev[3]

[1,2,3]North Caucasus Federal University, Stavropol, Russia

[1]Corresponding author: kbytw@mail.ru

Abstract: Modern satellite navigation systems have become rather advanced. US-developed GPS and Russia-developed GLONASS are the most widely used systems today. However, artificial ionospheric formations negatively affect satellite navigation systems and cause errors in the accuracy of unmanned aircraft positioning. Disturbances of communication channels are described by the laws of random variables distribution. Being based on predictive models, the existing methods of counteracting these disturbances cannot adequately meet the emerging practical needs. A promising method for solving this problem is automated monitoring of the state of satellite communication channels. The number of errors related to omission of abnormal events should be minimized.For this, it is necessary to determine the classification threshold. The solution to this problem is a system of equations, which is reduced to a convenient form. This article presents the statement and the solution of the problem of optimizing the classificationthresholds for the state of transionospheric communication channels during signal fading, described by the Nakagami distribution.

Keywords: transionospheric communication channels, Nakagami law, satellite navigation systems, ionospheric formations, optimization, Quasi-Newton Interior point methods

Introduction

Nonlinear fluctuations in amplitudes of radio signals, passing through ionosphere irregularities, are of a random nature [1,2]. When analyzing radio signal fading, one has to examine the amplitude of quadrature components that arises from interference of signals having the same purity, but different phases, and its distribution function. In order to maintain the required noise immunity and generate control input, it is necessary to assess the state of transionospheric communication channels (TCC) in a near real-time mode, thus ensuring the required accuracy ofunmanned aircraft system (UAS) positioning. Hence, to ensure the UAS positioning accuracy, it is necessary to tackle the problem of identifying the TCCstate.

The most obvious method is an automatic monitoring and control system that can quickly assess the state of the ionosphere and determine the necessary control actions on the transmitted GPS/GLONASS signals directly in the zones of origin of artificial ionospheric formations. When implementing such a system, therearises a task of classifying the TCC states, which can lead to type I and type II errors. Incorrect classification of the TCC states entails:

- loss of control actions,
- deterioration of their noise immunity,
- formation of erroneous control actions,
- misallocation of their potential resources.

The first three cases lead to a decrease in UAS positioning accuracy, and the fourth one, in the energy potential, depending on the coding method used. To maintain the signal-to-noise ratio when using less powerful codes, it is necessary to reduce the capacity of satellite communication channels and the frequency bandwidth of the transmitted signal [3]. This imposes significant limitations on the performance of satellite communication channels.

Let us analyze the factors affecting the UAS positioning accuracy. The sources of positioning errors are [4–6,17,19]:

- ephemeris data: the UAS positioning error occurs due to the fact that the satellite location during signal transmission is known with an accuracy of 1 to 5 m,

ISSN- 2394-5125 VOL 7, ISSUE 08, 2020

- satellite clock: each satellite has an atomic clock, and its time error of mere 10 ns gives a distance error of up to 3 m,
- ionospheric effect: signal fades as it passes through ionospheric formations. The ionization level may vary depending on the time and place of observation. As a rule, ionization level increases during daytime, at the equator and in the vicinity of Aurora Borealis. The error introduced by the ionosphere can be from 2 to 50 m.
- troposphericeffect: weather conditions, such as temperature, pressure and humidity, exert the greatest impact on the transmitted navigation signal. The errors introduced by the troposphere are about 1 m.
- signal reflection: these errors are caused by re-reflection of signals from large objects, for example, buildings, trees, mountains, etc. These errors are about 1 m.
- receiver measurements: positioning errors caused by the receiver do not exceed 0.5 m.

Thus, the ionosphere has the most significant impact on the UAS positioning accuracy. It is known [2] that fading during transionospheric data transmission is described by suchrandom variable distribution laws as the normal distribution, the Rice distribution, the Rayleigh distribution, and the Nakagamidistribution [18]. When analyzing radio signal fading, one has to examine the amplitude of quadrature components that arises from interference of signals having the same purity, but different phases, and its distribution function. Fading in satellite radio signals is usually categorized into slow and fast.

Generally, for the fading amplitudes of individual beams and the whole signal, the *Nakagamim-distribution* is accepted and justified [11,12]:

$$w_1(\gamma) = \frac{2 \cdot m^m \cdot \gamma^{2m-1}}{\Gamma(m) \cdot \alpha^{2m}} \cdot \exp\left(-\frac{m}{\alpha^2} \cdot \gamma^2\right), \gamma \geq 0$$

where $\alpha$and$m \geq 1/2$ are the distribution parameters and $\Gamma(m)$ is the gamma function.

The parameter $\alpha = \sqrt{\gamma^2}$ expresses the mean square value of the amplitude γ, and $m = a^4/\left(\overline{\gamma^2 - \alpha^2}\right)^2$ is the ratio of the squared average power of the fading signal to the dispersion of its momentary power, i.e. it characterizes the fading depth. The form $w_1(\gamma)$was theoretically obtained for the distribution of the non-negative function of many random arguments and experimentally confirmed when testing various radio paths.

There are ways to reduce the ionospheric effect on the transmitted signal. However, such methods, for the most part, are based on long-term forecast models that cannot fully meet the existing practical requirements for maintaining the positioning accuracy. What is also essential is to monitorthe state of the ionosphere in a near real-time mode. Some methods [7–9] use stationary objects. Some GPS/GLONASS receivers apply dual-frequency reception with subsequent processing [20,21]. However, in case of lightweight UASs, solving the GPS/GLONASS signal correction problem by adding moreonboard equipment is unfeasible because lightweight UASs must have small navigation equipment.

A more effective approach can be the calculation of the GPS/GLONASS signal correction components directly in the UAS location since ionospheric formations can have different characteristics [10,13,14]. Flight control centers (FCC) can be deployed in the UAS takeoff/landing area. Since FCCs are immobile, it is possible to regularly update data on the state of the ionosphere. Surely, changes in the state of the ionosphere are contingent [15], and manual adjustment of satellite communication channel parameters is unreliable. [16]. Efficient signal processing in fading channels can be achieved by using adaptive algorithms in the automatic monitoring and control system. Such adaptive approach allows changing the parameters of communication channels as often as required, thus maintaining the positioning accuracy in the ionosphere. Such automatic monitoring and control system should be able to perform the following functions:

- identify the state of the controlled object,
- generate the control input based on the operation objectives and the current state of the environment,
- implementthe control action.

To identify the state of complex objects, it is advisable to use pattern recognition techniques, including machine learning methods. However, at the identification stage, type I and type II errors are inevitable. The identification of each state of the satellite communication system could produce a distribution or a list of all recognition classes in descending order by the degree of similarity of the identified object to them. The identification and the selection of the decision rule depends on *a priori* and *a posteriori*knowledge of the states of the data transmission system. It is necessary to determine optimal values of the classification thresholds, since such errors directly affect the positioning accuracy.

This article proposes a solution to the problem of optimizing the classification thresholds for type I and type II errors in an automatic system that monitors TCCswhose state is described by the Nakagamim-distribution.

ISSN- 2394-5125 VOL 7, ISSUE 08, 2020

Materials and Methods

The identification system searches and recognizes by the given parameters. The detecting device makes a selection by the given attribute x; the recognition device determines if the selected object belongs to the corresponding class based on the presented attribute y. At that, the distribution of attributes x and $y$follows the Rice distribution. In the process of operation, the identification system inevitably makes type I and type II errors, and it is necessary to minimize the probability of such errors.

To do this, let us formally reduce the problem of object recognition to testing a seriesof hypotheses $B_1, B_2 \dots, B_i, \dots B_k$, where B_i is the hypothesis that the object belongs to the classA_i. Assume that the *a priori* probability distributions for these hypotheses are given, i.e. it is known with what probability $P(B_i)$an object can belong to the class A_i. Moreover, $\sum_{i=1}^{k} P(B_i) = 1$ since the object must belong to a certain class. Under this condition, the distribution can be written as

$$p_i(x) = p\left(\frac{x_i}{B_i}\right) (1)$$

The identification system uses two hypotheses $B_1 = N$and$B_2 = \overline{N}$ with the corresponding *a priori* probabilities of the normal $p_1 = p(B_1) = p(N)$ and abnormal$p_2 = p(B_2) = p(\overline{N})$situations in the system. Moreover, $p_1 + p_2 = 1$.

Let us use the Neumann-Pearson criterion as a decision rule ensuring the highest accuracy of the identification system. Fix the probability of a false alarm $P_{f.a.}$ at the constant level C and require minimum skipping error P_{om}^{min} in the system disturbances. Then

$$P_{om}^{min} = \min\left(p_2 \beta(x_o)\right) (2)$$

under the constraint

$$P_{f.a.} = p_1 \alpha(x_0) = C = const, (3)$$

where$\alpha(x_0)$are the type I errorsand $\beta(x_0)$are the type II errors.

Define the type I errors ('false alarm' errors) and the type II errors ('abnormal situation' errors):

$$\alpha(x_0) = \int_{x_0}^{\infty} f\left(\frac{x}{N}\right) dx, (4)$$

$$\beta(x_0) = \int_{-\infty}^{x_0} f(x/\overline{N}) dx. (5)$$

Write the function of attribute x distribution density for the normally operating$f(x/N) = f_1(x)$ and the faulty $f(x/\overline{N}) = f_2(x)$ system. Then type I and type II errors of the detector (first stage) will be respectively equal:

$$\alpha_{det} = \int_{x_0}^{\infty} f_1(x) dx\,;\ \beta_{обн} = \int_{-\infty}^{x_0} f_2\,(x) dx\ (6)$$

Type I and type II errors of the identifier are determined (second stage) in the same way:

$$\alpha_{id} = \int_{y_0}^{\infty} f_1\,(y) dy;\ \beta_{pac} = \int_{-\infty}^{y_0} f_2\,(y) dy\ (7)$$

It is necessary to find the optimal classification thresholds $x_0$andy_0.

The problem of optimizing the classification thresholds $x_0$andy_0 can be reduced to solving a system of equations:

$$\begin{cases} \int_{x_0}^{\infty} f_1(x) dx \cdot \int_{y_0}^{\infty} f_1(y) dy = \dfrac{C}{p_1} \\ \dfrac{dx_0}{dy_0} \cdot f_2(x_0) \cdot \int_{y_0}^{\infty} f_2\,(y) dy + f_2(y_0) \cdot \int_{x_0}^{\infty} f_2\,(x) dx = 0 \end{cases} \quad (8)$$

under the constraint$p_1 = const$at$p_1 + p_2 = 1$.

Now, let us determine the optimal classification thresholds x_0 and y_0 from the solution of Equations 8. In the event of fading, described by the Nakagamim-distribution, the final system of equations will be as follows:

$$\begin{cases} H(x_0) \cdot H(y_0) = \dfrac{C}{p_1} \\ f(y_0) \cdot f(x_0 - a) \cdot H(x_0) \cdot H(y_0 - b) \\ -f(x_0) \cdot f(y_0 - b) \cdot H(y_0) \cdot H(x_0 - a) = 0, \end{cases} \quad (9)$$

where$f(x) = \frac{2 \cdot \mu^{\mu}}{\Gamma(\mu) \cdot \omega^{\mu}} \cdot x^{2 \cdot \mu - 1} \cdot e^{\frac{-\mu}{\omega} x^2}$; $H(z) = \int_z^{\infty} f(x)\, dx$.

under the constraints$\omega > 0, \mu \geq \frac{1}{2}, x \geq 0$.

There are several methods of solving non-linear equations (or systems of non-linear equations), such as graphical, analytical and numerical.

The g*raphical methods* are the least accurate, but they can determine the most approximate values in complex equations, and this may a starting point for finding more accurate solutions to the equations.

The *analytical methods* (or *direct methods*) can determine the exact values of the solutionsto the equations. With these methods, the roots can be written in the form of ratios (formulas). However, in practice the vast majority of nonlinear equations cannot be solved by direct methods, and onehas to turn to the numerical methods that make it possible to obtain an approximate root value with any given accuracy ξ.

The *numerical methods*of solving nonlinear equations is an iterative calculation process, which consists in sequentially refining the initial approximation of the root valuesin an equation (a system of equations).In the numerical approach, nonlinear equations are solved in two steps:

1) localization of the roots;
2) refinement of the roots.

The *root localization* is understood as the process of finding the approximate root value or finding the ranges that contain solution. The *root refinement* refers to the process of calculating approximate root values with a given accuracy by any numerical method for solving nonlinear equations.

The disadvantage of nearly all iterative methods for finding the roots is that, when used once, they can only find one root of the function, and it is unknown which one.The Quasi-Newtonian methods are also based on an iterative formula. All the Quasi-Newtonian methods are first-order methods [22,26,27].

Let us consider Broyden's method [24]. In order for the approximation of the inverse Hessian matrix to produce a symmetric matrix, the correction ΔG_kin the formula must also be a symmetric matrix. Take an arbitrary nonzero vector $u = (u_1 u_2 \dots u_n)^T$and construct a matrix:

$$\mathbf{u}\mathbf{u}^T = \begin{pmatrix} u_1 \\ u_2 \\ \dots \\ u_n \end{pmatrix} \begin{pmatrix} u_1 & u_2 & \dots & u_n \end{pmatrix} = \begin{pmatrix} u_1u_1 & u_1u_2 & \dots & u_1u_n \\ u_2u_1 & u_2u_2 & \dots & u_2u_n \\ \dots & \dots & \dots & \dots \\ u_nu_1 & u_nu_2 & \dots & u_nu_n \end{pmatrix}. \quad (10)$$

This is a symmetric matrix with proportional rows. Therefore, it has a unity rank $rank(uu^T) = 1$. Let us form a rank correction and express it as$\Delta G_k = \alpha \cdot uu^{\mathrm{T}}$, where α is some real coefficient. Based on the iterative formula, we obtain:

$$G_{k+1} = G_k + \alpha \cdot uu^{\mathrm{T}} (11)$$

The system of linear algebraic equations takes the following form:

$$(G_k + \alpha \cdot uu^{\mathrm{T}})p_k = s_k.$$

Open the brackets:

$$G_k \cdot p_k + \alpha \cdot uu^{\mathrm{T}} \cdot p_k = s_k,$$

or

$$(\alpha \cdot u^{\mathrm{T}} \cdot p_k) \cdot u = s_k - G_k \cdot p_k,$$

where the brackets contain the scalar value. This vector equality is satisfied if $u$is set to $s_k - G_k \cdot p_k$ and α is set to $1/(u^{\mathrm{T}} \cdot p_k)$under the condition $u^{\mathrm{T}} \cdot p_k \neq 0$. Then, the equality takestheform:

$$G_{k+1} = G_k + \frac{(s_k - G_k \cdot p_k) \cdot (s_k - G_k \cdot p_k)^{\mathrm{T}}}{(s_k - G_k \cdot p_k)^{\mathrm{T}} \cdot p_k}. (12)$$

This formula for correcting the approximation of the inverse Hessian matrix was introduced by English mathematician C. G. Broydenin 1967 and is called Broyden'sformula, and the Quasi-Newton method using it is referred to asBroyden'smethod. Thus, the Broyden's method goes back to Equations 10 and 12, and the principles of the Newton method. The first iteration starts from the given starting point x_0, and the one-dimensional search is performed in the anti-gradient direction:

$$G_0 = E, d_0 = -g_0, \lambda = arg\min_{\lambda} f(x_0 + \lambda \cdot d_0), x_1 = x_0 + \lambda_0 \cdot d_0 \#$$

The subsequent iterations for $k = 1, 2, 3, K$ are performed according to the formulas:

$$p_{k-1} = g_k - g_{k-1}, \quad s_{k-1} = x_k - x_{k-1}, \#$$

$$u_{k-1} = s_{k-1} - G_{k-1} \cdot p_{k-1}, \quad G_k = G_{k-1} + \frac{u_{k-1} \cdot u_{k-1}^{\mathrm{T}}}{u_{k-1}^{\mathrm{T}} \cdot p_{k-1}} \#$$

$$d_k = -G_k \cdot g_k, \quad \lambda_k = arg\min_{\lambda} f(x_k + \lambda \cdot d_k), \quad x_{k+1} = x_k + \lambda_k \cdot d_k$$

The iterations continue until the following condition is met:

$$\|x_{k+1} - x_k\| > \varepsilon,$$

where ε is the permissible error.

ISSN- 2394-5125 VOL 7, ISSUE 08, 2020

Now, let us consider Powell's method [25]. Introduced by English mathematician M. J. D. Powell in 1964, it was one of the first conjugate directions methods to minimize the function of several variables $f(x), x \in R_n$. This method is a development of the method of cyclic coordinate descent in which a sequential one-dimensional search is performed in the directions of the unit vectors of the coordinate axes $e_1 = (1, 0, 0 \ldots, 0)^T, e_2 = (0, 1, 0 \ldots, 0)^T, \ldots, e_n = (0, 0, 0 \ldots, 1)^T$. Unlike the coordinate descent method, Powell's method builds a system of n conjugate directions.

Each iteration of Powell's method begins with n linearly independent directions $d_1, d_2, \ldots, d_n$. Initially, these directions coincide with the unit vectors of the coordinate axes: $d_i = e_i, i = \overline{1,n}$.

The auxiliary one-dimensional search is performed in the last direction d_n from the starting point $u_0 \in R_n$:

$$x_0 = u_0 + \lambda_0 \cdot d_n, \ \ \lambda_0 = arg \min_{\lambda} f(u_0 + \lambda \cdot d_n) . \#$$

Next, the iteration of the coordinate search method is performed from the point x_0 by conducting n consecutive one-dimensional searches in the directions of the unit vector of the coordinate axes:

$$x_k = x_{k-1} + \lambda_k \cdot d_k, \ \ \lambda_k = arg \min_{\lambda} f(x_{k-1} + \lambda \cdot d_k), \ \ k = \overline{1,n} . \#$$

The last point x_n is obtained by minimizing the function from the point x_{n-1} in the direction d_n. Thus, the points x_0 and x_n are found by minimizing the function in the same direction d_n. In the case of a quadratic objective function, by the parallel subspace property, the direction s $s = x_n - x_0$ is conjugate to the direction d_n. The first direction d_1 is excluded from the system of directions $d_1, d_2, \ldots, d_n$, and new direction s is added. At that, it is assumed that $d_i = d_{i+1}$ for $i = \overline{1, n-1}$ and $d_n = s$. For a quadratic function, after the first iteration is complete, the last two search directions d_{n-1} and d_n are conjugate.

The starting point for the next iteration is set to $u_0 = x_n$. The iterations continue until the norm of the vector s remains greater than the permissible error ε.

Therefore, for a quadratic function after k iterations of Powell's method $k+1$, the last search directions $d_{n-k}, \ldots, d_n$ selected for the $(k+1)$-th iteration will be mutually conjugate. After $n-1$ iterations, all directions will be mutually conjugate; hence Powell's method ensures minimization of the quadratic function in no more than n iterations. Thus, to minimize a quadratic function with two variables $n = 2$, two iterations are enough. In this case, the minimum point will be found after performing four one-dimensional searches.

Recently, 'matrixless' iterative methods for solving systems of nonlinear equations have gained popularity. By this we mean calculations that do not require matrix storage and allow using only the result of multiplying the matrix by the vector, that is a vector. An example of suchtechnique is the GRIMES algorithm. In the 1980s, the Newton–Krylov method [23] was developed to solve systems of nonlinear algebraic equations based on the GRIMES algorithms. This method does not require calculation and storage of the system's Jacobian matrix;it only approximates the result of multiplying this matrix by the vector.

The programming language used in this study was Python. It has established itself as a powerful facility for scientific copmuting due to its simple syntax and the availability of developed tools for solving a wide range of practical problems. Python's main advantages are:

- simple and concise syntax,
- cross-platform framework, no need recompile the program due to the scripting nature of the language,
- Availability of application libraries that expand the problem-solving capabilities of the language.

Also, the SciPy library waschosen as the basis for the calculations. Additionally, the SymPylibrary was used to automate the calculations by analytical methods and to obtain formulas of the derivatives. The Matplotlib library was used to visualize obtained the results.

The developed software models are operator-oriented and work in an intective text mode. The method for finding the classification thresholds using the developed models is similar for both cases and consists of the following steps:

1) the operator runs the main file *main.py* in the model directory in text mode;
2) the program introduces constants a and b, which describe the information content of the system state attributes. To describe the laws, μ and ω are also introduced (they are necessary for calculating the system of equations with the Nakagamidistribution);
3) after receiving the initial data, the program displays graphs of the probability distributions of the system states for visual assessment of the type I and type II error regions based on information indicators and the distribution laws;
4) after the calculation is completed, a 3-dimensional graph of the system of equations is displayed for visual assessment and selection of the most suitable starting point for solving the system of equations.This step is most important because poor choice of the starting point can lead the search algorithm away from the solution to one of the neighboring ones, thus hindering the search for the proper solution.

ISSN- 2394-5125 VOL 7, ISSUE 08, 2020

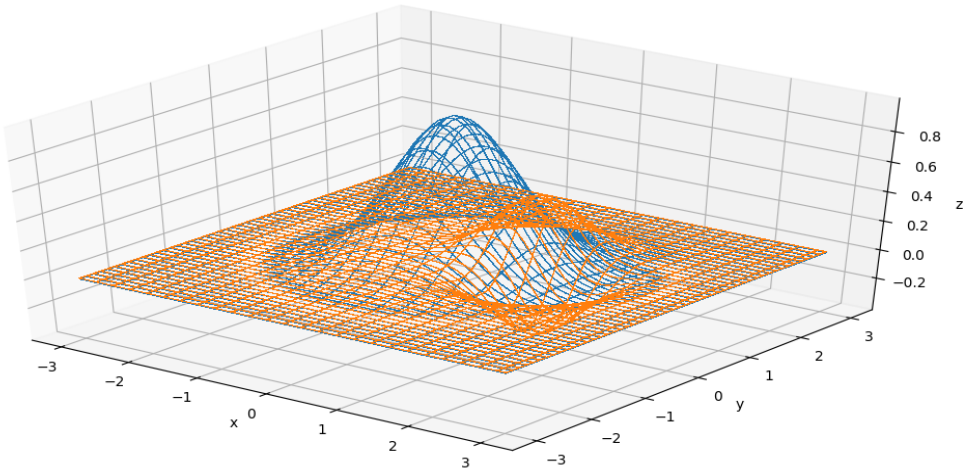

Fig. 1 General view of the system of equations for the Nakagami distribution

5) once the starting point of the solution is determined, one needs to enter its location in the video coordinates x_0, y_0;
6) the program performs the calculations by three different methods and displays the results. The tuple x corresponds to the resulting values of x, y, respectively.

The result is the values of x_0 and y_0.

Results and Discussion

The examples presented below are the solutions to one problem by three methods ($a = 1, b = 1.5, mu = 1, omega = 2$) at different starting points x_0 and y_0.

Table 1. Analysis of the obtained solutions

Startingpoints x_0 and y_0	Obtained values x_0 and y_0		
	Powell	Krylov	Broyden
(0.0, 0.0)	(1.2275035474901574, 2.2773520849797)	(1.2283199559138636, 2.2766282759620733)	(1.227966741105207, 2.27703184471988)
(0.5, 0.0)	(1.2275035474918843, 2.277352084978635)	(1.227535176545848, 2.277334317816012)	(1.2276156121234743, 2.2773018941553502)
(1.0, 0.0)	(3.023341439739129, -7.41525170976003)	(-1.6398462907020985, 4.006657739884625)	(3.023519968589054, -3.249660951806811)
(1.5, 0.0)	(3.0233414397391476, -8.193411541188574)	(3.0233696246574646, -3.412246904451698)	(3.023275555456474, -3.0531509453517547)
(2.0, 0.0)	(3.0233414397391476, -8.212554288554939)	(3.0234016172861637, -3.240366924784658)	(-8662184.216819776, 19439849.258796554)
(0.0, 0.5)	(1.2275035474813618, 2.27735208498532)	(1.2282510719035555, 2.2769861079235603)	(1.229891271358504, 2.2758995124157857)
(0.5, 0.5)	(1.2275035474920244, 2.2773520849784448)	(1.226953922604321, 2.277725031840208)	(1.2301349320620636, 2.2758433656493096)
(1.0, 0.5)	(1.2275035473754867, 2.277352085050647)	(1.2275022293135798, 2.2773439955773616)	(1.2301102537147202, 2.275943073243638)
(1.5, 0.5)	(3.0233414397391467, -8.236802997796786)	(3.023373431331797, -3.383698630097059)	(1.2259703068698353, 2.2781485111710826)
(2.0, 0.5)	(3.0233414397391476, -8.121102613050837)	(3.0233398916253127, -5.2193193072717445)	(-12714848.077741433, 13847999.063017955)
(0.0, 1.0)	(1.2275035474888298, 2.2773520849803486)	(1.2275005642301575, 2.2773540723510384)	(1.2250554438379064, 2.278966763525886)
(0.5, 1.0)	(1.2275035474920524, 2.277352084978426)	(1.2275013398970718, 2.2773532579602627)	(1.2299480994326522, 2.276027323906365)

(1.0, 1.0)	(1.2275035476653038, 2.277352084877108)	(1.2276778551177254, 2.2771252279697416)	(1.2295255592974077, 2.2763425002618334)
(1.5, 1.0)	(1.2275035474920948, 2.2773520849783893)	(1.22749899529813, 2.277355775174491)	(1.2265945986690387, 2.277950442981315)
(2.0, 1.0)	(1.227503547492061, 2.277352084978427)	(1.227468196574656, 2.2773670959383607)	(1.2276554020295103, 2.277118208341149)
(0.0, 1.5)	(1.227503547492058, 2.2773520849784235)	(1.2276774811755626, 2.2772538276724696)	(1.225236091590747, 2.2787461724977196)
(0.5, 1.5)	(1.2275035474919396, 2.2773520849784914)	(1.2269288027140473, 2.277527059407258)	(1.2265695442603393, 2.2779158120342253)
(1.0, 1.5)	(1.2275035474978668, 2.2773520849757207)	(1.227400661041429, 2.2774823470118664)	(1.2272372212261056, 2.2773521251100344)
(1.5, 1.5)	(1.2275035473877953, 2.2773520850246416)	(1.2262767993189607, 2.2779693432904833)	(1.2277580960231325, 2.2772324897040788)
(2.0, 1.5)	(1.2275035474912062, 2.2773520849789155)	(1.2276994631652192, 2.277152403216245)	(1.2269398358637067, 2.277493498256505)
(0.0, 2.0)	(1.2275035478430294, 2.277352084829311)	(1.2273624496680675, 2.277414846533952)	(1.2279720295694143, 2.277131789388571)
(0.5, 2.0)	(1.2275035474922138, 2.2773520849783297)	(1.2275027060227715, 2.2773519540704195)	(1.228135220529438, 2.2769960319074234)
(1.0, 2.0)	(1.227503547458006, 2.2773520849952296)	(1.2275037841119627, 2.277351484923138)	(1.2275461955856914, 2.277328176798017)
(1.5, 2.0)	(1.2275035474830172, 2.2773520849854436)	(1.2275107892284411, 2.277345532338507)	(1.2266960774441968, 2.2778267587967416)
(2.0, 2.0)	(1.2275035474435476, 2.277352085004722)	(1.229112264658322, 2.276419413683161)	(-3.613833979471394, 3.0232450998326446)

Table 1 demonstrates that in some cases there is a discrepancy in the solution, and the obtained roots fail to fit into the adequate range of values.

Table 2. Mean deviation of the obtained values (MSD – mean-squaredeviation; MV – meanvalue; MD – mean deviation)

#	Powell x	Powell y	Broyden x	Broyden y	Krylov x	Krylov y
1	1.22750354749015	2.277352085	1.227966741	2.277031845	1.228319956	2.276628276
2	1.22750354749188	2.277352085	1.227615612	2.277301894	1.227535177	2.277334318
3	1.22750354748136	2.277352085	1.229891271	2.275899512	1.228251072	2.276986108
4	1.22750354749202	2.277352085	1.230134932	2.275843366	1.226953923	2.277725032
5	1.22750354737548	2.277352085	1.230110254	2.275943073	1.227502229	2.277343996
6	1.22750354748882	2.277352085	1.225970307	2.278148511	1.227500564	2.277354072
7	1.22750354749205	2.277352085	1.225055444	2.278966764	1.22750134	2.277353258
8	1.22750354766530	2.277352085	1.229948099	2.276027324	1.227677855	2.277125228
9	1.22750354749209	2.277352085	1.229525559	2.2763425	1.227498995	2.277355775
10	1.22750354749206	2.277352085	1.226594599	2.277950443	1.227468197	2.277367096
11	1.22750354749205	2.277352085	1.227655402	2.277118208	1.227677481	2.277253828
12	1.22750354749193	2.277352085	1.225236092	2.278746172	1.226928803	2.277527059
13	1.227503547	2.277352085	1.226569544	2.277915812	1.227400661	2.277482347

	49786					
14	1.22750354738779	2.277352085	1.227237221	2.277352125	1.226276799	2.277969343
15	1.22750354749120	2.277352085	1.227758096	2.27723249	1.227699463	2.277152403
16	1.22750354784302	2.277352085	1.226939836	2.277493498	1.22736245	2.277414847
17	1.22750354749221	2.277352085	1.22797203	2.277131789	1.227502706	2.277351954
18	1.22750354745800	2.277352085	1.228135221	2.276996032	1.227503784	2.277351485
19	1.22750354748301	2.277352085	1.227546196	2.277328177	1.227510789	2.277345532
20	1.22750354744354	2.277352085	1.226696077	2.277826759	1.229112265	2.276419414
MSD	9.711E-11	4.69728E-11	0.001549833	0.000897447	0.000562909	0.000334739
MV	1.22750354750209	2.2773520849 7473	1.2277279266 2965	2.2772298147 4324	1.2274795009 8760	2.2773305426 0742
MD	5.041E-11	2.430E-11	0.001189086	0.000673468	0.000315883	0.000200916

However, some of these cases can still be solved by one of the methods. Also, it is worth noting the drift of the values in Broyden's method; it is the most noticeable among all the methods in calculating x_0 (Table 2). The Newton–Krylov method and Powell's method show great stability in the roots found, but all the methods provide satisfactory accuracy.

The obtained data require a more detailed analysis. Overall, it can be concluded that all the methods show satisfactory convergence, butBroyden's method is the least preferred due to the instability in the found values.

Let us analyze the execution time of the algorithms for the same values. The average runtime was 1.204948453 sec of the Newton–Krylov method, 0.207069251 sec for Powell's method, and 1.233867905 sec for Broyden's method (Table 3).

However, these values for the Newton–Krylov method were due to the fact that in one of the samples the execution time was 27 seconds. When restrictions were introduced, the average runtime for the Newton–Krylov method was 0.121402934 sec. The Newton-Krylov method was the fastest-performing method, although in some cases it can be recommended to use Broyden's method instead.

Table 3. Algorithm execution time

Time consumed		
Powell	Krylov	Broyden
0.3031167880000001	0.1423163220000001	0.7637078720000001
0.15913422399999977	0.14534456499999981	1.2242261829999999
0.21961959699999944	27.210040906	2.0598159219999985
0.5714401410000036	0.1751353769999966	1.5623113560000021
0.5124958299999989	0.18170836499999865	7.805860731000003
0.1395910549999968	0.12855332000000175	1.3408058059999988
0.1626585380000023	0.15319850300000581	0.9465874950000028
0.20444523100000112	0.1680408769999957	0.872501548999999
0.5489383179999976	0.1694431589999965	0.9060260680000027
0.5328905970000051	0.10056049900000374	8.026601857000003
0.1349581050000026	0.13610255400000426	1.7780190029999972
0.14670130100000023	0.1492340130000045	0.5278627020000002
0.1554254549999996	0.13044990499999898	0.4504877399999998
0.1177973659999978	0.12191647299999886	0.7008386910000013
0.10562297199999904	0.08045322899999974	0.18074797300000256
0.1481643399999939	0.1311049899999972	0.24113859899999568
0.1256710890000079	0.0977284349999934	0.19238207000000784
0.13313891500000352	0.09799041099999783	0.1766878340000062

0.12940089100000307	0.0773086749999976	0.10982611699999723
0.10664126199999657	0.08339259500000651	0.10361611999999809
0.11527472299999886	0.09350444199999686	0.38221397600000273
0.11109182399999895	0.10357336300000952	0.16111022699999467
0.08228639400000759	0.08782046999999693	0.07648376300001303
0.08777811199999519	0.06565349999999626	0.05892773300000442
0.12244821800000238	0.09313637700000754	0.19791023499999483
	Average runtime	
0.20706925	1.20494845	1.2338679

Table 3 demonstrates the Newton–Krylov method is the fastest of the considered ones. However, when the system solution diverges, the execution runtime of this method increases by orders of magnitude. The most stable runtime was shown by Powell's method.

Conclusion

We have found a solution to the problem of determining optimal classification thresholds in the case when theTCCstate is described by the Nakagamim-distribution. The presented solution was obtained using quasi-Newtonian methods and showed sufficient accuracy. The obtained algorithm can be used to optimize the classification of the TCC states during Nakagamifading.

Future work will focus on the optimization of classification thresholds for the states of the ionosphere described by the normal distribution, the Rice distribution, the Rayleigh distribution, and the four-parameter distribution.

Acknowledgments

This study was carried out within the framework of the scientific project *Development of a multi-rotor robotic unmanned aerial vehicle using a strapdown inertial navigation system* that is a part of the 2014–2020 Federal Target Program (Project ID RFMEFI57818X0222) funded by the Ministry of Science and Higher Education of Russia.

References

[1]. Prakanrattana, K., Satirapod, C.: Comparative study of using different ionosphere models in Thailand for single-frequency GNSS users.Surv. Rev. (2019).https://doi.org/10.1080/00396265.2018.1426260

[2]. Deshpande, K.B., Zettergren, M.D.: Satellite-beacon ionospheric-scintillation global model of the upper atmosphere (SIGMA) III: Scintillation simulation using a physics-based plasma model.Geophys. Res.Lett. (2019).https://doi.org/10.1029/2019GL082576

[3]. Zhou, D., Sheng, M., Wang, X., Xu, C., Liu, R., Li, J.: Mission aware contact plan design in resource-limited small satellite networks. IEEE Trans.Commun. 65(6), 2451–2466 (2017)

[4]. Konrad, T., Breuer, M., Engelhardt, T., Abel, D.: State estimation for a multirotor using tight-coupling of GNSS and inertial navigation. IFAC-PapersOnLine 50(1),11683–11688(2017)

[5]. Luzzatto, A., Haridim, M.: Wireless Transceiver Design, 2nd ed.JohnWiley&Sons, Chichester, UK; Hoboken, NJ (2016)

[6]. Bercher, J.-F.: Entropies et Radiotechnique. Traitement du Signal et de l'Image [eess.SP].Université Paris-Est (2009)

[7]. Katkov,K.A.,Pashintsev, V.P., Katkov,E.K.: Information system for monitoring the ionosphere. IzvestiyaSamarskogonauchnogotsentra RAN 18, 2(3), 907–912 (2016) (in Russian)

[8]. Sazhin, V.I., Unuchkov, V.E.: Possibility of correcting ionospheric model, using data from single-frequency receivers of global navigation satellite systems.Proc. SPIE10466 (2017).https://doi.org/10.1117/12.2284635

[9]. Neusypin, K.A., Proletarsky, A.V., Shen, K., Liu, R., Guo, R.: Aircraft self-organization algorithm with redundant trend. Nanjing Li Gong DaxueXuebao/Journal of Nanjing University of Science and Technology 38(5), 602–607 (2014)

[10]. Pashintsev, V.P., KovalS.A, StrekozovV.I., BessmertnyiyM.Yu.: Detection of artificial ionospheric disturbances via satellite navigation systems. Teoriyaitehnikaradiosvyazi1, 112–116 (2013)(in Russian)

[11]. Golubkov, G.V., Manzheliy, M.I., Berlin, A.A., Morozov, A.N., Eppelbaum, L.V.: Problems of satellite navigation and remote sensing of Earth's surface. Herald of the Bauman Moscow State Technical University, Series Natural Sciences 1, 61–73(2018)

[12]. Ippolito, L.J., Jr.: Satellite Communications Systems Engineering: Atmospheric Effects, Satellite Link Design and System Performance. John Wiley & Sons, Chichester, UK (2008)

[13]. Zhao, Q., Wang, Y.T., Gu, S., Zheng, F., Shi, C., Ge, M., Schuh, H.: Refining ionospheric delay modeling for undifferenced and uncombined GNSS data processing. J.Geod. 93(4), 545–560(2019)

[14]. Binns, C.: Aircraft Systems: Instruments, Communications, Navigation, and Control. John Wiley & Sons, Hoboken, NJ (2019)

[15]. Chen, Y.-H., Perkins, A., Lo, S., Akos, D.M., Blanch, J., Walter, T., Enge, P.: Demonstrating ARAIM on UAS using software defined radio and civilian signal GPS L1/L2C and GLONASS G1/G2. Proc. 2016 International Technical Meeting of the Institute of Navigation, 231–238 (2016).https://doi.org/10.33012/2016.13420

[16]. Moraes, A.O., Vani, B.C., Costa, E., Abdu, M.A., de Paula, E.R., Sousasantos, J., Monico, J.F.G., Forte, B., de SiqueiraNegreti, P.M., Shimabukuro, M.H.: GPS availability and positioning issues when the signal paths are aligned with ionospheric plasma bubbles. GPS Solutions 22(4), 95(2018)

[17]. Makarenkova, E.V., Gherm, V.E.: Multifractal phase screen model for scintillation of transionospheric signals. 2nd URSI Atlantic Radio Science Meeting, AT-RASC (2018).https://doi.org/10.23919/URSI-AT-RASC.2018.8471300

[18]. de O. Moraes, A., Vani, B.C., Costa, E., Sousasantos, J., Abdu, M.A., Rodrigues, F., Gladek, Y.C., de Oliveira, C.B.A., Monico, J.F.G.: Ionospheric scintillation fading coefficients for the GPS L1, L2, and L5 frequencies. Radio Sci. 53(9), 1165–1174 (2018).

[19]. Myer, G.T., Morton, YuT.J.: Ionosphere scintillation effects on GPS measurements, a new carrier-smoothing technique, and positioning algorithms to improve accuracy. Proc. 2018 International Technical Meetingof the Institute of Navigation, 420–439(2018).https://doi.org/10.33012/2018.15581

[20]. Veettil, S.V., Aquino, M., De Franceschi, G., Spogli, L., Cesaron, C., Romano, V.: Statistical models to provide meaningful information to GNSS end-users under ionospheric scintillation conditions. Proc. 31st International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2018, 3827–3832(2018).https://doi.org/10.33012/2018.16034

[21]. Ivanov, D.V., Ivanov, V.A., Ryabova, N.V., Ryabova, M.I., Chernov, A.A., Konkin, N.A., Kislitsin, A.A.: The plotting algorithm of coherence band maps of transionospheric radio channels. Proc. SPIE 10466 (2017). https://doi.org/10.1117/12.2285658

[22]. Fang, X., Ni, Q., Zeng, M.:A modified quasi-Newton method for nonlinear equations.J. Comp. Appl. Math. 328, 44–58(2018)

[23]. Yildirim, A., Kenway, G.K.W., Mader, C.A., Martins, J.R.R.A.: A Jacobian-free approximate Newton–Krylov startup strategy for RANS simulations. J. Comp. Phys. 397, 108741 (2019)

[24]. Gondzio, J., Sobral, F.N.C.: Quasi-Newton approaches to interior point methods for quadratic problems.Comput.Optim. Appl. 74(1), 93–120(2019)

[25]. Ghosh, D.: A Davidon-Fletcher-Powell type quasi-Newton method to solve fuzzy optimization problems.Commun. Comp. Inform. Sci. 655, 232–245(2017)

[26]. Brust, J.J., Marcia, R.F., Petra, C.G.: Large-scale quasi-Newton trust-region methods with low-dimensional linear equality constraints. Comput. Optim. Appl. (2019).https://doi.org/10.1007/s10589-019-00127-4

[27]. Brust, J., Burdakov, O.,Erway, J.B., Marcia, R.F.: A dense initialization for limited-memory quasi-Newton methods. Comput. Optim. Appl. (2019). https://doi.org/10.1007/s10589-019-00112-x