ISSN 2782-2427

CONTROL SCIENCES

2/2023

[1]Russian Academy of Sciences.

CONTROL SCIENCES
Scientific Technical Journal

6 issues per year
ISSN 2782-2427
Open access

Published since 2021

Original Russian Edition
Problemy Upravleniya
Published since 2003

FOUNDER AND PUBLISHER
V.A. Trapeznikov
Institute of Control Sciences
of Russian Academy of Sciences

Editor-in-Chief
D.A. Novikov, RAS Academician

Deputy Editor-in-Chief
F.F. Pashchenko

Executive Editor-in-Chief
N.E. Maximova

Editor
L.V. Petrakova

Editorial address
65 Profsoyuznaya st., office 410,
Moscow 117997, Russia

☎/🖷 +7(495) 198-17-20, ext. 1410

✉ pu@ipu.ru

URL: http://controlsciences.org

Published: May 12, 2023

Registration certificate of
Эл № ФС 77-80482
of 17 February 2021
issued by the Federal Service
for Supervision of Communications,
Information Technology, and Mass
Media

CONTROL SCIENCES

2.2023

CONTENTS

DOI: http://doi.org/10.25728/cs.2023.2.1

METHODS FOR SOLVING SOME PROBLEMS OF AIR TRAFFIC PLANNING AND REGULATION. PART II: Application of Deep Reinforcement Learning

E.L. Kulida[1] and V.G. Lebedev[2]

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

[1]✉ elena-kulida@yandex.ru, [2]✉ lebedev-valentin@yandex.ru

Abstract. Following part I of the survey, this paper considers the problems of improving the safety and efficiency of air traffic flows. The main challenge in conflict detection and resolution by traditional optimization methods is computation time: tens and even hundreds of seconds are required. However, this is not so much for response in real situations. Deep reinforcement learning has recently become widespread due to solving high-dimensional decision problems with nonlinearity in an acceptable time. Research works on the use of deep reinforcement learning in air traffic management have appeared in the last few years. Part II focuses on the application of this promising approach to the following problems: detecting and resolving aircraft conflicts, reducing the complexity of air traffic at the national or continental level (a large-scale problem), and increasing the efficiency of airport runways through the improved planning of aircraft landings.

Keywords: air traffic management, strategic planning of 4D trajectories, aircraft conflict detection and resolution, reinforcement learning.

INTRODUCTION

Due to the growing air traffic flows and overload of major airports, there is an increasing demand for automating the work of air traffic controllers through developing decision support systems and automated air traffic management systems. Part I of the survey [1] was devoted to the problem of minimizing the number of potential conflicts between aircraft.

The paper [2] overviewed current trends in the application of artificial intelligence (AI) to air traffic management based on conference proceedings and publications on the subject in high-rank journals. Despite significant progress in research on AI for air traffic management, it has not yet become "fully functional" for end users. The slow pace of using AI in air traffic management is due to the critical role of this area: lives are at stake here, and safety is the top priority. Currently, safety in air traffic management is achieved through human participation in the control loop. According to the authors cited, safety will evolve by designing human-oriented systems, understandable to the end user and adaptable to their psychological state. This requires moving toward a more user-oriented, eXplainable AI, where the AI system and the end user can understand each other and interact with each other.

Optimization-based approaches are often computationally expensive, which limits their application. Impressive results were obtained in several research works on air traffic management based on deep reinforcement learning; for details, see [3].

In [4], a reinforcement learning-based model was first formulated and an AI agent was presented to mitigate conflicts and minimize aircraft delays when reaching checkpoints. In [5, 6], different levels of environment uncertainty and traffic density were considered and their effect on the performance of the reinforcement learning-based model to resolve aircraft conflicts was investigated.

If the solutions offered by automatic conflict resolution do not match the dispatchers' thinking or preferences, they are unlikely to be accepted. The paper

[7] developed an interactive AI agent based on reinforcement learning with conflict resolution maneuvers used by a human dispatcher. This approach can potentially increase the dispatcher's level of confidence in the solutions proposed by the agent. The hybrid algorithm proposed in [8] uses known geometric methods in the deep reinforcement learning stage to resolve low-altitude airspace conflicts.

These approaches are effective under low air traffic densities, but centralized architectures cannot cope with intensive air traffic flows when the number of conflicting aircraft increases. In most complex systems, distributed decision-making is believed to have higher efficiency than centralized control. A critical challenge for distributed decision-making in air traffic management is the development of a system that provides recommendations to the aircraft for ensuring safe separation and elimination of uncertainty in real time. Several multi-agent approaches were proposed to deal with high air traffic densities. As was demonstrated in [9–11], multiple agents in a decentralized system can access the complete information about all aircraft in a sector using a scalable and efficient method to achieve high throughput under uncertainty. The agents were trained by one neural network with centralized learning, and a decentralized decision-making scheme was adopted. Many of the proposed agents based on reinforcement learning must be trained in an environment with a fixed number of conflicting aircraft. The computational complexity of learning grows rapidly with increasing the number of conflicting aircraft. In [12], image-based deep reinforcement learning was suggested for resolving aircraft conflicts. Image-based deep learning largely solves the scalability problem. The algorithm can process an arbitrary number of aircraft since their states are replaced by their images. The paper [13] presented an autonomous air traffic management model with aircraft collision prevention in free airspace. A graphical neural network approach to resolving conflicts in free airspace was introduced. Representing each aircraft as a graph node, this approach can handle an arbitrary number of aircraft.

Expectedly, deep reinforcement learning will play a significant role in building future air traffic management systems, but more research is needed here. In real-world applications, deep reinforcement learning raises two significant problems.

The first problem is the safety of air traffic management systems. Modern models use deep neural networks as function approximators. Deep neural networks were discovered to suffer vulnerability to adversarial examples [14, 15] (carefully designed quasi-negligible perturbations that mislead the deep neural network when added to its input data). Furthermore, adversarial examples also appear in the real world without any intruder or maliciously chosen noise [16]. Consequently, it is necessary to scrutinize the mechanisms of adversarial attacks, ways to detect errors or undesirable behavior of AI algorithms, and methods to overcome them.

The second problem consists in explainability. The decision-making process of deep reinforcement learning technology is opaque. Due to its insufficient transparency, pilots and air traffic controllers cannot understand the internal mode of operation. The "black box" nature of the model may prevent users from accepting the predicted outcomes, especially when the model makes key decisions.

The paper [17] presented safety-aware deep Q networks. In this model, two separate neural networks jointly investigate security and optimization costs. According to the authors, their work is the first to consider the vulnerability of the model to adversarial attacks; moreover, the model is safe and well-explainable.

The bottleneck in air traffic management systems is the capacity of the runways of major airports. Aircraft maneuvering control operations in the airport area, such as arrival control, landing sequence, and time planning, are performed by air traffic controllers. The paper [18] systematically overviewed past and most recent theoretical studies of the aircraft landing problem for airports with one or more runways, including their comparison.

The remainder of this paper discusses in detail the deep reinforcement learning approach with application to the following problems: detecting and resolving aircraft conflicts, reducing the complexity of air traffic at the national or continental level (a large-scale problem), and increasing the efficiency of airport runways through the improved planning of aircraft landings.

1. CONFLICT DETECTION AND RESOLUTION BASED ON DEEP REINFORCEMENT LEARNING

1.1. A deep reinforcement learning approach to general problems

Reinforcement learning [19] is a method in which an agent interacts with an environment to maximize a long-term reward. It can be treated as a Markov decision process (S, A, T, R, γ) with the following notations: S is the set of environment states; A is the set of agent's actions; T is the probability of transition between states; R is the reward function; finally, γ is the discount rate.

Being in a state $S_t \in S$ at a training step t, the agent generates an action $A_t \in A$ following a policy π: $S \times A \to R$. Then the agent receives the reward R_t and passes to the next state S_{t+1}. The agent's goal is to maximize the total reward by learning the policy $\pi : S \to A$ that determines the required action in each state.

The states are assessed using a function $V(S)$. The state-value function S_t is updated by the formula

$$V(S_t) \leftarrow V(S_t) + \alpha\left(R_t + \gamma V(S_{t+1}) - V(S_t)\right),$$

where α is the learning rate.

The algorithm involves two neural networks: a critic network and an actor network. The former network updates the parameters of the value function w, whereas the latter network updates the parameters of the policy θ; for details, see [20].

Due to the infinite state space, the state-value function is approximated. The approximation is based on the deep learning of the neural network [21]:

$$\hat{V}(S, w) \approx V_n(S),$$

where w denotes the weights of neurons. In one-step temporal difference learning with the approximate function, the update formula is given by

$$w \leftarrow w + \alpha\left(R_t + \gamma \hat{V}(S_{t+1}, w) - \hat{V}(S_t, w)\right)\nabla_w \hat{V}(S_t, w).$$

The action is selected using a policy $\pi(a_s, \theta)$. For the space of continuous actions, the gradient policy [22] based on the gradient descent method is often adopted. The differentiable policy is defined as $\pi(a_s, \theta)$ and the parameters θ are updated at each step as follows:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \ln \pi(A_t | S_t, \theta) V(S_t).$$

1.2. Conflict detection and resolution (CD&R): problem description [23]

Consider N aircraft in a particular air traffic scenario (Fig. 1). Of these, $(N-1)$ ones are in a sector with radius L, and another aircraft flies into the sector. Each aircraft has an initial position and a target position. The goal of each aircraft is to fly from the initial position to the target position in the minimum time without conflicts with other aircraft. A conflict arises when the distance between two aircraft is below the minimum safe distance (usually 5 nautical miles (nm)). At step t, the position and course angle of the aircraft form the vector $\left(x_n(t),\ y_n(t), \varphi_n(t)\right)$.

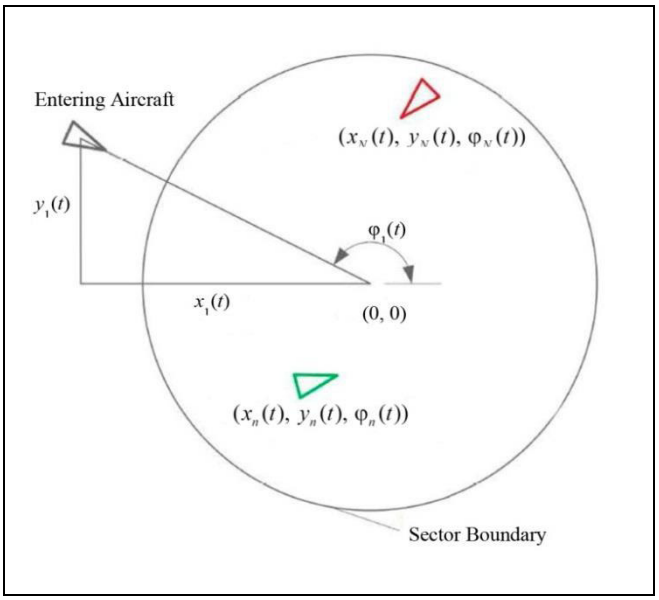


Fig. 1. Conflict detection and resolution problem.

The action for passing to a new state is the new position where the aircraft will fly from its current position.

Figure 2 shows the learning process for one episode.

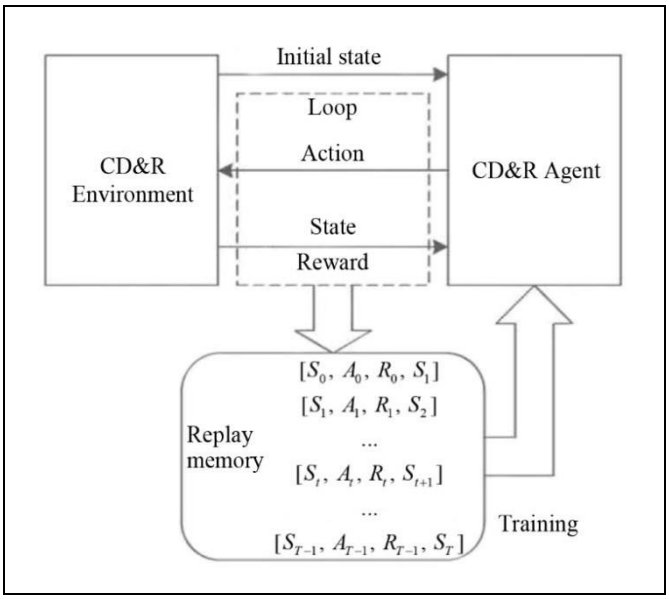


Fig. 2. Training process for one episode.

At the beginning of each training episode, the state S_0 is initialized in the environment. At each step t, the agent obtains the state S_t and implements the action A_t. After that, the environment passes to the state S_{t+1} and returns the reward R_t. This process is repeated until reaching the terminal state S_{t+1}. The value sets

$[S_t, A_t, R_t, S_{t+1}]$ are stored in memory. The agent selects data from memory and learns according to the algorithm.

Possible maneuvers include turns, vertical adjustments, and speed changes. The output state of the environment includes the position of each aircraft in the sector. The control experiment with one aircraft serves for checking that the environment can be used to train agents. Altitude and speed are fixed in this experiment.

The states form a vector of the dimension $N = N_n \times N_p \times N_d$, where N_n is the total number of aircraft (the incoming aircraft plus all aircraft in the sector), N_p is the number of aircraft waypoints (including all waypoints from epy current position to destination), and N_d is the dimension of aircraft location (equals 3). The height dimension is fixed, and the other two dimensions are variable. At each step, the agent perceives the state vector and, after normalization, takes it as the input of neural networks.

The action set is defined as

$$A = \{\rho, \varphi | \rho \in [0, L], \varphi \in [-\pi, \pi]\},$$

where L is the radius of the sector and ρ and φ are the polar radius and angle, respectively. An action is a position described by the two-dimensional polar coordinate. At each step t, the agent chooses an action $A_t \in A$. Depending on this action, the incoming aircraft flies from its current position to A_t.

The agent's goal is to maximize the long-term reward and update the parameters of the neural networks according to the immediate reward. Four rules are used to create the reward function: no conflict between the aircraft, the minimum control time, the minimum course angle change, and the minimum flight distance.

The reward function is given by

$$R_t = \begin{cases} -1 & \text{if a conflict occurs,} \\ |\Delta\varphi_t / \pi| & \text{otherwise,} \end{cases}$$

with the following notations: $\Delta\varphi_t \in [-\pi, \pi]$ is the change of the course angle at step t. Thus, $\left|\frac{\Delta\varphi_t}{\pi}\right| \le 1$, meaning that conflict resolution has the highest priority. Since a change in the course angle will affect the distance, the latter characteristic is omitted in the reward function.

The action is to determine a polar coordinate where the sector's center is the pole and the length shorter than the sector radius is the polar radius. The four outputs of the agent's neural network, $\mu_\rho, \sigma_\rho, \mu_\varphi,$ and σ_φ, are the mean and standard deviation of the polar radius and polar angle, respectively. For learning, the radius and angle are supposed to have the Gaussian distribution: $\rho \sim N(\mu_\rho, \sigma_\rho)$ and $\varphi \sim N(\mu_\varphi, \sigma_\varphi)$. They are generated, and the two-dimensional action is formed. After the agent's neural network is well trained, μ_ρ and μ_φ are taken as ρ and φ, respectively.

The neural networks are trained as follows. For the critic network, the parameter δ is determined to evaluate the chosen action:

$$\delta_t = R_t + \gamma \hat{V}(S_{t+1}, w) - \hat{V}(S_t, w),$$

where R_t is the immediate reward and $\hat{V}(S_t, w)$ and $\hat{V}(S_{t+1}, w)$ are the values of the current and next states, respectively.

The parameters w are updated using the least squares method:

$$w \leftarrow w + \alpha^w \nabla \delta^2.$$

The policy gradient method is applied for the actor network. The policy equation has the form

$$\ln \pi(\rho_t, \varphi_t | S_t, \theta) = \ln \pi(\rho_t | S_t, \theta) + \ln \pi(\varphi_t | S_t, \theta)$$

with the following notations: $\ln \pi(\rho_t, \varphi_t | S_t, \theta)$ is the probability of choosing ρ and φ in the state S_t with parameters θ; $\ln \pi(\rho_t | S_t, \theta)$ is the probability of choosing ρ in the state S_t with the parameters θ; finally, $\ln \pi(\varphi_t | S_t, \theta)$ is the probability of choosing φ in the state S_t with the parameters θ. The parameters θ are updated as follows:

$$\theta \leftarrow \theta + \alpha^\theta \delta_t \nabla \ln \pi(\rho_t, \varphi_t | S_t, \theta).$$

The effectiveness of the proposed approach was demonstrated by numerical simulations. As was declared by the authors, a well-trained agent can generate a solution within 200 ms, whereas previous methods require tens or even hundreds of seconds for calculation. In addition, the turning radius of the aircraft is properly considered, which corresponds to realistic situations.

2. A HYPER-HEURISTIC APPROACH TO REDUCE AIR TRAFFIC COMPLEXITY [24]

The paper [24] considered the problem of improving the airspace structure by air traffic complexity mitigation; see *Section 2.6*.

The hyper-heuristic approach does not simply follow a particular meta-heuristic but also involves flexible integration and adaptive control of low-level heuristics. Several studies confirmed the effectiveness of

Q-learning when selecting an appropriate low-level heuristic at the decision point. Q-learning lies in evaluating the best state-action pair using memory with Q-tables. Each entry in such a table expresses the long-term value of choosing a particular action in a particular state.

The traditional hyper-heuristic selection structure consists of two levels. The first level contains the problem representation, the state-value function, and a set of low-level heuristics. The second level performs two separate tasks as follows. First, it selects a low-level heuristic and applies it to the solution. Second, it decides to accept or reject the new solution. Selecting appropriate heuristics and decision methods is a non-trivial task when developing a robust hyper-heuristic model. A hyper-heuristic operates without any information needed about the functionality of the low-level heuristics but provides useful feedback on the utilization rate of each heuristic and the change of the objective function. This information is crucial for the learning process.

The algorithm uses a Q-learning agent to select a heuristic operator or a low-level heuristic (Fig. 3). The latter is then applied to generate a candidate solution and recalculate its performance in a simulation system.

A decision is then made regarding the candidate solution. If the candidate solution can improve performance, the current solution will be updated. Otherwise, the candidate solution will also be updated with some probability, decreasing with the course of learning. If the candidate solution is not accepted, all changes in the solution space associated with that candidate solution will be undone by a return operation. The Q-learning agent determines the reward by checking the current state, the chosen operator, and the evolution of solutions and then updates the Q-values in the Q-table.

The following heuristic operators are proposed (Fig. 4).

- h_1, a local search that randomly changes the departure time by no more than 5 min (Fig. 4a);
- h_2, a local search that flips the current waypoints in the horizontal plane XY along the trajectory (Fig. 4b);
- h_3, a local search that flips the current waypoints in the horizontal plane XY perpendicular to the trajectory (Fig. 4c);
- h_4, a local search that randomly deviates the route by changing the position of each waypoint (Fig. 4d);

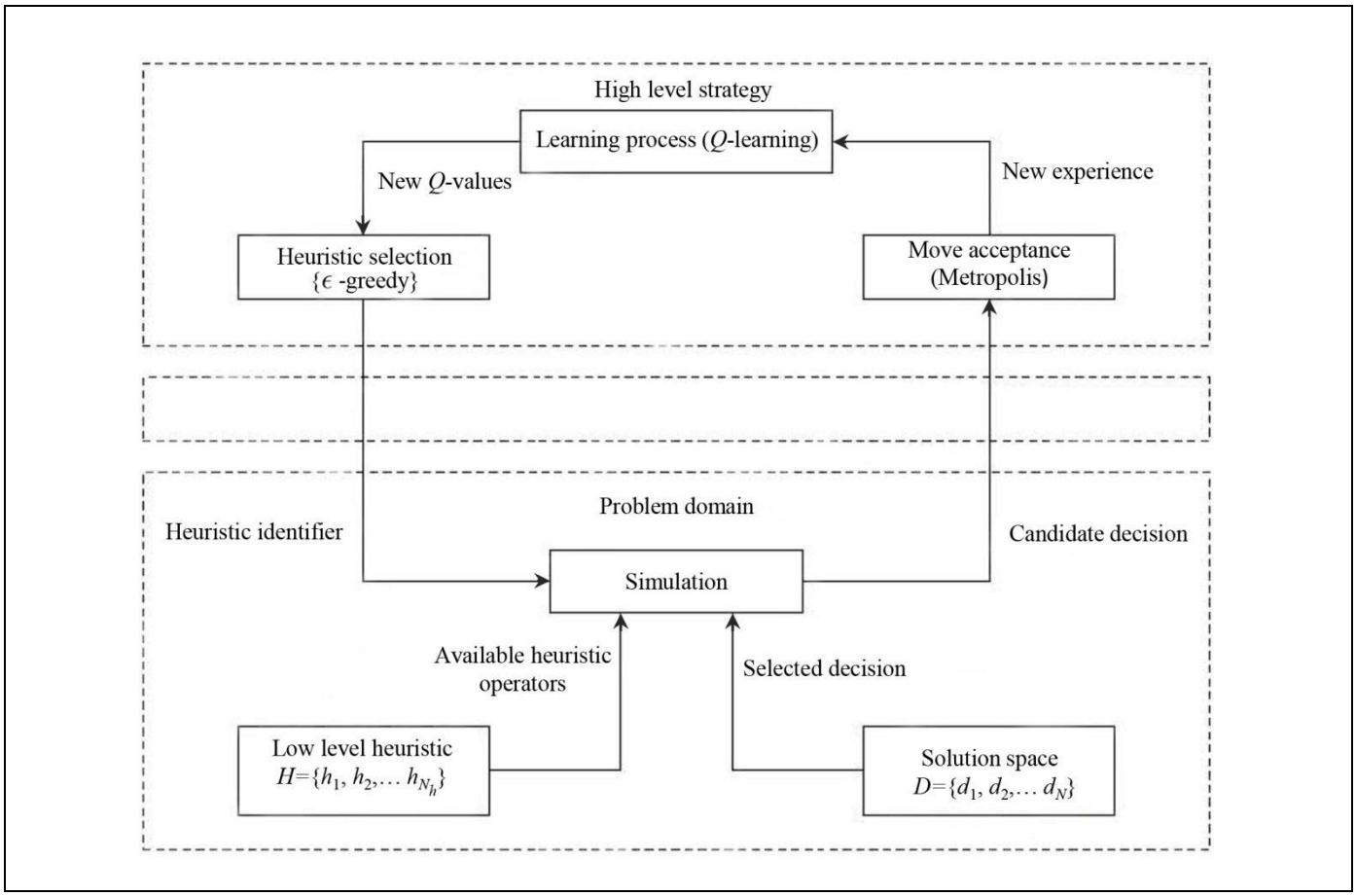


Fig. 3. Framework of Hyper-heuristic based on Q-learning with the high-level strategy and problem domain.

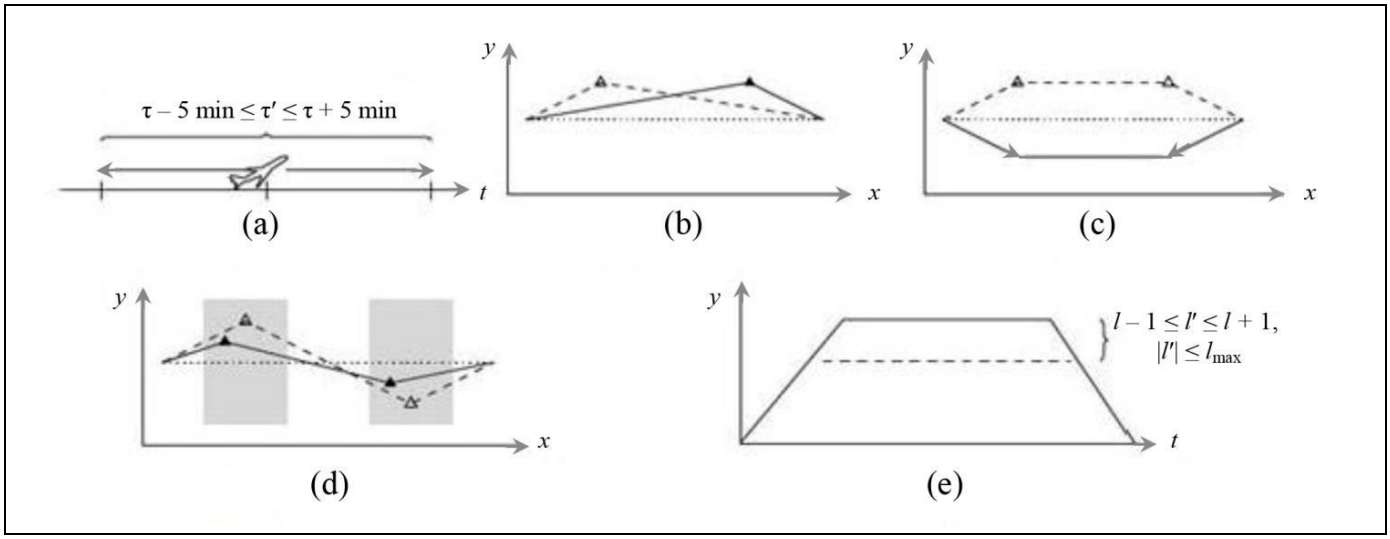


Fig. 4. Representation of intensification heuristic operators which allow the algorithm to refine the search in the vicinity of the current decision:

• h_5, a random selection of a neighbor flight altitude level within the maximum permissible limits (Fig. 4e).

Three generation operators diversify optimization as follows:

• h_6, an operator that randomly changes the departure time within the maximum permissible limits;

• h_7, an operator that randomly adds/removes one or more waypoints according to the problem constraints;

• h_8, an operator that randomly changes the flight altitude level within the maximum permissible limits.

The *Q*-learning agent follows a ϵ-greedy approach to select the heuristic operator based on the *Q*-table. A random action is chosen with probability ϵ and the action based on the *Q*-table with probability $(1-\epsilon)$. At first, ϵ is assigned a maximum user-defined value $\epsilon_{\max}$. During the learning process, it decreases until reaching a minimum user-defined value $\epsilon_{\min}$.

The learning cycle begins by selecting one of the diversification operators h_6, h_7, or h_8. Each state is defined based on the operator applied previously.

Seven states are considered as follows:

• s_0: one of the diversification operators h_6, h_7, or h_8 was previously applied;

• s_1: the operator h_1 was previously applied;

• s_2: the operator h_2 was previously applied;

• s_3: the operator h_3 was previously applied;

• s_4: the operator h_4 was previously applied;

• s_5: the operator h_5 was previously applied;

• s_6: two consecutive operators from the set $\{h_1, h_2, h_3, h_4, h_5\}$ were previously applied without changing the current solution.

The *Q*-table is initialized with *Q*-values according to Table 1.

Some *Q*-values are set to 1 to ensure that a particular heuristic operator can be selected; others are set to 0 to ensure the transition from one state to another.

At each iteration, the system stores the experience containing the current states with the selected operator *h* and the payoff *g* (the difference between the values of the new and previous solutions). At the end of the cycle, the *Q*-learning agent determines the rewards for updating the *Q*-values in the *Q*-table.

At step *t*, the *Q*-values for the found state–action pair are updated as follows:

$$Q(s_t, h_t) = (1-\alpha)Q(s_t, h_t) + \alpha\left(r_t + \gamma \max_{h\in H} Q(s_{t+1}, h)\right),$$

where $\alpha \in [0, 1]$ and $\gamma \in [0, 1]$ are the learning and discount rates, respectively, r_t is the immediate reward, and $H = \{h_i, i = 1,\ldots,8\}$ denotes the set of heuristic operators.

Transfer acceptance determines whether to accept or reject the new solution at each step of the search process. Iterations continue until satisfying a termination criterion.

In [24], the proposed approach was empirically assessed within a real-data experiment for a full day of traffic in the French airspace (8836 trajectories, see Fig. 5).

The initial trajectory plan for the full day of traffic was compared with the final trajectory plan calculated by the proposed algorithm in terms of complexity. The corresponding results are shown in Fig. 6.

Table 1

Q-table initialization

State	Operator							
	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8
s_0	1	1	1	1	1	–	–	–
s_1	0	1	1	1	1	–	–	–
s_2	1	0	1	1	1	–	–	–
s_3	1	1	0	1	1	–	–	–
s_4	1	1	1	0	1	–	–	–
s_5	1	1	1	1	0	–	–	–
s_6	–	–	–	–	–	1	1	1

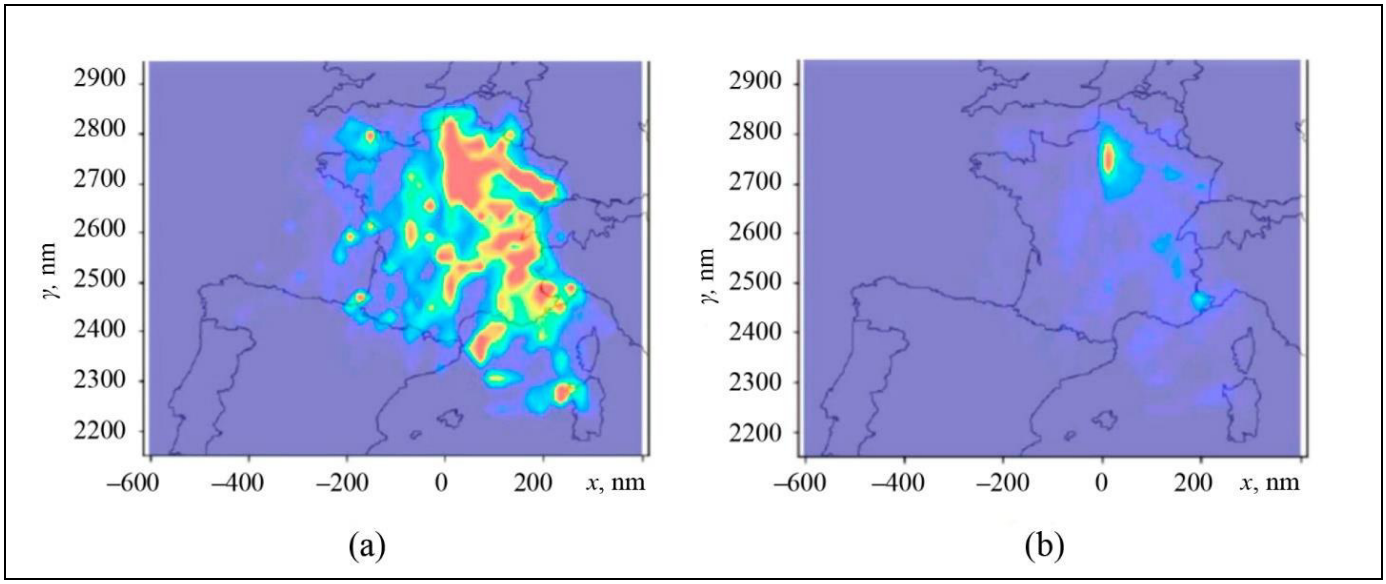


Fig. 5. Complexity map of (a) initial trajectories and (b) final trajectories of a full day of traffic in the French airspace.

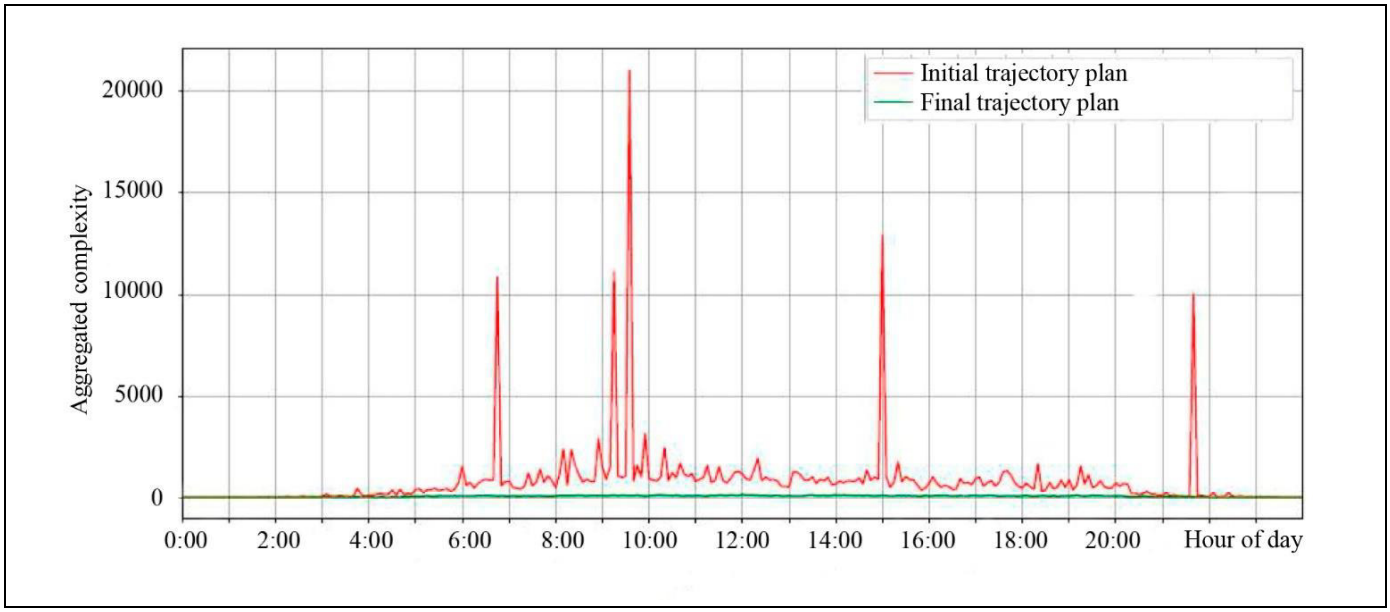


Fig. 6. Comparison of initial complexity and final complexity over time for a full day of traffic in the French airspace.

3. IMPROVING RUNWAY EFFICIENCY USING REINFORCEMENT LEARNING

Airport runways are a major bottleneck in air traffic and a key factor determining airport capacity. Building a new runway is not always possible. One approach to solving the capacity problem is the modernization of the airspace structure and airfield infrastructure. In [25], such an approach was implemented by mathematical modeling. This survey considers another approach: optimizing the use of infrastructure through the improved planning of aircraft landings.

The landing optimization problem is solved in three steps. First, an initial schedule is created on the first-come, first-served basis. Then this schedule is modified during the landing approach phase and finally frozen when the aircraft reaches the final stage of this phase. The initial schedule includes aircraft within the range of the airport landing radar (a time horizon of about 40 min before landing). The update process is executed each time a new aircraft enters the radar range to improve the landing schedule [26].

The most common requirements include a safe separation between consecutive aircraft, allowed time intervals determined by the earliest and latest flight times based on fuel consumption, and priority constraints. Different objective functions serve for increasing runway capacity, meeting schedules, minimizing fuel consumption, etc.

As is known, optimal aircraft sequencing and landing are an *NP*-hard problem [27]. Consequently, the solution time by exact methods grows rapidly with increasing the number of aircraft. Since the first solution [28] published in 1976, several new models and approaches have appeared in the literature, including genetic and heuristic algorithms to obtain a suboptimal but sufficiently efficient solution in an acceptable time [29, 30]. The survey [31] was devoted to some exact approaches to the problem (mainly mixed integer programming), whereas the paper [32] overviewed approximate solution methods, mainly genetic and memetic algorithms. A recent promising approach to the problem is based on reinforcement learning.

The authors [33] considered the problem of planning aircraft takeoffs on a single runway to observe the established time intervals. The problem was modeled as a Markov decision process and solved using the *Q*-learning algorithm [34] as follows. Let the agents be the aircraft and let their states be the aircraft position on the ground depending on its phase (parking, taxiing, and takeoff). The action is to delay the aircraft, and the reward is defined to minimize the delay during taxiing with observing the time intervals established for the aircraft. The algorithm was tested on real data from John F. Kennedy International Airport (JFK, New York), which included departures of 698 flights (two days of operation). Note that 42 training scenarios were generated from the data. According to the results, the algorithm has a performance similar to or greater than that of air traffic controllers.

The paper [35] proposed a framework to model the problem of aircraft sequencing and separation in accordance with the NASA sector-33 application [36]. This air traffic management application contains 35 examples of tasks involving up to 5 aircraft, including speed and route control for aircraft.

The proposed model consists of agents, states, actions, and rewards. There are two types of agents: parent and child. The parent agent's state contains a snapshot of the game screen. The child agent's state contains information about the measurement target, speed and acceleration of the aircraft, and route identifier in addition to information about the *N* closest agents to allow communication between agents. The actions for the parent/child agent are to change or maintain the route/speed of the aircraft. The reward is designed to penalize conflicting agents (separated by less than 3 nm).

The problem within the model was solved using a hierarchical deep learning algorithm with reinforcement. This algorithm combines the *Q*-learning algorithm [18] and neural networks [5]. It has a hierarchical nature because the actions are executed at two levels: the parent level selects the route and then the child level selects the speed for the aircraft. According to the tests in the NASA application (involving 2–5 aircraft), the proposed approach is viable.

3.1. The Mathematical Model

Let us consider in detail the approach [37]. The distributed algorithm proposed therein is based on *Q*-learning with the parameters optimally tuned by a genetic algorithm. The algorithm was implemented using the sliding window mechanism.

Consider a graph $G(N, L)$, where *N* and *L* are the sets of nodes and links, respectively. The node set has two subsets: $N_e \subset N$ (the entry points of the Terminal Maneuvering Areas (TMA)) and $N_r \subset N$ (runways). The final links connecting the runways are also grouped into a link subset: $L_r \subset L$.

Figure 7 shows the network at Paris Charles de Gaulle Airport (CDG). Aircraft enter at LORNI or OKIPA points; there are two runways, 27R and 26L, and two merge points, IF_27R and IF_26L. The two-point merge system is used (IF_27R–RWY_27R and IF_26L–RWY_26L, respectively).

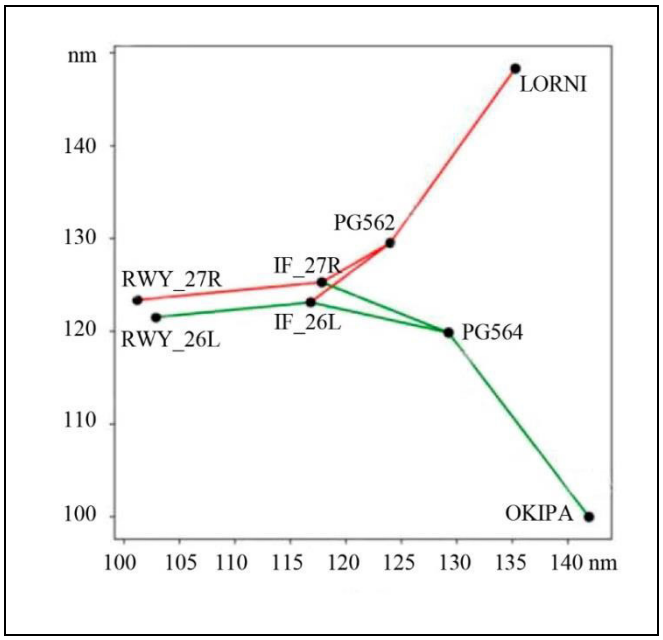


Fig. 7. Simplified STAR model at CDG: aircraft enter at LORNI or OKIPA. Merge points are located on both IF 27R and IF 26L.

For each aircraft, the procedure is executed at a constant speed. This is the landing speed that depends on the wake vortex category.

The point merge system (PMS) structure is presented in Fig. 8. For each aircraft, the length of the PMS arc will be treated as a decision variable for the algorithm.

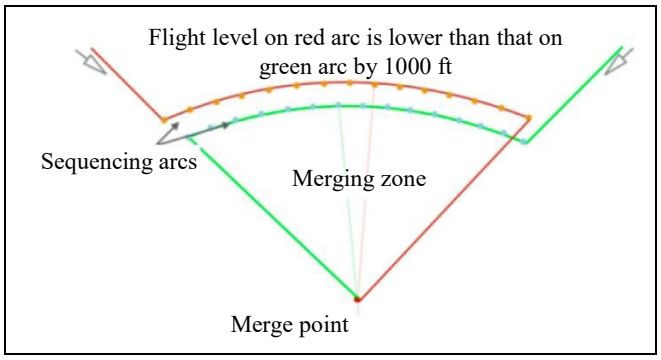


Fig. 8. Merge Point Topology: for each aircraft, the length of the flown sequencing arc is a decision variable.

A flight f is characterized with the following information:

- $V_{0,f}$ is the initial true airspeed of the aircraft;
- $t_{0,f}^{\mathrm{TMA}}$ is the initial entry time in the TMA;
- $r_{0,f} \in N_r$ is the runway on which the aircraft is planned to land;
- t_f^{RTA} is the time at which the aircraft is required to land (the *required time of arrival*, RTA);
- C_f is the wake vortex category.

For each flight f, the following decision variables are considered:

- V_f is the speed of the aircraft;
- t_f^{TMA} is the entry time in the TMA;
- r_f is the runway assigned for landing;
- l_f^{MP} is the length of the merge point arc.

The speed of the aircraft has to stay in a given range of the initial speed:

$$V_f \in V_{0,f} + p\Delta V,$$

where p is the number of increments and ΔV is the speed increment:

$$p \in Z,\ p\Delta V \in \left[\Delta V^{\min}, \Delta V^{\max}\right].$$

Here, $\Delta V^{\max}$ and $\Delta V^{\min}$ are the maximum speed increase and decrease from $V_{0,f}$, respectively, that can be assigned to an aircraft. The minimum speed decrease depends on the wave vortex category.

The entry time decision corresponds to a delay that can be absorbed in the En-Route airspace before the aircraft enters the TMA. In this airspace, the aircraft can be slowed down or accelerated in a given range. As a result, the entry time in the TMA could also change in a given range:

$$t_f^{\mathrm{TMA}} \in t_{0,f}^{\mathrm{TMA}} + p\Delta T,$$

where p is the number of increments and ΔT is the time increment:

$$p \in Z,\ p\Delta T \in \left[\Delta T^{\min}, \Delta T^{\max}\right].$$

Here, $\Delta T^{\max}$ and $\Delta T^{\min}$ are the maximum and minimum time increments from $t_{0,f}^{\mathrm{TMA}}$, respectively, that can be assigned to an aircraft.

To keep a balanced flow between runways, it may sometimes be more appropriate to change the landing runway of an aircraft ($r_f \in N_r$).

As the network contains merge points, one of the decision variables, l_f^{MP}, is the length of the arc that an aircraft will fly in one of the merge points, i.e.,

$$l_f^{\mathrm{MP}} \in p\Delta L,$$

where p is the number of increments and ΔL is the length increment:

$$p \in N,\ p\Delta L \le L_{MP}^{\max},$$

where $L_{\mathrm{MP}}^{\max}$ is the maximum arc length that a merge point can have.

3.2. Description of the Deep Learning Algorithm [37]

This section describes a deep learning algorithm based on the model presented in Section 3.1. It resolves potential aircraft conflicts during heavy traffic in the airport area in a reasonable time.

Each flight is a Markov decision process $MDP\{S, A, P_a, R_a\}$. All decision variables represent the state space S. This means that for every aircraft, a state is defined by {speed, entry time in the TMA, PMS arc length, runway assignment}. In each state, the following actions are considered: A = {increasing/decreasing the speed, increasing/decreasing the entry time in the TMA, increasing/decreasing the PMS arc length, changing the landing runway, no action}. For states that are not direct neighbors to the current state, the value of the transition function is 0. For neighbor states, the transition function is an equiprobabilistic one:

$$P_a(s, s') = \begin{cases} 0 \text{ if } s' \text{ is not a neighbor of } s, \\ \dfrac{1}{Card(A)} \text{ otherwise,} \end{cases}$$

where $Card(A)$ is the number of elements in A (in this case, 8).

Q-learning is a model-free reinforcement learning algorithm. This means that the algorithm does not need a model of the environment, it only interacts with the environment without knowing it. Every aircraft is considered an agent, which makes the algorithm multi-agent.

Q-learning is used to learn the optimal policy of a Markov decision process. This is done by computing the Q-function for each aircraft, i.e., representing the expected reward an agent can receive if he takes a given action in a given state. The Q-learning used is distributed, meaning that the reward of each agent is treated individually at each iteration.

For each agent, the expected reward $Q(s, a)$ in a given state s for a given action a is updated as follows:

$$Q(s, a) = Q(s, a) + \alpha\left(R + \gamma \max_{a'} Q(s', a') - Q(s, a)\right).$$

where s' is the new state when the action a is taken in the state s; R is the reward the agent will receive by making the action a in s; α is the learning rate; finally, γ is the discount factor.

The expected reward $Q(s, a)$ in a given state s for a given action a is updated at each iteration considering an estimation of the optimal future value $\max_{a'} Q(s', a')$. This is done independently of the policy being followed. Precisely, this is a one-step algorithm since the estimation is done only by looking one iteration ahead.

For a state $s \in S$, an action $a \in A$, and a parameter T called temperature, the probability $\pi(s, a)$ to choose a in s is given by

$$\pi(s, a) = \frac{e^{Q(s,a)/T}}{\sum_{a' \in A} e^{Q(s,a')/T}}.$$

The temperature at iteration k is given by a geometric law of the parameter β, i.e., $T_k = T_0\beta^k$, where T_0 is the initial temperature. This temperature sets a trade-off between exploration and exploitation: a relatively high temperature will promote the exploration of the Q-table, whereas a low temperature will be in favor of the exploitation of the Q-table.

In this distributed Q-learning, every aircraft is considered a learning agent and consequently has a Q-table. All the Q-tables are initialized at a value Q_0 chosen relatively low to enforce the state exploration. This is done on purpose since the reward (and Q-table) of an aircraft depends on agents close to it (which can be in conflict). Every agent is seen as an independent learner and does not consider the chosen action of other agents but only their actual states. Therefore, between the two decisions of an agent, its environment may have been changed. An agent can choose the specific action of doing nothing and then its state will not change.

For each aircraft, a reward function is computed and then used by the reinforcement learning algorithm. The reward given at each state and action depends on the other aircraft's state and is computed as the weighted sum of the rewards described below.

All rewards are negative (penalties):

$$R = \omega_{\text{RTA}}\left(R_{\text{RTA}} + 5R_{runway}\right) + \omega_{conflict}(\sum R_{link} + \sum R_{node}).$$

If an aircraft f does not land on $r_{0,f}$, its preferred runway, the reward added is 5 times the value R_{runway} weighted by ω_{RTA}. Note that ω_{RTA} and $\omega_{conflict}$ are the algorithm parameters.

Different components of the reward function are described below.

• Required Time of Arrival. All airlines have a schedule for each aircraft and on-time aircraft should have a better reward. Then, a reward corresponding to the absolute difference between the RTA and the real arrival time is added for every aircraft:

$$R_{\text{RTA}} = -\left|t_f^{\text{RTA}} - t_{arrival}\right|.$$

- Runway number

$$R_{runway} = \begin{cases} 0, & \text{landing on the required runway } r_{0,f}, \\ -1, & \text{otherwise.} \end{cases}$$

• Conflicts. The model considers two kinds of conflicts: link conflict, when two aircraft do not respect the wake vortex category separation, and node conflict, when the aircraft do not respect observe the horizontal separation at merge points [38] (3 nm). For each link, at the entrance and the exit, the minimum separation between two aircraft f and g must correspond to Table 2.

Table 2

Minimum separation for link conflict, nm

Category		Leading aircraft, f		
		Heavy	Average	Light
Trailing aircraft, g	Heavy	4	3	3
	Average	5	3	3
	Light	6	5	3

Assuming that $s_{f,g}$ is the minimum separation and $d_{f,g}$ is the actual distance between the leading aircraft f and the trailing aircraft g (Fig. 9), the criticality of a potential conflict, C_{link}, is proportional to the distance between the aircraft. Overtakings are also calculated; if this occurs, then $d_{f,g} < 0$ and the criticality of the conflict is set to –1:

$$C_{link} = \begin{cases} -1 \text{ if } d_{f,g} < 0, \\ -\dfrac{\left| s_{f,g} - d_{f,g} \right|}{s_{f,g}} \text{ if } d_{f,g} < s_{f,g}, \\ 0 \text{ otherwise.} \end{cases}$$

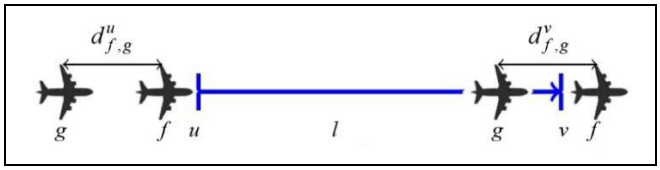


Fig. 9. Link conflict detection based on the comparison of distance between aircraft at the beginning or the end of a link with the separation minima.

The function C_{link} is piecewise linear and continuous, which is necessary for the learning algorithm to know if the conflict is getting better or worse. Since C_{link} can be close to 0, the learning algorithm can improve R_{RTA} instead of resolving the conflict. To prioritize the conflict resolution objective, the value of the reward function for the link is artificially set between –0.3 and –1 using the formula

$$R_{link} = -0.3 + \left(C_{link} \cdot (1 - 0.3) \right).$$

If there is no link conflict between two aircraft f and g, conflicts may still occur on nodes. In TMA, every aircraft has to be separated by 3 nm from others in order to respect the separation distance. As was shown in [38], in many airports, due to the network geometry, the detection area can be reduced to a circle of 2.2-nm radius. As for the links, the criticality of a node conflict is given by

$$C_{node} = \begin{cases} -\dfrac{2.2 - d_{f,g}}{2.2} \text{ if } d_{f,g} < 2.2, \\ 0 \text{ otherwise.} \end{cases}$$

As for the links, the value of the reward function for a node conflict is artificially set between –0.3 and –1 using the formula

$$R_{node} = -0.3 + \left(C_{node} \cdot (1 - 0.3) \right).$$

In this problem, if an aircraft enters the TMA many hours before another one, their decisions can be considered independent. Therefore, the dynamic aircraft landing optimization problem is solved on the basis of a sliding window.

During the optimization process, the aircraft inside the sliding window are divided into four groups:

• completed, the latest landing time is before the starting time of the sliding window;

• ongoing, the earliest entry time is before the starting time of the sliding window (decisions to land have already been made);

• active, the earliest and latest entry times are in the window;

• planned, the latest entry time is after the end of the sliding window.

At each iteration of the sliding window, the optimization algorithm is run on active flights.

Running the algorithm on every active flight in the sliding window is not efficient enough; some of the active flights may have good rewards, and other aircraft may have multiple conflicts. To speed up the optimization process, decisions are changed with a higher priority on aircraft with the worse reward. Those aircraft are indicated as critical flights. They are computed using a threshold that is greater than 70% of the worst aircraft reward. Since these aircraft are learning, their rewards decrease and more and more aircraft become "critical."

The algorithm was successfully tested on data from Paris Charles de Gaulle airport with the total number of aircraft landings artificially increased to 687. A conflict-free solution for a full day of traffic was calculated in less than 30 s, which is acceptable for real-time planning.

CONCLUSIONS

For several decades, extensive research was conducted on decision support automation in ATM systems. Mathematical models developed for this problem either minimize the number of potential conflicts between 4D aircraft trajectories or redistribute aircraft flows to reduce airspace congestion. The number of potential aircraft conflicts is often decreased using one or several methods as follows: shifting flight departure times, regulating airspeeds, changing flight trajectories, and changing flight altitude.

As shown, minimizing the number of potential aircraft conflicts is an *NP*-hard problem. Consequently, various metaheuristic algorithms emerged to solve it. A hybrid metaheuristic approach based on the simulated annealing algorithm, improved by local search methods, was developed for the strategic planning of air traffic flows considering the uncertainty of aircraft positions.

The complexity and scale of minimizing the number of potential conflicts in airspace require new approaches to this problem. Some publications in recent years have been devoted to deep reinforcement learning methods for improving the safety and efficiency of air traffic. The effectiveness of the proposed approaches has been investigated using computational experiments, which have shown encouraging results. Further extensive research is needed to assess the applicability of these approaches in real-world conditions.

REFERENCES

1. Kulida, E.L. and Lebedev, V.G., Methods for Solving Some Problems of Air Traffic Planning and Regulation. Part I: Strategic Planning of 4D Trajectories, *Control Sciences*, 2023, no. 1, pp. 1–12.
2. Degas, A., Islam, M.R., Hurter, C., et al., A Survey on Artificial Intelligence (AI) and eXplainable AI in Air Traffic Management: Current Trends and Development with Future Research Trajectory, *Applied Sciences*, 2022, vol. 12, no. 3, art. no. 1295. DOI: 10.3390/app12031295.
3. Wang, Z., Pan, W., Li, H., et al., Review of Deep Reinforcement Learning Approaches for Conflict Resolution in Air Traffic Control, *Aerospace*, 2022, vol. 9, no. 6, art. no. 294. DOI: 10.3390/aerospace9060294.
4. Brittain, M. and Wei, P., Autonomous Aircraft Sequencing and Separation with Hierarchical Deep Reinforcement Learning, *Proceedings of the 8th International Conference on Research in Air Transportation*, Barcelona, 2018. URL: https://www.reseachgate.net/publication/327287314.
5. Pham, D.T., Tran, N.P., Alam, S., et al., A Machine Learning Approach for Conflict Resolution in Dense Traffic Scenarios with Uncertainties, *Proceedings of the 13th USA/Europe Air Traffic Management Research and Development Seminar (ATM 2019)*, Vienne, 2019.
6. Pham, D.T., Tran, N.P., Alam, S., et. al., Deep Reinforcement Learning based Path Stretch Vector Resolution in Dense Traffic with Uncertainties, *Transportation Research. Part C. Emerging Technologies*, 2021, vol. 135, art. no. 103463. DOI: 10.1016/j.trc.2021.103463.
7. Tran, P.N., Pham, D.T., Goh, S.K., et al., An Interactive Conflict Solver for Learning Air Traffic Conflict Resolutions, *Journal of Aerospace Information Systems*, 2020, vol. 17, no. 6, pp. 271–277.
8. Ribeiro, M., Ellerbroek, J., and Hoekstra, J., Improvement of Conflict Detection and Resolution at High Densities through Reinforcement Learning, *Proceedings of the International Conference on Research in Air Transportation*, Tampa, 2020.
9. Brittain, M. and Wei, P., Autonomous Separation Assurance in a High-Density en Route Sector: A Deep Multi-Agent Reinforcement Learning Approach, *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)*, Aukland, 2019, pp. 3256–3262.
10. Brittain, M., Yang, X., and Wei, P., A Deep Multi-Agent Reinforcement Learning Approach to Autonomous Separation Assurance, *Arxiv:2003.08353v2*, 2020. DOI: https://doi.org/10.48550/arXiv.2003.08353.
11. Brittain, M. and Wei, P., One to Any: Distributed Conflict Resolution with Deep Multi-Agent Reinforcement Learning and Long Short-Term Memory, *Proceedings of the AIAA SciTech 2021 Forum*, Nashville, 2021, p. 1952.
12. Zhao, P. and Liu, Y., Physics Informed Deep Reinforcement Learning for Aircraft Conflict Resolution, *IEEE Transactions on Intelligent Transportation Systems*, 2021, vol. 23, iss. 7, pp. 8288–8301. DOI: 10.1109/TITS.2021.3077572.
13. Mollinga, J., and Hoof, H., An Autonomous Free Airspace En-route Controller Using Deep Reinforcement Learning Techniques, *Arxiv:2007.01599*, 2020. DOI: https://doi.org/10.48550/arXiv.2007.01599.
14. Khan, N.A., Brohi, S.N., and Jhanjhi, N., UAV's Applications, Architecture, Security Issues and Attack Scenarios: A Survey, *Intelligent Computing and Innovation on Data Science*, 2020, vol. 183, pp. 753–760. DOI: 10.1007/978-981-15-3284-9_86.
15. Szegedy, C., Zaremba, W., Sutskever, I., et al., Intriguing Properties of Neural Networks, *Arxiv:1312.6199v3*, 2013. DOI: https://doi.org/10.48550/arXiv.1312.6199.
16. Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K., Synthesizing Robust Adversarial Examples, *Proceedings of the International Conference on Machine Learning*, Stockholm, 2018, pp. 284–293.
17. Wang, L., Yang, H., Lin, Y., et al., Explainable and Safe Reinforcement Learning for Autonomous Air Mobility, *arXiv:2211.13474v1*, 2022. DOI: https://doi.org/10.48550/arXiv.2211.13474.
18. Messaoud, M., A Thorough Review of Aircraft Landing Operation from Practical and Theoretical Standpoints at an Airport Which May Include a Single or Multiple Runways, *Applied Soft Computing*, 2020, vol. 98, no. 12, art. no. 106853. DOI: 10.1016/j.asoc.2020.106853.
19. Sutton, R.S. and Barto, A.G., *Reinforcement Learning: An Introduction*, London: MIT Press, 2017.
20. Degris, T., Pilarski, P.M., and Sutton, R.S., Model-Free Reinforcement Learning with Continuous Action in Practice, *Proceedings of the American Control Conference*, Montréal, 2012, pp. 2177–2182.
21. LeCun, Y., Bengio, Y., and Hinton, G., Deep Learning, *Nature*, 2015, vol. 521, pp. 436–444.

22. Sutton, R.S., McAllester, D.A., Singh, S.P., et al., Policy Gradient Methods for Reinforcement Learning with Function Approximation, *Proceedings of the 12th Conference on Advances in Neural Information Processing Systems (NIPS 1999)*, Denver: MIT Press, 1999, pp. 1057–1063.
23. Wang, Z., Li, H., Wang, J., and Shen, F., Deep Reinforcement Learning Based Conflict Detection and Resolution in Air Traffic Control, *IET Intell. Trans. Syst.*, 2019, vol. 13, pp. 1041–1047.
24. Juntama, P., Delahaye, D., Chaimatanan, S., and Alam, S., Hyperheuristic Approach Based on Reinforcement Learning for Air Traffic Complexity Mitigation, *Journal of Aerospace Information Systems*, 2022, vol. 19, no. 9. DOI: 10.2514/1.i011048.
25. Vishnyakova, L.V. and Popov, A.S., Selection of Airspace Structure and Aerodrome Infrastructure During Their Modernization by Methods of Mathematical Modeling, *J. Comput. Syst. Sci. Int.*, 2021, vol. 60, no. 6, pp. 918–955. https://doi.org/10.1134/S1064230721060174.
26. Bennell, J.A., Mesgarpour, M., and Potts, C.N., Airport Runway Scheduling, *Semantic Scholar*, 2011, vol. 4OR, pp. 115–138. DOI: 10.1007/s10288-011-0172-x.
27. Prakash, R., Piplani, R., and Desai, J., An Optimal Data-Splitting Algorithm for Aircraft Scheduling on a Single Runway to Maximize Throughput, *Transportation Research, Part C: Emerging Technologies*, 2018, vol. 95, pp. 570– 581.
28. Dear, R.G., The Dynamic Scheduling of Aircraft in the Near Terminal Area, *Technical Report no. R76-9*, Flight Transportation Laboratory, Cambridge, MIT, 1976.
29. Kulida, E.L., Genetic Algorithm for Solving the Problem of Optimizing Aircraft Landing Sequence and Times, *Automation and Remote Control*, 2022, vol. 83, no. 3, pp. 426–436.
30. Kulida, E., Egorov, N., and Lebedev, V., Comparison of Two Algorithms for Solving the Problem Aircraft Arrival Sequencing and Scheduling, *Proceedings of the 14th International Conference "Management of Large-Scale System Development" (MLSD)*, September 27–29, 2021. URL: https://ieeexplore.ieee.org/document/9600243.
31. Veresnikov, G.S., Egorov, N.A. Kulida, E.L., and Lebedev, V.G., Methods for Solving of the Aircraft Landing Problem. I. Exact Solution Methods, *Automation and Remote Control*, 2019, vol. 80, pp. 1317–1334.
32. Veresnikov, G.S., Egorov, N.A. Kulida, E.L., and Lebedev,V.G., Methods for Solving of the Aircraft Landing Problem. II. Approximate Solution Methods, *Automation and Remote Control*, 2019, vol. 80, pp. 1502–1518.
33. Soares, I.B., De Hauwere, Y.M., Januarius, K., et al., Departure Management with a Reinforcement Learning Approach: Respecting CFMU Slots, *Proceedings of the IEEE 18th International Conference on Intelligent Transportation Systems*, Las Palmas de Gran Canaria, 2015.
34. Watkins, C.J. and Dayan, P., Q-learning, *Machine Learning*, 1992, vol. 8, pp. 279–292.
35. Brittain, M. and Wei, P., Autonomous Aircraft Sequencing and Separation with Hierarchical Deep Reinforcement Learning, *Proceedings of the International Conference for Research in Air Transportation*, Barcelona, 2018.
36. Colen, J., NASA sector 33 application, 2013. URL: https://www.nasa.gov/centers/ames/Sector33/iOS/index.html.
37. Henry, A., Delahaye, D., and Valenzuela, A., Conflict Resolution with Time Constraints in the Terminal Maneuvering Area Using a Distributed Q-learning Algorithm, *Proceedings of the International Conference on Research in Air Transportation (ICRAT 2022)*, 2022, Tampa, Hal-03701660.
38. Ma, J., Delahaye, D., Sbihi, M., and Mongeau, M., Integrated Optimization of Terminal Manoeuvring Area and Airport, *6th SESAR Innovation Days*, Delft, Netherlands, 2016.

This paper was recommended for publication by A.A. Lazarev, a member of the Editorial Board.

Received November 10, 2022, and revised December 19, 2022. Accepted December 20, 2022.

Author information

Kulida, Elena L'vovna. Cand. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ elena-kulida@yandex.ru

Lebedev, Valentin Grigor'evich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ lebedev-valentin@yandex.ru

Cite this paper

Kulida, E.L. and Lebedev, V.G., Methods for Solving Some Problems of Air Traffic Planning and Regulation. Part II: Application of Deep Reinforcement Learning. *Control Sciences* **2**, 2–14 (2023). http://doi.org/10.25728/cs.2023.2.1

Original Russian Text © Kulida, E.L., Lebedev, V.G., 2023, published in *Problemy Upravleniya*, 2023, no. 2, pp. 3–18.

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com

DOI: http://doi.org/10.25728/cs.2023.2.2

INTERVAL OBSERVER DESIGN FOR DISCRETE LINEAR TIME-INVARIANT SYSTEMS WITH UNCERTAINTIES[1]

A.N. Zhirabok[2], A.V. Zuev[3], and C.I. Kim[4]

[2,3] Institute of Marine Technology Problems, Russian Academy of Sciences, Far Eastern Branch, Vladivostok, Russia
[2-4] Far Eastern Federal University, Vladivostok, Russia

[2]✉ zhirabok@mail.ru, [3]✉ alvzuev@yandex.ru, [4]✉ kim.ci@dvfu.ru

Abstract. This paper considers the problem of constructing an interval observer for systems described by discrete-time linear models under uncertainties in the form of exogenous disturbances and measurement noise (unknown bounded functions). Such an observer is designed using the minimal-dimension model of the original system invariant with respect to the disturbances. The dynamic matrix of this model is defined in the identification canonical form. We present relations to design an interval observer of minimal complexity for estimating the set of admissible values of a given linear function of the state vector. If the observer invariant with respect to the disturbances does not exist, we suggest a method to construct an observer with minimal sensitivity to them based on the singular value decomposition of system matrices. Theoretical results are illustrated by an example.

Keywords: linear systems, uncertainties, models, interval observers.

INTRODUCTION

This paper is a logical continuation of the research work [1], which considered the design of interval observers for systems described by linear models with continuous time.

In recent years, numerous studies have been devoted to the design of interval observers; for a survey, see the publications [2, 3]. The papers [4–10] presented the solution of this problem for different classes of systems as well as practical applications. As a rule, the cited authors estimated the set of admissible values of the full state vector. However, in many cases, it is of interest to estimate only a given linear function of this vector. The corresponding interval observer turns out to be significantly simpler than the full-order counterpart, and the class of systems for which such an observer can be designed is wider. In addition, when estimating a given linear function, the observer dynamics can be represented in a canonical form, which simplifies the solution procedure and extends the class of systems with interval observers.

In what follows, we state and solve the interval observer design problem for time-invariant systems described by discrete linear dynamic models with exogenous disturbances and measurement noise. The resulting interval observer estimates the set of admissible values for a given linear function of the system's state vector. This paper therefore differs from [2–10], where interval observers were designed to estimate the full state vector.

1. BASIC MODELS AND PROBLEM STATEMENT

We consider a system described by the discrete linear model

$$\begin{aligned} x(t+1) &= Fx(t) + Gu(t) + L\rho(t), \\ y(t) &= Hx(t) + v(t), \end{aligned} \tag{1}$$

with the following notations: $x(t) \in R^n$, $u(t) \in R^m$, and $y(t) \in R^l$ are the state, control, and output vectors,

[1] This work was supported by the Russian Science Foundation, project no. 23-29-000191, https://rscf.ru/project/23-19-00191/

respectively; F, G, and H are constant matrices of dimensions $n\times n$, $n\times m$, and $l\times n$, respectively; L is a known matrix of dimensions $n\times q$; $\rho(t)\in R^q$ is an unknown bounded time-varying function that describes the disturbances affecting the system, and $\|\rho(t)\|\le\rho_*$ for all $t\ge 0$; finally, $v(t)\in R^l$ is an unknown bounded time-varying function that describes measurement noise, and $\|v(t)\|\le v_*$ for all $t\ge 0$. (The symbol $\|\cdot\|$ indicates the Euclidean norm.)

According to (1), the uncertainties in the problem are represented by the measurement noise $v(t)$ and the exogenous disturbance $\rho(t)$ with the upper bounds v_* and ρ_* of their amplitudes, respectively, for all $t\ge 0$.

It is required to design a minimal-order interval observer producing the lower $\underline{z}(t)$ and upper $\overline{z}(t)$ estimates of the linear function $z(t)=Mx(t)\in R^p$ of the state vector with a given matrix M so that inequality $\underline{z}(t)\le z(t)\le\overline{z}(t)$ will hold componentwise for all $t\ge 0$.

As was demonstrated in the paper [1], for continuous-time systems, an interval observer can be designed based on the minimal-dimension model by two methods. In the first method, the matrices describing this model are found in the identification canonical form (ICF); the observer's stability is ensured using feedback, and the observer is then reduced to the Jordan canonical form to provide the Metzler property of the matrix reflecting its dynamics. In the second method, this matrix is immediately found in the Jordan form, which considerably simplifies the problem: stability and the Metzler property directly follow from the Jordan form.

In the discrete-time case, the Metzler property is not required: the matrix under consideration has to be stable and nonnegative. The ICF satisfies these two requirements and is therefore is preferable here. In addition, the feedback may not be used for the observer's stability: the ICF has zero eigenvalues, ensuring stability in the discrete-time case.

The solution is based on a minimal-dimension model insensitive to the disturbance:

$$\begin{aligned} x_*(t+1) &= F_*x_*(t)+J_*Hx(t)+G_*u(t),\\ z(t) &= H_zx_*(t)+Qy_0(t). \end{aligned} \qquad (2)$$

This model estimates the variable $z(t)$ and has the following notations: $x_*\in R^k$ is the observer's state vector; k is the model dimension; F_*, J_*, G_*, H_z, and Q are matrices to be determined; finally, $y_0(t)=N_2y(t)$ for some matrix N_2 defined below. The vector $x(t)$ and the unknown vector $x_*(t)$ are related by

$$x_*(t)=\Phi x(t),$$

where the matrix Φ has to be determined. The term $J_*Hx(t)$ in formula (2) can be explained as follows. Being a reduced part of system (1), model (2) does not include the output vector $y(t)$. Hence, this vector appears as the term $J_*y(t)$ in the observer (12). Such an approach allows considering the measurement noise.

The solution of equation (2) insensitive to the disturbance $\rho(t)$ is the best in terms of the interval $\underline{z}(t)\le z(t)\le\overline{z}(t)$. As is known [11], it satisfies the condition $\Phi L=0$. To make the estimated variable $z(t)$ in model (2) insensitive to the disturbance, the variable $y_0(t)$ in this equation must be formed as follows.

Let us introduce a matrix L_0 of maximal rank such that $L_0L=0$. Then $\Phi=NL_0$ for some matrix N. Since the vector $x'(t)=L_0x(t)$ is insensitive to the disturbance, $y_0(t)=N_1x'(t)$ for some matrix N_1. On the other hand, $y_0(t)$ is part of the output vector $y(t)$, i.e., $y_0(t)=N_2y(t)$ for some matrix N_2. Then the matrices N_1 and N_2 satisfy the equation $N_1L_0=N_2H$. It has a nontrivial solution if

$$\operatorname{rank}\begin{pmatrix}H\\L_0\end{pmatrix}<\operatorname{rank}(L_0)+\operatorname{rank}(H).$$

Under this condition, the matrices N_1 and N_2 are determined from the equation

$$(N_1\vdots\ -N_2)\begin{pmatrix}L_0\\H\end{pmatrix}=0, \qquad (3)$$

where the symbol $\vdots$ separates two matrices. Otherwise, we should use $y(t)$ instead of $y_0(t)$ in model (2). As a result, the interval $(\underline{z}(t),\overline{z}(t))$ will be extended.

According to [11, 12], the matrices describing the model satisfy the equations

$$\Phi F=F_*\Phi+J_*H,\ \ G_*=\Phi G,\ \ \Phi L=0. \qquad (4)$$

An additional condition is due to the second equation in model (2). With $z(t)=Mx(t)$, we write it as

$$Mx(t)=H_z\Phi x(t)+QN_2Hx(t),$$

arriving at the equation

$$M = H_z\Phi + QN_2H = (H_z \vdots \ Q)\begin{pmatrix}\Phi\\ N_2H\end{pmatrix}. \quad (5)$$

It has a solution if

$$\operatorname{rank}\begin{pmatrix}\Phi\\ N_2H\end{pmatrix} = \operatorname{rank}\begin{pmatrix}\Phi\\ N_2H\\ M\end{pmatrix}. \quad (6)$$

Based on equations (4) and (5), we can obtain relations to analyze the existence of such a solution in several cases. The first of such conditions has the form [11]

$$\operatorname{rank}\begin{pmatrix}L_0F\\ H\\ L_0\end{pmatrix} < \operatorname{rank}\begin{pmatrix}H\\ L_0\end{pmatrix} + \operatorname{rank}(L_0F). \quad (7)$$

To derive the second one, let us replace the matrix Φ in equation (5) with NL_0. Then obvious transformations yield

$$M = (H_zN \vdots \ Q)\begin{pmatrix}L_0\\ N_2H\end{pmatrix}.$$

The resulting equation is resolvable if

$$\operatorname{rank}\begin{pmatrix}L_0\\ N_2H\end{pmatrix} = \operatorname{rank}\begin{pmatrix}L_0\\ N_2H\\ M\end{pmatrix}. \quad (8)$$

An algorithm to check these conditions includes the following steps:

1. Determine the matrix L_0 and find the matrix N_2 from equation (3).

2. Check conditions (7) and (8). If they are true, find the matrices H_z and Q from equation (5) and construct the model and observer insensitive to the disturbance.

3. If just condition (8) fails, examine each row M_i, $i = 1, 2, ..., p$, of the matrix M by replacing M in condition (8) with M_i. Take the rows satisfying this condition to form the matrix M_0 and then design an interval observer insensitive to the disturbance to estimate the variable $z_0(t) = M_0x(t)$. For the other rows of the matrix M, combined into the matrix M_*, find the robust solution described in Section 4 and then design a second observer to estimate the variable $z_*(t) = M_*x(t)$. This observer will have minimal sensitivity to the disturbance.

4. If just condition (7) fails, a robust solution is only possible. Find it by the methods described in Section 4. In this case, the interval observer will be minimally sensitive to the disturbance.

5. If conditions (7) and (8) fail both, the robust solution (see Section 4) is only possible as well. If the resulting matrix Φ satisfies condition (6), then the variable $y_0(t)$ (the undisturbed part of the vector $y(t)$) can be found. Otherwise, it is impossible, and the interval $(\underline{z}(t), \overline{z}(t))$ is further extended due to the term $Qy(t)$ corrupted by the disturbance.

2. MODEL CONSTRUCTION

The matrix F_* is found in the ICF:

$$F_* = \begin{pmatrix}0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & 0\\ 0 & 0 & 0 & \cdots & 0\\ \cdots & \cdots & \cdots & \ddots & \cdots\\ 0 & 0 & 0 & \cdots & 0\end{pmatrix}.$$

As is well known, the model is stable if the eigenvalues of the matrix F_* do not exceed 1 by magnitude. For the ICF under consideration, they equal 0.

The problem is solved based on the equation [11]

$$(\Phi_1 \vdots \ -J_{*1} \vdots \ \ldots \vdots \ -J_{*k})(V^{(k)} \vdots \ L^{(k)}) = 0, \quad (9)$$

where

$$V^{(k)} = \begin{pmatrix}F^k\\ HF^{k-1}\\ \vdots\\ H\end{pmatrix},$$

$$L^{(k)} = \begin{pmatrix}L & FL & \ldots & F^{k-1}L\\ 0 & HL & \ddots & HF^{k-2}L\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \ldots & 0\end{pmatrix}, \ k = 1, 2, \ldots,$$

and Φ_i and J_{*i} indicate the rows of the matrices Φ and J_*, respectively. Note that the matrix $V^{(k)}$ serves to construct model (2) whereas the matrix $L^{(k)}$ to ensure its insensitivity to the disturbances. Equation (9) has a nontrivial solution if

$$\operatorname{rank}(V^{(k)} \vdots \ L^{(k)}) < lk + n. \quad (10)$$

To design the model, we determine the minimum k from inequality (10) and the row $(\Phi_1 \vdots \ -J_{*1} \vdots \ \ldots \vdots \ -J_{*k})$ from equation (9). Then, based on the relations

$$\Phi_iF = \Phi_{i+1} + J_{*i}H, \ i = \overline{1, k-1}, \ \Phi_kF = J_{*k}H, \quad (11)$$

obtained from equations (4) and the ICF [11], we construct the matrix Φ. After that, condition (6) is verified. If it holds, the matrix M can be expressed through $(\Phi^{\mathrm{T}}\ H^{\mathrm{T}})^{\mathrm{T}}$, and the designed linear model will estimate the desired variable $z(t) = Mx(t)$; the matrices H_z and Q are determined from the algebraic equation (5) and the matrix G_* from equations (4). If condition (6) fails, another solution of equation (9) should be found for the same or increased dimension of the model. If it fails for all $k \le n$, the robust solution should be used; see Section 4.

3. INTERVAL OBSERVER DESIGN

The observer is found in the form

$$\begin{aligned}
\underline{x}_*(t+1) &= F_*\underline{x}_*(t) + J_*y(t) + G_*u(t) - |J_*|E_k v_*,\\
\overline{x}_*(t+1) &= F_*\overline{x}_*(t) + J_*y(t) + G_*u(t) + |J_*|E_k v_*,\\
\underline{z}(t) &= H_z\underline{x}_*(t) + Qy_0(t),\\
\overline{z}(t) &= H_z\overline{x}_*(t) + Qy_0(t),\\
\underline{x}_*(0) &= \underline{x}_{*0}, \quad \overline{x}_*(0) = \overline{x}_{*0},
\end{aligned} \tag{12}$$

where the matrix E_k of dimensions $k \times 1$ is composed of unities and the matrix $|J_*|$ is composed of the absolute values of the corresponding elements of the matrix J_*. By assumption, $x_*(0) \in [\underline{x}_{*0}, \overline{x}_{*0}]$ for some known vectors $\underline{x}_{*0}, \overline{x}_{*0} \in R^k$.

Theorem. *Let* $\underline{x}_*(0) \le x_*(0) \le \overline{x}_*(0)$. *Then the interval observer* (12) *satisfies the relations*

$$\underline{x}_*(t) \le x_*(t) \le \overline{x}_*(t) \ \textit{and}\ \underline{z}(t) \le z(t) \le \overline{z}(t)$$

for all $t \ge 0$, *where*

$$\begin{aligned}
\underline{z}(t) &= H_z\underline{x}_*(t) + Qy_0(t),\\
\overline{z}(t) &= H_z\overline{x}_*(t) + Qy_0(t)
\end{aligned} \tag{13}$$

for $H_z \ge 0$ *and*

$$\begin{aligned}
\underline{z}(t) &= H_z\overline{x}_*(t) + Qy_0(t),\\
\overline{z}(t) &= H_z\underline{x}_*(t) + Qy_0(t)
\end{aligned} \tag{14}$$

for $H_z \le 0$.

P r o o f. By analogy with [2], we introduce the estimation errors

$$\begin{aligned}
\underline{e}_*(t) &= x_*(t) - \underline{x}_*(t),\ \overline{e}_*(t) = \overline{x}_*(t) - x_*(t),\\
\underline{e}_z(t) &= z(t) - \underline{z}(t),\ \overline{e}_z(t) = \overline{z}(t) - z(t).
\end{aligned} \tag{15}$$

In view of (2) and (12), it is possible to obtain the difference equations

$$\begin{aligned}
\underline{e}_*(t+1) &= F_*\underline{e}_*(t) + J_*(Hx(t) - y(t)) + |J_*|E_k v_*\\
&= F_*\underline{e}_*(t) - J_*v(t) + |J_*|E_k v_*,\\
\overline{e}_*(t+1) &= F_*\overline{e}_*(t) + J_*(y(t) - Hx(t)) + |J_*|E_k v_*\\
&= F_*\overline{e}_*(t) + J_*v(t) + |J_*|E_k v_*.
\end{aligned} \tag{16}$$

Since $\underline{x}_*(0) \le x_*(0) \le \overline{x}_*(0)$, formula (15) implies $\underline{e}_*(0) \ge 0$ and $\overline{e}_*(0) \ge 0$. Note that in system (16), $|J_*|E_k v_* \pm J_*v(t) \ge 0$ for all $t \ge 0$ and $F_* \ge 0$. Given $\underline{e}_*(0) \ge 0$ and $\overline{e}_*(0) \ge 0$, its solutions will be elementwise nonnegative: $\underline{e}_*(t) \ge 0$ and $\overline{e}_*(t) \ge 0$ for all $t \ge 0$ [2]. Due to $z(t) = H_z x_*(t) + Qy_0(t)$, for $H_z \ge 0$, from (13) and (15) we have

$$\begin{aligned}
\underline{e}_z(t) &= z(t) - \underline{z}(t)\\
&= H_z x_*(t) + Qy_0(t) - (H_z\underline{x}_*(t) + Qy_0(t)) = H_z\underline{e}_*(t),\\
\overline{e}_z(t) &= \overline{z}(t) - z(t)\\
&= H_z\overline{x}_*(t) + Qy_0(t) - (H_z x_*(t) + Qy_0(t)) = H_z\overline{e}_*(t).
\end{aligned}$$

Considering the inequalities $\underline{e}_*(t) \ge 0$, $\overline{e}_*(t) \ge 0$, and $H_z \ge 0$, we obtain $\underline{e}_z(t) \ge 0$ and $\overline{e}_z(t) \ge 0$, which is equivalent to $\underline{z}(t) \le z(t) \le \overline{z}(t)$. In the case $H_z \le 0$, from (14) and (15) it follows that

$$\begin{aligned}
\underline{e}_z(t) &= z(t) - \underline{z}(t)\\
&= H_z x_*(t) + Qy_0(t) - (H_z\overline{x}_*(t) + Qy_0(t)) = -H_z\overline{e}_*(t),\\
\overline{e}_z(t) &= \overline{z}(t) - z(t)\\
&= H_z\underline{x}_*(t) + Qy_0(t) - (H_z x_*(t) + Qy_0(t)) = -H_z\underline{e}_*(t).
\end{aligned}$$

In view of $\underline{e}_*(t) \ge 0$, $\overline{e}_*(t) \ge 0$, and $H_z \le 0$, we finally arrive at $\underline{e}_z(t) \ge 0$ and $\overline{e}_z(t) \ge 0$. The proof of this theorem is complete. ♦

Remark 1. If the matrix H_z is indefinite, the final result remains the same, but the formulas for calculating the upper and lower bounds become more complicated. We consider two cases as follows.

- Let H_z be a row; without loss of generality, assume that its first p components are positive and the rest are negative: $H_z = (H_z^+ \vdots\ H_z^-)$. We define

$$\underline{z}(t) = H_z^+\underline{x}_{*(p)}(t) + H_z^-\overline{x}_*^{(k-p)}(t) + Qy_0(t),$$

where $\underline{x}_{*(p)}$ and $\overline{x}_*^{(k-p)}$ are the subvectors of the

state vectors $\underline{x}_*$ and $\overline{x}_*$ containing the first p and the last $(k-p)$ components, respectively. Then

$$\underline{e}_z = z - \underline{z} = H_z^+ x_{*(p)} + H_z^- x_*^{(k-p)} + Qy_0$$
$$-(H_z^+ \underline{x}_{*(p)} + H_z^- \overline{x}_*^{(k-p)} + Qy_0)$$
$$= H_z^+ \underline{e}_{*(p)} - H_z^- \overline{e}_*^{(k-p)},$$

and $\underline{e}_z(t) \ge 0$ due to $H_z^+ \ge 0$ and $H_z^- \le 0$. By analogy, it is demonstrated that $\overline{e}_z(t) \ge 0$ for

$$\overline{z}(t) = H_z^+ \overline{x}_{*(p)}(t) + H_z^- \underline{x}_*^{(k-p)}(t) + Qy_0(t).$$

- Let the matrix H_z have several rows: $H_z = \begin{pmatrix} H_z^+ \\ H_z^- \end{pmatrix}$, where H_z^+ and H_z^- are submatrices such that $H_z^+ \ge 0$ and $H_z^- \le 0$. We define

$$\underline{z}(t) = \begin{pmatrix} H_z^+ \underline{x}_*(t) \\ H_z^- \overline{x}_*(t) \end{pmatrix} + Qy_0(t);$$

then, obviously,

$$\underline{e}_z = \begin{pmatrix} H_z^+ \\ H_z^- \end{pmatrix} x_* + Qy_0 - \left(\begin{pmatrix} H_z^+ \underline{x}_* \\ H_z^- \overline{x}_* \end{pmatrix} + Qy_0 \right)$$
$$= \begin{pmatrix} H_z^+ \underline{e}_* \\ -H_z^- \overline{e}_* \end{pmatrix} \ge 0.$$

A more complex case is when H_z has several rows, each of the structure $(H_z^+ \vdots\ H_z^-)$. It reduces to a combination of the two cases considered.

Remark 2. The condition $\underline{x}_*(0) \le x_*(0) \le \overline{x}_*(0)$ is crucial in the theorem. For the positive system (14), it gives $\underline{e}_*(t) \ge 0$ and $\overline{e}_*(t) \ge 0$ for all $t \ge 0$. Due to no feedback in the observer and the stability of the matrix F_*, these inequalities will hold for some $t > 0$ even without the condition $\underline{x}_*(0) \le x_*(0) \le \overline{x}_*(0)$: the initial conditions are "forgotten" for $t \ge k$.

Indeed, let us denote $v_0(t) = |J_*| E_k v_* - J_* v(t) \ge 0$ and consider the first equation in system (14). According to [12], its solution can be represented as

$$\underline{e}_*(t) = F_*^t \underline{e}_*(0) + \sum_{i=0}^{t-1} F_*^{t-i-1} v_0(i). \qquad (17)$$

It is easy to check that $F_*^k = 0$. Then, for $t \ge k$, the value $\underline{e}_*(t)$ will be determined by the second term on the right-hand side of equality (17). By construction, this term is nonnegative, so $\underline{e}_*(t) \ge 0$ for all $t \ge k$. Similarly, we can show that $\overline{e}_*(t) \ge 0$ for all $t \ge k$.

4. ROBUST SOLUTION

If condition (10) does not hold for all $k < n$, we find a robust solution minimizing the contribution of the disturbance to the model. It is almost identical to the solution proposed in [1], except for minimizing the norm $\|(\Phi_1 \vdots\ -J_1 \vdots\ \ldots \vdots\ -J_k) L^{(k)}\|_F$ subject to the condition

$$(\Phi_1 \vdots\ -J_1 \vdots\ -J_2 \vdots\ \ldots \vdots\ -J_k) V^{(k)} = 0. \quad (18)$$

In other words, the matrix R_* from the paper [1] is replaced by Φ_1. We can say that the problem is to determine a solution $(\Phi_1 \vdots\ -J_1 \vdots\ \ldots \vdots\ -J_k)$ with "maximal orthogonality" to the columns of the matrix $L^{(k)}$.

Following [1], based on all linearly independent solutions of equation (18), for some fixed dimension k we construct the matrix

$$W = \begin{pmatrix} \Phi_1^{(1)} & -J_1^{(1)} & -J_2^{(1)} & \ldots & -J_k^{(1)} \\ & & \cdots & & \\ \Phi_1^{(N)} & -J_1^{(N)} & -J_2^{(N)} & \ldots & -J_k^{(N)} \end{pmatrix}$$

and find the singular value decomposition $WL^{(k)} = U_L \Sigma_L V_L$. We choose the first transposed column of the matrix U_L as the vector of weight coefficients $w = (w_1, \ldots, w_N)$ and let $(\Phi_1 \vdots\ -J_1 \vdots\ \ldots \vdots\ -J_k) = wW$. Finally, we determine the rows of the matrix Φ from formula (11) and the matrices $G_* = \Phi G$ and $L_* = \Phi L$ to design model (2) with minimal sensitivity to the disturbances.

Due to the additional term $L_* \rho(t)$ in model (2), the dynamics of the interval observer for $v \ne 0$ are corrected as follows:

$$\underline{x}_*^+ = F_* \underline{x}_* + J_* y + G_* u - |J_*| E_k v_* - |L_*| E_k \rho_*,$$
$$\overline{x}_*^+ = F_* \overline{x}_* + J_* y + G_* u + |J_*| E_k v_* + |L_*| E_k \rho_*.$$

The expressions (14) for the estimation errors are modified appropriately:

$$\underline{e}_*^+ = F_*\underline{e}_* - J_*v+ | J_* | E_k v_* + L_*\rho+ | L_* | E_k\rho_*,$$
$$\bar{e}_*^+ = F_*\bar{e}_* + J_*v+ | J_* | E_k v_* - L_*\rho+ | L_* | E_k\rho_*.$$

Clearly, the desired result is immediate from the proof of the theorem and the obvious additional inequality $| L_* | E_k\rho_* \pm L_*\rho(t) \geq 0$.

5. INTERVAL ESTIMATION OF THE FULL STATE VECTOR

In several cases, this interval estimation approach for the variable $z(t) = Mx(t)$ can be applied for the full state vector $x(t)$ as follows. Without loss of generality, assume that the matrix H has the maximal rank and

$$H = (H_0 \quad 0),\ y(t) = H_0 x^{(1)}(t) + v(t),$$
$$x(t) = \begin{pmatrix} x^{(1)}(t) \\ x^{(2)}(t) \end{pmatrix},$$

where H_0 is a nonsingular matrix. Let us define

$$\underline{y}(t) = y(t) - E_l v_*,\ \ \bar{y}(t) = y(t) + E_l v_*,$$
$$\underline{x}^{(1)}(t) = H_0^{-1}\underline{y}(t),\ \ \bar{x}^{(1)}(t) = H_0^{-1}\bar{y}(t). \tag{19}$$

Then

$$\underline{e}^{(1)}(t) = x^{(1)}(t) - \underline{x}^{(1)}(t)$$
$$= H_0^{-1}(y(t) - v(t)) - H_0^{-1}\underline{y}(t) = H_0^{-1}(E_l v_* - v(t)),$$

$$\bar{e}^{(1)}(t) = \bar{x}^{(1)}(t) - x^{(1)}(t)$$
$$= H_0^{-1}\bar{y}(t) - H_0^{-1}(y(t) - v(t)) = H_0^{-1}(E_l v_* + v(t)).$$

Under the assumption $H_0^{-1} \geq 0$, from $E_l v_* \pm v(t) \geq 0$ we obtain $\underline{e}^{(1)}(t) \geq 0$ and $\bar{e}^{(1)}(t) \geq 0$ and, consequently, $\underline{x}^{(1)}(t) \leq x^{(1)}(t) \leq \bar{x}^{(1)}(t)$. Thus, the variable $x^{(1)}(t)$ given the condition $H_0^{-1} \geq 0$ is estimated based on the expression (19), and the disturbance $\rho(t)$ has no effect on this estimate.

Remark 3. The condition $H_0^{-1} \geq 0$ obviously holds in application-relevant cases when the components of the vector $x^{(1)}(t)$ are measured by separate sensors and $H_0 = H_0^{-1} = I_l$.

The variable $x^{(2)}(t)$ can be assigned an interval estimate using the observer (12). Assuming $z(t) = x^{(2)}(t) = M^{(2)}x(t)$ for some matrix $M^{(2)}$ and using the criterion (7) with the matrix $M^{(2)}$ instead of M, we check the possibility of designing an observer insensitive to the disturbance. Then, depending on the check results, we design an observer of the form (12) or the robust one.

Remark 4. The condition $\underline{x}_*(0) \leq x_*(0) \leq \bar{x}_*(0)$ of the theorem follows from $\underline{x}(0) \leq x(0) \leq \bar{x}(0)$ only when $\Phi \geq 0$. Indeed, for $\Phi \geq 0$ and $x(0) - \underline{x}(0) \geq 0$, we obtain $\Phi(x(0) - \underline{x}(0)) \geq 0$ and, consequently, $x_*(0) = \Phi x(0) \geq \Phi\underline{x}(0) = \underline{x}_*(0)$. The inequality $x_*(0) \leq \bar{x}_*(0)$ is established by analogy. According to Remark 1, however, this is not critical since the relation $\underline{z}(t) \leq z(t) \leq \bar{z}(t)$ will necessarily hold for $t \geq k$.

6. AN ILLUSTRATIVE EXAMPLE

We consider a discretized model of an electric drive

$$x_1(t+1) = k_1 x_2(t) + x_1(t),$$
$$x_2(t+1) = k_2 x_3(t) + x_2(t) + \rho(t),$$
$$x_3(t+1) = k_3 x_2(t) + k_4 x_3(t) + k_5 u(t),$$
$$y_1(t) = x_1(t) + v_1(t),\ \ y_2(t) = x_3(t) + v_2(t), \tag{20}$$

with the following notations: the coefficients $k_1,...,k_5$ are some drive parameters depending on the sampling interval; the disturbance $\rho(t)$ is due to an external load torque applied to the motor shaft. The model under consideration is described by the matrices

$$F = \begin{pmatrix} 1 & k_1 & 0 \\ 0 & 1 & k_2 \\ 0 & k_3 & k_4 \end{pmatrix},\ G = \begin{pmatrix} 0 \\ 0 \\ k_5 \end{pmatrix},$$

$$H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},\ L = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

Choosing $M = (0 \quad 1 \quad 0)$, we design an interval observer for the variable $x_2(t)$. Since the disturbance enters the equation of this variable, the model will be sensitive to it. Therefore, we construct the model by letting $L = 0$. Equation (9) with $L^{(k)} = 0$, $k = 1$, takes the form

$$(\Phi \quad -J_*)\begin{pmatrix}1 & k_1 & 0\\ 0 & 1 & k_2\\ 0 & k_3 & k_4\\ 1 & 0 & 0\\ 0 & 0 & 1\end{pmatrix}=0.$$

The solution is $\Phi=(1/k_1 \quad -1 \quad 0)$ and $J_*=(1/k_1 \quad -k_2)$, yielding $G_*=0$ and $L_*=-1$. As is easily verified, condition (5) holds and $H_z=-1$ and $Q=(1/k_1 \quad 0)$. The desired model has the form

$$x_*(t+1)=(1/k_1)H_1x(t)-k_2H_2x(t)-\rho(t),$$
$$z(t)=-x_*(t)+(1/k_1)y_1(t).$$

Based on this model, considering $H_z=-1$, we design the following interval observer for the variable $z(t)=x_2(t)$:

$$\begin{aligned}\underline{x}_*(t+1)&=(1/k_1)y_1(t)-k_2y_2(t)\\ &-(1/k_1)v_{*1}-k_2v_{*2}-\rho_*,\\ \overline{x}_*(t+1)&=(1/k_1)y_1(t)-k_2y_2(t)\\ &+(1/k_1)v_{*1}+k_2v_{*2}+\rho_*,\\ \underline{z}(t)&=-\overline{x}_*(t)+(1/k_1)y_1(t),\\ \overline{z}(t)&=-\underline{x}_*(t)+(1/k_1)y_1(t).\end{aligned} \tag{21}$$

The variables $x_1(t)$ and $x_3(t)$ can be estimated using the expression (19):

$$\underline{x}_1(t)=y_1(t)-v_{*1},\ \underline{x}_3(t)=y_2(t)-v_{*2},$$
$$\overline{x}_1(t)=y_1(t)+v_{*1},\ \overline{x}_3(t)=y_2(t)+v_{*2}.$$

Comparing these estimates with those from [2] and similar papers, we can conclude the following: the proposed approach provides a simpler observer and smaller intervals because, in particular, the intervals for the variables $x_1(t)$ and $x_3(t)$ do not contain the disturbance $\rho(t)$.

For numerical simulation, we selected system (20) and the observer (21) with $u(t)=0.2\sin(t/100)$ and the noises $v_1(t)$, $v_2(t)$, and $\rho(t)$ described by random processes with a variance of 0.5. For simplicity, we assumed that $k_1=k_2=k_5=1$ and $k_3=k_4=-1$. The simulation results are shown in Figs. 1 and 2, i.e., the variable $x_2(t)$ and its lower and upper bounds $\underline{x}_*(t)$ and $\overline{x}_*(t)$ for the initial conditions $x(0)=0$, $\underline{x}(0)=-0.05$, and $\overline{x}(0)=0.05$ and $x(0)=0$, $\underline{x}(0)=0.05$, and $\overline{x}(0)=-0.05$, respectively. As has been emphasized in Remark 1, the initial conditions affect only the estimates at the initial time instants.

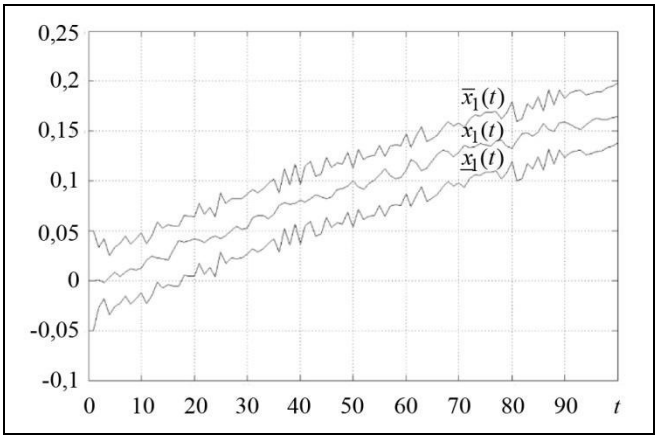


Fig. 1. The variable $x_1(t)$ and its lower $\underline{x}_1(t)$ and upper $\overline{x}_1(t)$ bounds for the initial conditions $x(0) = 0$, $\underline{x}(0) = -0.05$, and $\overline{x}(0) = 0.05$.

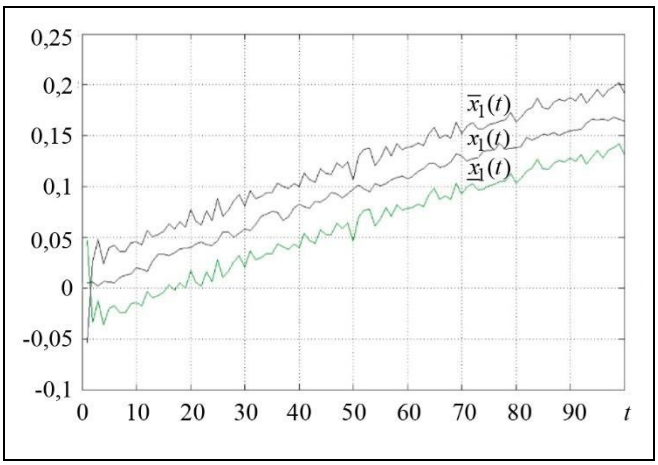


Fig. 2. The variable $x_1(t)$ and its lower $\underline{x}_1(t)$ and upper $\overline{x}_1(t)$ bounds for the initial conditions $x(0) = 0$, $\underline{x}(0) = 0.05$, and $\overline{x}(0) = -0.05$.

CONCLUSIONS

In this paper, we have designed interval observers for linear dynamic systems described by discrete-time models under exogenous disturbances and measurement noise. The relations based on the identification canonical form have been obtained to design a minimal-order interval observer estimating the set of admissible values for a given linear function of the system's state vector. A robust approach to solving the design problem has been considered as well. It has been demonstrated that the proposed solution can be used to estimate the full state vector. The theoretical results have been illustrated by a numerical example.

REFERENCES

1. Zhirabok, A., Zuev, A., and Kim, C.I., Method to Design Interval Observers for Linear Time-Invariant Systems, *Journal of Computer and Systems Sciences International*, 2022, vol. 61, no. 5, pp. 485–495.
2. Efimov, D. and Raissi, T., Interval Observer Design for Dynamic Systems with Uncertainties, *Automation and Remote Control*, 2016, vol. 77, no. 2, pp. 191–225.
3. Khan, A., Xie, W, Zhang, L., and Liu, L., Design and Applications of Interval Observers for Uncertain Dynamical Systems, *IET Circuits Devices Syst.*, 2020, vol. 14, pp. 721–740.
4. Kolesov, N., Gruzlikov, A., and Lukoyanov, E., Using Fuzzy Interacting Observers for Fault Diagnosis in Systems with Parametric Uncertainty, *Proc. of 12th International Symposium "Intelligent Systems" (INTELS'16)*, Moscow, Russia, 2016, pp. 499–504.
5. Kremlev, A. and Chebotarev, S., Synthesis of Interval Observer for Linear System with Variable Parameters, *Journal of Instrument Engineering*, 2013, vol. 56, no. 4, pp. 42–46. (In Russian.)
6. Chebotarev, S., Efimov, D., Raissi, T., and Zolghadri, A., Interval Observers for Continuous-Time LPV Systems with L_1/L_2 Performance, *Automatica*, 2015, vol. 51, pp. 82–89.
7. Mazenc, F. and Bernard, O., Asymptotically Stable Interval Observers for Planar Systems with Complex Poles, *IEEE Trans. Automatic Control*, 2010, vol. 55, no. 2, pp. 523–527.
8. Blesa, J., Puig, V., and Bolea, Y., Fault Detection Using Interval LPV Models in an Open-Flow Canal, *Control Engineering Practice*, 2010, vol. 18, pp. 460–470.
9. Zheng, G., Efimov, D., and Perruquetti, W., Interval State Estimation for Uncertain Nonlinear Systems, *Proc. IFAC Nolcos*, Toulouse, 2013.
10. Zhang, K., Jiang, B., Yan, X., and Edwards, C., Interval Sliding Mode Based Fault Accommodation for Non-minimal Phase LPV Systems with Online Control Application, *Intern. J. Control*, 2020, vol. 93, no. 11, pp. 2675–2689. DOI: 10.1080/00207179.2019.1687932.
11. Zhirabok, A., Zuev, A., and Shumsky, A.E., Methods of Diagnosis in Linear Systems Based on Sliding Mode Observers, *Journal of Computer and Systems Sciences International*, 2019, vol. 58, no. 6, pp. 898–914.
12. Kwarernaak, C. and Sivan, R., *Linear Optimal Control Systems*, New York: Wiley, 1972.

This paper was recommended for publication by S.A. Krasnova, a member of the Editorial Board.

Received January 22, 2023, and revised March 14, 2023. Accepted March 14, 2023.

Author information

Zhirabok, Aleksei Nilovich. Dr. Sci. (Eng.), Far Eastern Federal University, Vladivostok, Russia; Institute of Marine Technology Problems, Russian Academy of Sciences, Far Eastern Branch, Vladivostok, Russia
✉ zhirabok@mail.ru

Zuev, Aleksandr Valer'evich. Cand. Sci. (Eng.), Institute of Marine Technology Problems, Russian Academy of Sciences, Far Eastern Branch, Vladivostok, Russia; Far Eastern Federal University, Vladivostok, Russia
✉ alvzuev@yandex.ru

Kim, Chung Il. Postgraduate, Far Eastern Federal University, Vladivostok, Russia
✉ kim.ci@dvfu.ru

Cite this paper
Zhirabok, A.N., Zuev, A.V., and Kim, C.I., Interval Observer Design for Discrete Linear Time-Invariant Systems with Uncertainties. *Control Sciences* **2**, 15–22 (2023). http://doi.org/10.25728/cs.2023.2.2

Original Russian Text © Zhirabok, A.N., Zuev, A.V., Kim, C.I., 2023, published in *Problemy Upravleniya*, 2023, no. 2, pp. 19–27.

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com

DOI: http://doi.org/10.25728/cs.2023.2.3

TERMINAL CONTROL OF MOVING OBJECTS IN THE CLASSES OF PIECEWISE CONSTANT AND PIECEWISE CONTINUOUS FUNCTIONS

V.K. Zavadsky[1], V.P. Ivanov[2], E.B. Kablova[3], L.G. Klenovaya[4], and V.Yu. Rutkovskii

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

[1–4]✉ vladguc@ipu.ru

Abstract. This paper presents a terminal control problem with the separation of object's state coordinates into two types: the slowly changing coordinates figuring in boundary conditions and the coordinates of the stabilization loop. A predictive model of the object is introduced to design the control action. A differential equation is derived for predicted mismatches in the boundary conditions. The original system is discretized in time based on this equation. This problem is solved step-by-step in the classes of piecewise constant and piecewise continuous control actions. As an illustrative example, the problem of controlling the fuel consumption of a stage of a liquid-propellant launch vehicle is considered. The class of control actions is extended from piecewise constant to piecewise continuous functions in order to cover additional requirements for the control process. The continuous functions on intervals between control jumps are chosen using the local boundary conditions obtained during the terminal control design in the class of piecewise constant functions.

Keywords: terminal control, model predictive control (MPC), fuel consumption control for launch vehicles.

INTRODUCTION

Terminal control problems arise in many areas of engineering. In rocket dynamics, some examples of such problems include insertion into Earth orbit, fuel consumption until complete exhaustion from the tanks, rendezvous of spacecraft, etc. In these examples, the control problem is to bring an object (often called a plant in control theory) to given final states under known initial conditions. Terminal conditions can be defined as the required values of the object's state coordinates or in a more complex form, e.g., as some functions of the state coordinates.

The modern terminal control concept for the objects of rocket and space technology was thoroughly described in the monograph [1]. A fundamental element of terminal control is predicting the object's final state in the form of given boundary conditions.

Predictive methods in the field of rocket dynamics were considered in [2–4]. The application of modern model predictive control (MPC) methods to nonlinear systems was discussed in detail in the book [5]. The general idea of the MPC approach consists in constructing a predictive model of the object in order to find the optimal control actions at the current and subsequent time instants. Note that only the current control action is implemented and the optimization procedure is repeated at the next time instant. In the paper [6], for a given control action in a predictive model, the derivative of the predicted coordinate values over time was determined. MPC methods also evolved towards applying real-time optimization [7] and making the closed-loop system robust and adaptive [8, 12]. To reduce the computational burden, the prediction procedure is performed for a limited number of time instants.

In the rocket dynamics problems mentioned, terminal control is formed as part of the general problem of controlling an object by separating its relatively slow physical processes that determine the motion to a given target. In this case, the general control action is decomposed into the terminal control action and the

problem of stabilizing the object with respect to its motion to a given target. One example is angular position stabilization for a launch vehicle with respect to the pitch angle program during its orbital insertion control. Note that the control action is directly applied to the object's dynamic part belonging to the stabilization loop. Terminal control design is usually considered independently of the stabilization loop. The object's coordinates outputted by the stabilization loop are taken as the terminal control action.

The authors [9] studied the terminal control problem with the decomposition of the general problem considering the formal description of the plant's dynamics, including the stabilization loop. Such an approach takes into account the dynamics of the transient response of the stabilization loop to the control action instead of the operating errors of this loop. In this case, the idea of predicting the object's final state is implemented for all dynamic channels of the system: from the application point of control actions to mismatches in the boundary conditions. The described approach was considered in [9–11] for control actions in the class of piecewise constant functions. The paper "On a Terminal Control Problem with Prediction of Mismatches in the Boundary Conditions" by D.D. Tabalin was mentioned in the conference chronicle [11]. This paper solves the terminal control problem in the class of piecewise continuous functions. This problem is closely related to the approach presented in [9]. Therefore, we provide the main control design results in the class of piecewise continuous functions. The class of control actions is extended from piecewise constant to piecewise continuous functions in order to cover additional requirements for the control process. The continuous functions on intervals between control jumps are chosen using the local boundary conditions obtained during the terminal control design in the class of piecewise constant functions.

1. PROBLEM STATEMENT

To understand the essence of control processes in a system bringing an object to a given final state, it is useful to separate two interconnected systems of object's equations differing in transients.

We consider a dynamic system of the form

$$\begin{aligned} \frac{dx_1}{dt} &= f_1(x_1(t), x_2(t), t), \\ \frac{dx_2}{dt} &= f_2(x_2(t), u(t), t), \\ x(t_0) &= x_0, \end{aligned} \qquad (1)$$

where $x_1 \in R^{n_1}$, $x_2 \in R^{n_2}$, $x = (x_1, x_2)$, $x_0 \in X_0 \subset R^{n_1+n_2}$, u denotes the control action, $u \in U \subset R^m$, and $t \in [t_0, T]$.

Here, the first system of equations for the coordinates $x_1(t)$ describes the object's motion to a given target. The objects controlled by terminal systems are very inertial in terms of transients to a given final state. (As a rule, they are integrating links.)

Their transients are controlled through other object's coordinates $x_2(t)$ with rapidly decaying transient dynamics. The essence of such control is to set the values of these coordinates. Control in the traditional sense (i.e., the position of different actuators such as drives, rudders, etc.) stabilizes the object's coordinates with respect to the set values. The operation of the closed stabilization loop is described by a system of equations for the coordinates $x_2(t)$. The control action $u(t)$ on the right-hand side of these equations is the settings for the coordinates $x_2(t)$.

One example is angular position stabilization for a launch vehicle with respect to the pitch angle program during its orbital insertion control. Thus, the terminal control action is directly applied to the dynamic part of the object belonging to the stabilization loop. In mathematical formulations of optimal control problems, terminal control design is usually considered independently of the stabilization loop.

Assume that there exists a unique solution of system (1) under any initial conditions.

The terminal control problem is to transfer the system to a desired final state x that satisfies the following boundary conditions at a time instant T:

$$\psi(x_1) : \psi_i(x_{1i}(T)) = 0,\ i \in L \subset \overline{1,\ell},$$

where ℓ is the number of these conditions. The boundary conditions are imposed only on the coordinates x_1, representing given condition vectors for individual components x_{1i}. Let the function ψ be differentiable. Note that in some terminal control problems, boundary conditions may be imposed on part of the coordinates x_2.

The time instant $T > t_0$ is either fixed or determined by the first instant of satisfying the pth boundary condition $\psi_p(x_{1p}) = 0$.

The object's coordinates x_2 are outputted by the stabilizing loop of the control system. In this case, the operation of the loop is considered only in terms of the transient response to changing the control action. Assume that the transients terminate on an interval significantly smaller than the terminal control horizon. This approach somewhat restricts generality since the

dynamics of the coordinates x_2 and x_1 are supposed mutually independent.

With system (1) we associate a predictive model of the form

$$\frac{d\hat{x}_1}{d\tau} = f_1(\hat{x}_1(\tau), \hat{x}_2(\tau), \tau), \tau \in [t, T],$$
$$\frac{d\hat{x}_2}{d\tau} = 0, \quad (2)$$
$$\hat{x}(t) = x(t).$$

The function

$$z(t) \equiv \psi(\hat{x}_1(T|t)),$$

where the time instant T is calculated for system (2) by analogy with the original system (1), will be called the predicted mismatch in the boundary conditions due to the predictive model (2). Therefore, the time instant T for system (2) is either fixed or determined by the first instant of satisfying the boundary condition ψ_P.

We denote by $\hat{x}_1(T|t)$ the value of $\hat{x}_1(T)$ due to the system of equations (2) that is predicted at a time instant t.

Being a function of $x(t)$ and t, $z(t)$ satisfies the differential equation

$$\frac{dz(t)}{dt} = \frac{\partial\psi}{\partial\hat{x}_1(T|t)}\left[\frac{\partial\hat{x}_1(T|t)}{\partial x_2(t)} f_2(x_2(t), u(t), t) + \frac{dT}{dt} f_1(\hat{x}_1(T|t), x_2(t), T)\right].$$

For details, see the paper [9].

From this point onwards, let T = const. In this case,

$$\frac{dz(t)}{dt} = \frac{\partial\psi}{\partial\hat{x}_1(T|t)} \frac{\partial\hat{x}_1(T|t)}{\partial x_2(t)} f_2(x_2(t), u(t), t). \quad (3)$$

We introduce the notation dz/dx_2 for a matrix by which the function f_2 is multiplied in the expression for dz/dt. Then formula (3) can be written as

$$\frac{dz}{dt} = \frac{\partial z}{\partial x_2}(x(t), t) f_2(x_2(t), u(t), t). \quad (4)$$

We will solve the terminal control action for object (1) by designing a feedback control action as a function of the predicted mismatches in the boundary conditions described by the differential equation (3). The control action will be chosen in the class of piecewise continuous functions.

The idea consists in a two-step control design. In the first step, the control action is constructed in the class of piecewise constant functions. Local boundary conditions are formed by solving this problem. Fulfilling the local boundary conditions in aggregate allows solving the original terminal control problem. In the second step, we extend the class of control actions: on the intervals between control jumps, the control action is assumed to be a continuous function. Such a control action will be designed considering the local boundary conditions obtained with piecewise constant control.

2. LOCAL BOUNDARY CONDITIONS AT TERMINAL CONTROL JUMPS

Assume that $u(t)$ is a piecewise constant function and t_j, j = 1, 2,..., k – 1, are the time instants of its jumps, $t_k = T$. In this case, $z(t_j)$ satisfies a difference equation, an analog of the differential equation (4). To obtain this equation, we use the results from [9].

Integrating equation (4) on a small interval $[t_j, t_j + \delta t]$ yields

$$z(t_j + \delta t) = z(t_j) + \int_{t_j}^{t_j+\delta t} \frac{\partial z}{\partial x_2}(x(\tau), \tau) f_2(x_2(\tau), u(\tau), \tau) d\tau = z(t_j) + \frac{\partial z}{\partial x_2}(x(t_j), t_j)\Delta x_2 + o(\delta t).$$

Here, δt is the interval of the transient on the coordinate x_2 of the object (1) during a jump of the control action from $u(t_j)$ to $u(t_{j+1})$ at the time instant t_j. For $\tau \in [t_j + \delta t, t_{j+1}]$, $f_2(x_2(\tau), u(\tau), \tau) = 0$.

For small δt, we pass to the discrete system

$$z(t_{j+1}) = z(t_j) + \frac{\partial z}{\partial x_2}(x(t_j), t_j)\Delta x_2(t_j), \quad j = 0, 1, 2,..., k-1, \; z(t_k) = z(T), \quad (5)$$

where

$$\Delta x_2(t_j) = \int_{t_j}^{t_j+\delta t} f_2(x_2(\tau), u(\tau), \tau) d\tau.$$

Let us reformulate the original terminal control problem as follows. We will find a discrete sequence of the increments $\Delta x_2(t_j)$ of the coordinate $x_2(t)$ at the time instants t_j, j = 0, 1, 2,..., k – 1, instead of the control action $u(t)$ in the class of piecewise constant functions.

We consider the case of no restrictions on Δx_2 and solve the terminal control problem

$$\{\Delta x_2(t_j)\} : z(t_k) = 0. \quad (6)$$

Assume that $\frac{\partial z}{\partial x_2}(x(t_j), t_j) = \frac{\partial z}{\partial x_2}(t_j)$ in formula (5).

For the system of equations (5), problem (6) is solved backwards from the time instant t_{k-1}. For example, some condition is introduced for choosing

$\Delta x_2(t_{k-1})$ uniquely for the time instant t_{k-1}; simultaneously, a boundary condition is introduced for all control actions preceding t_{k-1}. These conditions are formulated by specifying linear operators with respect to the mismatches $z(t_k)$ and $z(t_{k-1})$. The resulting system of equations with respect to the mismatch $z(t_k)$ has the unique trivial solution. When passing to t_{k-2}, the solution for $\Delta x_2(t_{k-1})$ is taken into account.

As a result, the original control problem (6) with the right-end boundary conditions is reduced to an equivalent set of local control problems for a finite number of discrete time instants:

$$\begin{gathered}\left\{\Delta x_2(t_{j-1})\right\}: \\ \frac{\partial z^{\mathrm{T}}}{\partial x_2}(t_{j-1})K(t_j)z(t_j)=0 \quad \forall j\in\overline{p,k}, \\ \left\{\Delta x_2(t_1), \Delta x_2(t_2),..., \Delta x_2(t_{j-2}), K(t_{j-1})\right\}: \\ K(t_{j-1})z(t_{j-1})=0.\end{gathered} \tag{7}$$

Note that the first equation in (7) is intended to choose the current control action $\Delta x_2(t_{j-1})$. Multiplication by $\frac{\partial z^{\mathrm{T}}}{\partial x_2}(t_{j-1})$ gives the mismatches, whose number coincides with the dimension of the control vector.

The second equation specifies the local boundary conditions for choosing control actions at the time instants before t_{j-1}.

The matrix $K(t_{j-1})$ is given by the recurrence relation

$$\begin{cases}K(t_k)=\mathrm{E}, \\ K(t_{j-1})=\left(\mathrm{E}-K(t_j)\frac{\partial z}{\partial x_2}\left(\frac{\partial z^{\mathrm{T}}}{\partial x_2}K(t_j)\frac{\partial z}{\partial x_2}\right)^{-1}\right. \\ \left.\times\frac{\partial z^{\mathrm{T}}}{\partial x_2}K(t_j)\right)K(t_j), \ \forall j\in\overline{p,k},\end{cases} \tag{8}$$

where E is an identity matrix of compatible dimensions and

$$\frac{\partial z}{\partial x_2}=\frac{\partial z}{\partial x_2}(t_{j-1}).$$

We denote $z_j(t_j)=K(t_j)z(t_j)$. The vector $z_j(t_j)$ consists of the mismatches $z(t_k)$ under the control actions $\Delta x_2(t_{j-1}), \Delta x_2(t_j),..., \Delta x_2(t_{k-1})$ (7).

Let the values $\Delta x_2(t_{j-1})\ \forall j\in\overline{p,k}$ be obtained from formula (7):

$$\begin{gathered}\Delta x_2(t_{j-1})=-\left(\frac{\partial z^{\mathrm{T}}}{\partial x_2}(t_{j-1})K(t_j)\frac{\partial z}{\partial x_2}(t_{j-1})\right)^{-1} \\ \times\frac{\partial z^{\mathrm{T}}}{\partial x_2}(t_{j-1})K(t_j)z(t_{j-1}) \quad \forall j\in\overline{p,k}.\end{gathered} \tag{9}$$

We determine the corresponding mismatches $z_j(t_j)$. In this case, $z_j(t_j)=z(t_k)$, and the first equation in (7) can be written as the system of linear equations for $z(t_k)$:

$$\begin{gathered}\frac{\partial z^{\mathrm{T}}}{\partial x_2}(t_{j-1})K(t_j)z(t_j) \\ =\frac{\partial z^{\mathrm{T}}}{\partial x_2}(t_{j-1})z(t_k)=0 \quad \forall j\in\overline{p,k}.\end{gathered} \tag{10}$$

According to [9], the system of equations (10) has the unique trivial solution $z(t_k)=0$ if the rank of the matrix $\left[\frac{\partial z}{\partial x_2}(t_{k-1}) \ \frac{\partial z}{\partial x_2}(t_{k-2}) \ ... \ \frac{\partial z}{\partial x_2}(t_{p-1})\right]$ is equal to the dimension of the vector $z(t_k)$. In this case, the control action (9) is a solution of the original terminal control problem.

The control strategy corresponding to (7) imposes constraints on the object's trajectory for the values of $z(t)$ at a finite number of control jump instants. On the intervals between these instants, control actions may vary for the coordinates x_2. These variations can be chosen considering desired performance criteria for the state and control coordinates. Thus, it is possible to extend the class of control actions (functions) in order to optimize nonterminal criteria.

3. TERMINAL CONTROL DESIGN IN THE CLASS OF PIECEWISE CONTINUOUS FUNCTIONS

We formulate the terminal control problem in the class of piecewise continuous functions by supplementing the problem statement from Section 1.

As before, the controlled object is described by equation (1) and the predictive model by equation (2); the control action $u(t)$ has jumps at discrete time instants t_j, $j = 0, 1, 2,..., k-1$. On the intervals $[t_j, t_{j+1}]$, $u(t)$ is a continuous function. The predicted mismatch vector $z(t)$ satisfies the differential equation (4) with $\frac{\partial z}{\partial x_2}(x(t), t)=\frac{\partial z}{\partial x_2}(t)$. We know the solution of the

local control problems (7) in the class of piecewise constant functions: $\Delta x_2(t_{j-1})\ \ \forall j \in \overline{p,k}$ for the difference equation (5). We choose the control action $u(t)$ in the class of piecewise continuous functions under the following conditions on the intervals between time instants $t_{j-1}, t_j\ \ \forall j \in \overline{p,k}$:

$$\frac{\partial z^{\mathrm{T}}}{\partial x_2}(t_{j-1})K(t_j)z(t_j) = 0 \quad \forall j \in \overline{p,k},$$
$$K(t_{j-1})z(t_{j-1}) = 0,$$

where the matrix $K(t_{j-1})$ is given by (8). In this case,

$$z(t_k) = z(t_{k-1}) + \int_{t_{k-1}}^{t_k} \frac{\partial z}{\partial x_2}(\tau) f_2(x_2(\tau), u(\tau), \tau) d\tau \quad (11)$$

for the piecewise continuous function $u(\tau)$, $\tau \in [t_{k-1}, t_k]$.

Assume that

$$\frac{\partial z^{\mathrm{T}}}{\partial x_2}(t_{k-1})z(t_k) = 0. \quad (12)$$

This condition narrows the class of piecewise continuous control actions under consideration. The new class, restricted by condition (12), is defined below.

Let $u(\tau) = \text{const}$, $\tau \in (t_{k-1}, t_k)$. Due to the jump of the function $u(\tau)$ at the time instant t_{k-1}, in this case, we can write

$$z(t_k) = z(t_{k-1}) + \frac{\partial z}{\partial x_2}(t_{k-1})\Delta x_2(t_{k-1}). \quad (13)$$

The value $\Delta x_2(t_{k-1})$ is determined from condition (12) using the expression (9).

Suppose that on the interval $[t_{k-1}, t_k]$, the piecewise continuous function $u(\tau)$ satisfies the relation

$$\int_{t_{k-1}}^{t_k} \frac{\partial z}{\partial x_2}(\tau) f_2(x_2(\tau), u(\tau), \tau) d\tau = \frac{\partial z}{\partial x_2}(t_{k-1})\Delta x_2(t_{k-1}). \quad (14)$$

In this case, (11) is transformed to (13) and, therefore, condition (12) holds.

Let the control actions at the time instants preceding t_{j-1} be chosen so that $K(t_{j-1})z(t_{j-1}) = 0$. (In other words, the boundary conditions specified by the second equation in (7) are satisfied.) Subtracting $K(t_{j-1})z(t_{j-1})$ from the right-hand side of (11) and performing trivial transformations, we obtain

$$z(t_k) = \frac{\partial z}{\partial x_2}(t_{k-1})\left(\frac{\partial z^{\mathrm{T}}}{\partial x_2}(t_{k-1})\frac{\partial z}{\partial x_2}(t_{k-1})\right)^{-1}\frac{\partial z^{\mathrm{T}}}{\partial x_2}(t_{k-1})z(t_{k-1}) + \int_{t_{k-1}}^{t_k} \frac{\partial z}{\partial x_2}(\tau) f_2(x_2(\tau), u(\tau), \tau) d\tau = 0.$$

The first term on the right-hand side of this expression is $-\frac{\partial z}{\partial x_2}(t_{k-1})\Delta x_2(t_{k-1})$; see formula (9) for $j = k$. Due to the relation (14), we finally arrive at $z(t_k) = 0$.

We now proceed to choosing the control action on the interval $[t_{k-2}, t_{k-1}]$. It is necessary to determine the new mismatch vector $z_{k-1}(t_{k-1}) = K(t_{k-1})z(t_{k-1})$. For the vector $z_{k-1}(t_{k-1})$ and the piecewise continuous function $u(\tau)$, $\tau \in [t_{k-2}, t_{k-1}]$, we write

$$z_{k-1}(t_{k-1}) = K(t_{k-1})z(t_{k-1}) = K(t_{k-1}) \times \left(z(t_{k-2}) + \int_{t_{k-2}}^{t_{k-1}} \frac{\partial z}{\partial x_2}(\tau) f_2(x_2(\tau), u(\tau), \tau) d\tau \right). \quad (15)$$

Let the condition

$$\frac{\partial z^{\mathrm{T}}}{\partial x_2}(t_{k-1})z_{k-1}(t_{k-1}) = 0 \quad (16)$$

hold for $u(\tau)$.

The control action on the interval $[t_{k-2}, t_{k-1}]$ is chosen using the same considerations as on the interval $[t_{k-1}, t_k]$. For the constant control action $u(\tau)$, $\tau \in (t_{k-2}, t_{k-1})$, we write

$$z_{k-1}(t_{k-1}) = z_{k-1}(t_{k-2}) + \frac{\partial z_{k-1}}{\partial x_2}(t_{k-2})\Delta x_2(t_{k-2}), \quad (17)$$

where

$$\frac{\partial z_{k-1}}{\partial x_2}(t_{k-2}) = K(t_{k-1})\frac{\partial z}{\partial x_2}(t_{k-2}),$$
$$z_{k-1}(t_{k-2}) = K(t_{k-1})z(t_{k-2}).$$

The value $\Delta x_2(t_{k-2})$ is determined from condition (16) using the expression (9).

Suppose that on the interval $[t_{k-2}, t_{k-1}]$, the piecewise continuous function $u(\tau)$ satisfies the relation

$$K(t_{k-1})\left(\int_{t_{k-2}}^{t_{k-1}} \frac{\partial z}{\partial x_2}(\tau) f_2(x_2(\tau), u(\tau), \tau) d\tau - \frac{\partial z}{\partial x_2}(t_{k-2})\Delta x_2(t_{k-2})\right) = 0.$$

In this case, (15) is transformed to (17) and, therefore, condition (16) holds.

We define the new mismatch vector as $z_{k-2}(t_{k-2}) = K(t_{k-2})z(t_{k-2})$ under the condition $K(t_{k-2})z(t_{k-2}) = 0$, transforming (15) similarly to (11) on the interval $[t_{k-1}, t_k]$.

The procedure can be continued for all time instants preceding t_{k-1}.

As a result, for the time instant t_j, we have

$$z_j(t_j) = K(t_j)z(t_j) = K(t_j) \times \left(z(t_{j-1}) + \int_{t_{j-1}}^{t_j} \frac{\partial z}{\partial x_2}(\tau) f_2(x_2(\tau), u(\tau), \tau) d\tau \right), \quad (18)$$

$$K(t_j)\left(\int_{t_{j-1}}^{t_j} \frac{\partial z}{\partial x_2}(\tau) f_2(x_2(\tau), u(\tau), \tau) d\tau - \frac{\partial z}{\partial x_2}(t_{j-1})\Delta x_2(t_{j-1}) \right) = 0. \quad (19)$$

Here $K(t_{j-1})$ is given by (8) and $\Delta x_2(t_{j-1})$ by (9). Due to formulas (19) and (9),

$$\frac{\partial z^{\mathrm{T}}}{\partial x_2}(t_{j-1})K(t_j)z(t_j) = 0 \quad \forall j \in \overline{p,k}. \quad (20)$$

Thus, equation (19) and the expression (9) for $\Delta x_2(t_{j-1})$ define conditions equivalent to the initial condition (20).

Letting $K(t_j)z(t_j) = 0$, we employ simple transformations of (18) to show that $z_j(t_j) = K(t_j)z(t_j) = 0$.

Thus, the control action chosen by (19), (9) in the class of piecewise continuous functions solves the terminal control problem: $z(t_k) = 0$. Note that on the intervals between jumps, the control action can be chosen in a sufficiently wide class of functions.

4. AN EXAMPLE

As an example, we consider the problem of controlling the fuel consumption of a liquid-propellant rocket. We restrict the further analysis to the problem of synchronizing the depletion of oxidizer and propellant by the time of turning the rocket stage's power unit off. This problem will be studied in the linear approximation. Due to the nonsynchronous depletion, the unused remainders of propellant components remain in the tanks, reducing the power characteristics of the rocket. The synchronization process is controlled by changing the ratio of component consumption in the power unit. In turn, a deviation of this parameter from the nominal optimal value causes losses of specific thrust. The losses become most tangible when approaching the end of the flight. All these considerations lead to qualitatively formulated requirements for the control process and determine the type of boundary conditions.

The controlled object is described by the equations

$$\dot{x}_1(t) = \frac{1}{T}x_2(t),\ \dot{x}_2(t) = -k(x_2(t) - u(t)),$$
$$t \in [t_0, t_k],\ T = t_k - t_0,\ x_2(t_0) = 0.$$

Here, $x_1(t)$ is the mismatch between the mass fractions of the propellant components and $x_2(t)$ is the relative deviation of the ratio of component consumption from the nominal value.

The boundary conditions are given by

$$x_1(t_k) = 0,\ x_2(t_k) = 0.$$

We determine the vector of the predicted mismatches $z(t)$ and $\dot{z}(t)$:

$$z(t) = \begin{vmatrix} x_1(t) + x_2(t)\dfrac{t_k - t}{T} \\ x_2(t) \end{vmatrix},$$

$$\dot{z}(t) = \begin{vmatrix} x_2(t)\dfrac{t_k - t}{T} \\ 1 \end{vmatrix} \dot{x}_2(t).$$

First, let us solve the problem in the class of piecewise constant functions. In this case, it suffices to have two jumps of the control action $u(t)$ at time instants t_0, $t_0 < t_1 < T$. We determine the vector of predicted mismatches in the boundary conditions at the time instant t_1:

$$z(t_1) = \begin{vmatrix} x_1(t_1) + t_1' x_2(t_1) \\ x_2(t_1) \end{vmatrix},$$

$$t' = \frac{1}{T}(t_k - t),\ t_1' = \frac{1}{T}(t_k - t_1).$$

Note that under jumps of the function $u(t)$, the transient for the coordinate $x_2(t)$ terminates in a time significantly smaller than T. (The transient time is $\partial t < 0.01T$.) To reduce the system error due to the finite transient time ∂t, the partial derivative $\dfrac{\partial z}{\partial x_2}(t)$ in the example is taken at an intermediate time instant on the interval ∂t.

We have

$$z(t_k) = z(t_1) + \begin{vmatrix} t_1^* \\ 1 \end{vmatrix} \Delta x_2(t_1),$$

$$t_1^* \in (t_1', t_1' - \frac{\partial t}{T}),$$

where $\Delta x_2(t_1) = x_2(t_k) - x_2(t_1)$.

Under the jump of the function $u(t)$ at the time instant t_1, the condition $\begin{vmatrix} t_1^* \\ 1 \end{vmatrix} z(t_k) = 0$ implies

$$\Delta x_2(t_1) = -\frac{(x_1(t_1) + t_1' x_2(t_1))t_1^* + x_2(t_1)}{1 + t_1^{*2}}.$$

The matrix $K(t_1)$ takes the form

$$K(t_1)=\begin{vmatrix} 1 & -t_1^* \\ -t_1^* & t_1^{*2} \end{vmatrix}\frac{1}{1+t_1^{*2}}.$$

The new mismatch vector for choosing the control actions $\Delta x_2(t_0)=x_2(t_1)-x_2(t_0)$ is written in the form $z_1(t_1)=K(t_1)z(t_1)$,

where $z(t_1)=z(t_0)+\begin{vmatrix} t_0^* \\ 1 \end{vmatrix}\Delta x_2(t_0),\ t_0^*\in\left(t_0',\ t_0'-\frac{\partial t}{T}\right)$.

After trivial transformations we obtain the following condition for determining the control action $\Delta x_2(t_0)$:

$$z_1(t_1)=\begin{vmatrix} 1 \\ -t_1^* \end{vmatrix}(x_1(t_0)+(t_0^*-t_1^*)\Delta x_2(t_0))\frac{1}{1+t_1^{*2}}=0.$$

Hence, $\Delta x_2(t_0)=-\frac{x_1(t_0)}{t_0^*-t_1^*}$.

Note the possibility $z(t_1)\neq 0, x_1(t_1)\neq 0$.

We now solve the terminal control problem in the class of piecewise continuous functions. On the interval $[t_1, t_k]$ the control action $u(\tau)$ must satisfy the condition

$$\int_{t_1}^{t_k}\begin{vmatrix} \frac{t_k-\tau}{T} \\ 1 \end{vmatrix}\dot{x}_2(\tau)d\tau=\begin{vmatrix} t_1^* \\ 1 \end{vmatrix}\Delta x_2(t_1).$$

What is important, the function $u(\tau)$ can have a jump at the time instant t_1.

When choosing the continuous control function $u(\tau)$, $\tau\in[t_0, t_1]$, we must satisfy the condition

$$K(t_1)\int_{t_0}^{t_1}\left(\frac{t_k-\tau}{T}\right)\dot{x}_2(\tau)d\tau=K(t_1)\begin{vmatrix} t_0^* \\ 1 \end{vmatrix}\Delta x_2(t_0).$$

After trivial transformations, it takes the form

$$\int_{t_0}^{t_1}\left(\frac{t_k-\tau}{T}-t_1^*\right)\dot{x}_2(\tau)d\tau=(t_0^*-t_1^*)\Delta x_2(t_0).$$

Using integration by parts for the left-hand side, we obtain

$$x_1(t_0)+\frac{1}{T}\int_{t_0}^{t_1}x_2(\tau)d\tau=-x_2(t_1)(t'_1-t_1^*).$$

This expression can be written as the boundary condition $x_1(t_1)=-x_2(t_1)(t'_1-t_1^*)$.

Under this condition, continuous control actions can be chosen in a sufficiently wide class of functions. Considering the requirement for $x_2(t)$ (the deviation of the ratio of component consumption), $x_2(\tau)$ and the control action $u(\tau)$ on the interval $\tau\in[t_0, t_1]$ can be chosen as descending exponential functions.

Letting $t_0=0$, we set $u(\tau)=Be^{-r\tau}$, $\tau\in[t_0, t_1]$. In this case, $x_2(\tau)=\frac{kB}{k-r}(-e^{-k\tau}+e^{-r\tau})$, where $k>>r$. The parameters B and r are determined based on the initial and final conditions $x_1(t_0)$ and $x_2(t_0)$: $x_1(t_1)=-x_2(t_1)(t'_1-t_1^*)$. On the interval $\tau\in[t_1, t_k]$, we can take $u(\tau)=0$. In this case, $x_2(\tau)=0$ on the interval $\tau\in[t_1+\partial t, t_k]$. We integrate the equation for $\dot{x}_1(t)$ on the interval $[t_1, t_1+\delta t]$ to find $x_1(t_k)=x_1(t_1)+\frac{1}{kT}x_2(t_1)$. In view of the expression for $x_1(t_1)$, we obtain $x_1(t_k)=-x_2(t_1)(t'_1-t_1^*-\frac{1}{kT})$.

CONCLUSIONS

This paper has considered a terminal control problem in two statements: in the classes of piecewise constant and piecewise continuous functions. As has been shown, these statements are interconnected, and it is reasonable to consider them step-by-step.

During the design of piecewise constant control actions (the first step), local conditions are obtained for choosing control actions on each interval between control jumps. Fulfilling the local boundary conditions allows solving the original terminal control problem.

In the second step, control actions in the class of piecewise continuous functions are designed using the piecewise constant control actions constructed earlier. The local conditions yielded by the first step are used as boundary conditions for choosing control actions in the class of piecewise continuous functions on the intervals between control jumps. Note that under the local conditions, the continuous control actions can be chosen in a sufficiently wide class of functions.

REFERENCES

1. Sikharulidze, Yu.G., *Ballistika i navedenie letatel'nykh apparatov* (Ballistics and Guidance of Aircraft), Moscow: Binom. Laboratoriya Znanii, 2011. (In Russian.)
2. Haeussermann, W., Description and Performance of the Saturn Launch Vehicle's Navigation, Guidance, and Control System, *Proc. 3rd Int. IFAC Conf. on Automatic Control in Space*, Toulouse, France, 1970, vol. 3, pp. 275–312.
3. Petrov, B.N., Portnov-Sokolov, Yu.P., Andrienko, A.Ya., and Ivanov, V.P., *Bortovye terminal'nye sistemy upravleniya* (Onboard Terminal Control Systems), Moscow: Mashinostroenie, 1983. (In Russian.)
4. Veremei, E.I. and Eremeev, V.V., Introduction to Prediction-Based Control Problems, *Trudy Vserossiiskoi nauchnoi konferentsii "Proektirovanie nauchnykh i inzhenernykh prilozhenii v srede MATLAB"* (Proceedings of the All-Russian Scientific Conference "Development of Scientific and Engineering Applications in MATLAB"), Moscow, 2004, pp. 98–115. (In Russian.)

5. Grüne, L. and Pannek, J., *Nonlinear Model Predictive Control. Theory and Algorithms*, Springer, 2011.
6. Gul'ko, F.B. and Novosel'tseva, Zh.A., Application of Prediction Methods in the Design Problems of Automatic Control Systems, *Trudy VIII Vsesoyuznogo soveshchaniya po problemam upravleniya* (Proceedings of the 8th All-Union Meeting on Control Problems), Moscow, 1980, vol. 1, pp. 32–34. (In Russian.)
7. Klaučo, M., Kalúz, M., and Kvasnica, M., Real-time Implementation of an Explicit MPC based Reference Governor for Control of a Magnetic Levitation System, *Control Engineering Practice*, 2017, vol. 60, no. 3, pp. 99–105.
8. Langson, W., Chryssochoos, I., Rakovic, S.V., and Mayne, D.Q., Robust Model Predictive Control Using Tubes, *Automatica*, 2004, vol. 40, no. 1, pp. 125–133.
9. Ivanov, V.P. and Tabalin, D.D., On a Deterministic Terminal Control Method with Predictive Forecasting of Mismatches in the Boundary Conditions, *Automation and Remote Control*, 2022, vol. 83, no. 1, pp. 62–77.
10. Tabalin, D.D., Deterministic Design of Terminal Control Algorithms with Prediction of Mismatches in the Boundary Conditions, *Tezisy mezhdunarodnogo molodezhnogo nauchnogo foruma "LOMONOSOV-2020"* (Abstracts of the International Youth Scientific Forum "LOMONOSOV-2020"), Moscow, 2020. (In Russian.)
11. Seminar on Problems of Nonlinear Dynamics and Control in Moscow State University, *Diff. Uravnen.*, 2020, vol. 56, no. 8, pp. 1134–1144. (In Russian.)
12. Chai, J., Medagoda, E., and Kayacan, E., Adaptive and Efficient Model Predictive Control for Booster Reentry, *Journal of Guidance, Control, and Dynamics*, 2020, vol. 43(12), pp. 2372–2382. DOI: https://doi.org/10.2514/1.G004925.

This paper was recommended for publication by B.V. Pavlov, a member of the Editorial Board.

Received October 11, 2022, and revised January 16, 2023. Accepted February 14, 2023.

Author information

Zavadsky, Vladimir Konstantinovich. Cand. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ vladguc@ipu.ru

Ivanov, Vladimir Petrovich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ vladguc@ipu.ru

Kablova, Elena Borisovna. Researcher, Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ vladguc@ipu.ru

Klenovaya, Lyudmila Grigor'evna. Researcher, Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ vladguc@ipu.ru

Rutkovskii, Vladislav Yul'evich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

Cite this paper

Zavadsky, V.K., Ivanov, V.P., Kablova, E.B., Klenovaya, L.G., and Rutkovskii, V.Yu., Terminal Control of Moving Objects in the Class of Piecewise Constant and Piecewise Continuous Functions. *Control Sciences* **2**, 23–30 (2023). http://doi.org/10.25728/cs.2023.2.3

Original Russian Text © Zavadsky, V.K., Ivanov, V.P., Kablova, E.B., Klenovaya, L.G., Rutkovskii, V.Yu., 2023, published in *Problemy Upravleniya*, 2023, no. 2, pp. 28–36.

Translated into English by *Alexander Yu. Mazurov*, Cand. Sci. (Phys.–Math.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com

DOI: http://doi.org/10.25728/cs.2023.2.4

MODELS OF JOINT DYNAMICS OF OPINIONS AND ACTIONS IN ONLINE SOCIAL NETWORKS. PART I: Primary Data Analysis[1]

D.A. Gubanov and D.A. Novikov

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

✉ dmitry.a.g@gmail.com, ✉ novikov@ipu.ru

Abstract. Based on *VKontakte* data, we study the influence of various factors on the dynamics of opinions and actions both at the macro level ("public opinion") and at the micro level (the opinions and actions of individual agents). Primary analysis results are presented for the dynamics of opinions and actions of agents in this social network. In particular, the growing polarization of opinions at the macro level is detected; changes in the opinions of agents over time are observed; socio-demographic characteristics of agents who changed their opinions are determined; a good consistency between the opinions and actions of agents is revealed; finally, an explicit relationship between the opinions and actions of agents is established.

Keywords: social network, agent, opinion, action, social influence, cognitive dissonance, trust in information.

INTRODUCTION

Since the 1950s, researchers have been developing mathematical models of opinion dynamics to explain changes in the beliefs of individuals (*agents*) under the influence of socio-psychological factors; for example, we refer to the publications [1–12] on the subject. In parallel, the same effects have been studied in social psychology; see [13–15], etc.

These investigations are still topical today, particularly due to the rapid development of online social media, where information processes significantly influence the political, economic, and social life of society. For example, under uncertainty and no knowledge, the inaccurate information about the measures to combat the COVID-19 pandemic, which was once disseminated by reputable but often incompetent social networkers, caused a destructive information agenda and undermined the effectiveness of pandemic control efforts through changing the beliefs of network users [16]. At the same time, mathematical models of opinion and action dynamics can be used to predict changes in public beliefs and develop necessary strategies to protect public health. However, the identification of such models is a complex interdisciplinary task.

In this paper, the basic model is the mathematical model of the joint dynamics of opinions and actions of the agents proposed in [17]. As an "empirical base" we adopt the posts, comments, and likes in *VKontakte*, a popular online social network, on wearing medical masks that appeared from March 2020 to February 2021 inclusive. *An opinion* is conventionally interpreted as the "tone" of an agent's comment, as assessed by an automatic classifier; *an action* is conventionally[2]

[1] This research was partially supported by the Russian Science Foundation, project no. 23-21-00408 (D.A. Gubanov).

[2] Of course, commenting and liking are inherently actions. (Within the actional approach [1, 3], different types of actions and relations between them underlie modeling and the analysis of information processes in a network.) Therefore, the separation of opinions (comments) and actions (likes) has an obvious alternative, i.e., the introduction of hidden variables (opinions) and their identification by observable "actions" (comments and likes) within hidden Markov models, Bayesian networks, etc. Such approaches seem promising and the corresponding models will be considered in part III of the study.

interpreted as the tone of a comment with an agent's like.

This multi-part study attempts to answer the following questions:

1) How consistent are the opinions and actions of agents with each other?

2) Do agents change their opinions and actions over time?

3) Who are these (opinion- and action-changing) agents? Do they differ from others in their sociodemographic characteristics?

4) Which models better describe the dynamics of the opinions and actions of agents (linear, threshold, etc.)?

5) Are the influence of actions on opinions (*cognitive dissonance*) and the converse effect significant?

6) Under which factors do the opinions and actions of agents change? Among such factors, we consider:

- the agent's previous opinions or (and) actions;
- social influence:

  – *public opinion* (the averaged shares of certain opinions and actions of the entire social network, i.e., the so-called *macro model*, where the network is conventionally treated as one agent);

  – the opinions or (and) actions of the agent's environment (the agents with the friendship relation to a given agent), i.e., the averaged and (or) individual ones (the so-called *micro model*);
- some unobservable (*latent*) characteristics of the agent.

7) Does an agent's change in the opinion (action) depend on his trust in the source of information? Does it depend on the content of that information?

Question no. 6 has the highest complexity: to answer, we need to analyze all combinations of explanatory variables and order the models with a fixed number of variables by the maximum reduction of the prediction error of the explained variable.

In part I, we examine the dynamics of real opinions and actions of agents concerning their attitude toward wearing medical masks in *VKontakte* as an example. The remainder of this paper is organized as follows. Section 1 describes the initial data. In Section 2, we propose an approach to identifying agents' opinions in the network based on deep learning methods. Section 3 characterizes the dynamics of opinions and actions at the macro level (how much support individual users and the entire online community have for wearing medical masks, how much public opinion changes over time, etc.) as well as the features of information interaction between agents. Finally, the resulting relationship between the opinions and actions of social network agents is presented and analyzed in Section 4. Thus, the paper provides answers to Questions nos. 1–3. Parts II and III of the study will deal with the identification of macro and micro models of the joint dynamics of opinions and actions to answer Questions nos. 4–7.

1. ANALYSIS OF NETWORK INTERACTIONS: INITIAL DATA AND KEY FACTORS

The objects of the media landscape under consideration are information sources and users of *VKontakte*. Information sources publish news covering various aspects of the COVID-19 pandemic and influence network users. Network users (hereinafter referred to as "agents") respond to the messages of information sources and perform actions in the network according to their interests and opinions (comment and like), interacting with each other.

The data were collected for the information sources selected by experts based on the Medialogy's rating; see https://www.mlg.ru/. All sources have *VKontakte* pages and publish news on topics of public importance: *RIA Novosti* (2.9 million subscribers), *RT News* (1.3 million subscribers), *Komsomol'skaya Pravda* (1.1 million subscribers), *RBC* (0.9 million subscribers), *TSARGRAD TV* (0.7 million subscribers), *Moscow 24* (0.5 million subscribers), *Yekaterinburg News E1RU* (0.3 million subscribers), *Snob* (0.3 million subscribers), *Fontanka.ru* (0.3 million subscribers), *Gazeta.ru* (0.2 million subscribers), and *Interfax* (0.1 million subscribers).

We considered and analyzed the posts of these sources on COVID-19 (over 60 thousand posts) and the network response to them for the period from March 1, 2020, to March 1, 2021, (over 2 million comments to the posts and over 7 million likes to the posts and comments). A detailed description of the data collection approach can be found in the papers [18, 19], including some analysis results of network user activity.[3] Then the comments and likes directly related to wearing medical masks were identified.

In view of the initial data, we formalize the descriptive factors of the online social network needed to analyze and identify the models of the joint dynamics of opinions and actions. According to [1, 3], let the network participants be *agents* from a set $N = \{1, 2, ..., n\}$. They commit some *acts*[4] from a fixed set

[3] The data were collected within project 20-04-60296 supported by the Russian Foundation for Basic Research. We are grateful to E.V. Belyavskii and I.V. Kozitsin for their data collection efforts.

[4] The term "action" used in [3] is replaced here by "act" to avoid confusion with actions in models of the joint dynamics of opinions and actions.

$K = \{1, 2, ..., k\}$ at certain time instants t of an interval T. Our considerations are restricted to the following types of acts ($K = \{1, 2\}$):

– publishing a comment on a post or another comment,

– liking a comment.

We denote by Δ the set of acts.[5] Each act $a \in \Delta$ is described by three parameters: the agent who committed it, the type of the act, and the time instant when it was committed. We introduce the following functions to characterize acts:

- f_a: $\Delta \to N$, associating with each act $a \in \Delta$ the agent $i \in N$ who committed it;
- f_t: $\Delta \to T$, associating with each act $a \in \Delta$ the time instant $t \in T$ when it was committed;
- f_k: $\Delta \to K$, associating with each act $a \in \Delta$ its type $j \in K$.

On the set of acts, we define a binary partial-order relation of the form "a causes b": $a \to b$. If $a \to b$, $a \neq b$, and there does not exist $c \in \Delta$ such that $a \to c$ and $c \to b$, then a is *the direct cause of* b: $a \downarrow b$. The binary relation $a \to b$ is supposed to hold in the following cases:

- a is a comment and b is a like to it.
- a is a comment and b is a comment on it.
- a and b coincide.

For each agent $i \in N$, we define the set of all his acts $\delta_i = \{a \in \Delta \mid f_a(a) = i\}$ and the set of his friends $N_i \subseteq N$. (The formal "friendship" relation in an online social network implies that an agent can receive information about the comments posted by his friends, the likes they give, etc.).

Opinions and actions. When modeling the joint dynamics of opinions and actions, we conventionally interpret the agent's *opinion* as his attitude to wearing medical masks, expressed in a comment.

The agent's opinion in a comment $b \in \Delta$ ($f_k(b) = 1$) is formally defined in three ways as follows:

- $r' \in \{0, 1, 2\}$, where the classification results 0, 1, and 2 correspond to "against masks" (or "–"), "for masks" (or "+"), and "neutral/irrelevant" (or "="). This result is determined using the stochastic vector $(p_-, p_+, p_=)$ calculated by the classifier. In machine learning, the components of such a vector are interpreted as the probabilities of the object's belonging to appropriate classes.
- $r'' = \dfrac{p_+}{p_+ + p_-} \in [0, 1]$, the confidence that the comment reflects the "for masks" opinion. Note that $r' = 0$ or $r' = 1$ for this comment.
- $r = \dfrac{p_+ - p_-}{p_+ + p_-} \in [-1, 1]$, where $r = 1$ and $r = -1$ indicate strong confidence in expressing the "for masks" opinion (the "against masks" opinion, respectively). Note that either $r' = 0$ or $r' = 1$ for this comment.

Let a like to some comment be *an action* as well; its assessment coincides with that of the corresponding comment liked: $y' \in \{0, 1, 2\}$, $y'' \in [0,1]$, and $y \in [-1,1]$. For example, for a like $a \in \Delta$, $y'(a) = r'(b)$, where b is the corresponding comment liked (i.e., $b \downarrow a$). To simplify further notations, we adopt the conventions $r'(a) = y'(a)$, $r''(a) = y''(a)$, and $r(a) = y(a)$. Assume that the instant of liking coincides with the instant of publishing the corresponding comment liked.

The agent's position is an aggregate characteristic of the agent that reflects his attitude to wearing medical masks. We consider only the position s_i of agent i for which there exists $a \in \delta_i$, $r'(a) \in \{0, 1\}$, such that

$$s_i = \begin{cases} 0, & \bar{r}_i \le 0.5 - \epsilon, \\ 1, & \bar{r}_i \ge 0.5 + \epsilon, \\ 2, & |\bar{r}_i - 0.5| < \epsilon, \end{cases}$$

where $\bar{r}_i = \dfrac{\sum\limits_{a \in \delta_i \mid r'(a) \in \{0, 1\}} r'(a)}{|\{a \in \delta_i \mid r'(a) \in \{0, 1\}\}|} \in [0, 1]$ is the average of his opinions and the actions "for" and "against." In what follows, we choose $\epsilon = 0.05$.

2. IDENTIFICATION OF AGENTS' OPINIONS

The opinions of agents were determined based on their comments on COVID-19 posts containing the keywords "*mask*," "*muzzle*," and their derivatives (about 60 thousand comments).

Such comments were subjected to preliminary automatic text processing, particularly to remove references to the interlocutor and Internet addresses. Part of the collected sample (approximately 10 thousand comments) was labeled by experts: each comment was given an appropriate-class label reflecting the attitude to the masks: "0" ("against"), "1" ("for"), or "2"

[5] The set of relevant comments on wearing medical masks (see Section 2) and their likes.

("neutral"). Here are labeled examples (the original spelling and punctuation are preserved): "*Well we see the stats and so keep wearing masks and all that stuff*" (for masks), "*I go without a mask. I am a COVID dissident*" (against masks), "*He speaks funnily about masks in Russia*" (neutral/irrelevant).

To solve the classification problem, we developed a neural network classifier based on the pre-trained BERT language model (Conversational RuBERT) [20]. In addition to the BERT layer, its architecture includes additional fully connected layers, dropout layers, and a softmax layer. Note the socio-psychological studies of social network users during the COVID-19 pandemic [21–24], which are close to this problem. However, first, we identified the opinions of network users instead of, e.g., emotions or hate speech and, second, solved the problem for a large target data sample.

The labeled sample was subjected to transformations. After its random shuffle, the training (90%), validation (5%), and test (5%) samples were formed. The classes were balanced using weights to compensate the volume of a certain opinion class: the more examples were contained in a class, the smaller weight the class examples had in the loss function minimized by training. Then we found the hyperparameters of the classifier with the maximum quality value on the validation sample under the resource constraints: 192 tokens as the maximum input sequence length, 16 examples as the size of the training packet, and 7 training epochs. As a result, *the quality value of the trained classifier (accuracy) on the test sample was 0.82.* (For each class, the value of the measure F_1 was not less than 0.7.) For comparison, the baseline classifier (the logistic regression) showed an accuracy of 0.6 on the test sample after finding the optimal values of hyperparameters on the validation sample.

The trained classifier was applied to the entire dataset of mask-relevant comments.

3. STUDY OF THE OPINIONS AND ACTIONS OF AGENTS

This section is organized as follows. Subsection 3.1 presents the characteristics of social network agents with a position on the masks (the socio-demographic characteristics of agents as well as the characteristics of their opinions and actions). Subsection 3.2 is devoted to the dynamics of opinions and actions *at the macro level* (the shares of agents with a certain opinion, without analyzing the opinions of individual agents) and the connection with exogenous factors and trends. Subsection 3.3 considers the structure of information interaction between agents in the network: the characteristics of this structure, the information interaction preferences of agents, and the existence of isolated information communities.

3.1 The characteristics of agents

This subsection considers agents with a position (i.e., $s_i \in \{0, 1, 2\}$ for agent i) who expressed at least one opinion. The classifier determined a total of 16 thousand such agents who expressed their opinions in 50 thousand comments (including 38 thousand "for" or "against" opinions).

For the entire period under examination, the share of agents with the "for" position was 56%; "against," 37%; "neutral," 7%. However, only half of the agents (8 thousand) provided information in social network profiles (i.e., their profiles were not closed or deleted). For these agents, the proportions slightly changed to wearing masks: 58% "for," 35% "against," and 7% "neutral." *Socio-demographic indicators* (gender, age, country, and city) were analyzed for the agents with accessible profiles.

Gender and age of agents. The distribution of agents by gender is shown in Fig. 1. *Among the agents with the "against" position, there was a high share of males* (61%) *compared to the proportion of males among the agents with the "for" position* (51%).

Information about the day and month of birth was provided by 73% of the agents; the year of birth, by only 46% of the agents. The distribution of agents by age is shown in Fig. 2. (The age was defined as of March 1, 2020. The age category was determined based on the theory of generations [25]: the average life expectancy is 80 years, and it consists of four periods of about 20 years: childhood → youth → middle age → old age.)

For the agents with the "against" position, the share of older age groups was higher (19.2% in category 41–60 and 7.2% in category 60+) *than for the agents with the "for" position* (15.2% in category 41–60 and 6.9% in category 60+).

The geographical location of agents. The majority of agents were from Russia (79%), and Ukraine and Belarus completed the top three countries. (15% of agents did not specify the country.) Approximately the same distribution was observed for agents with a fixed position.

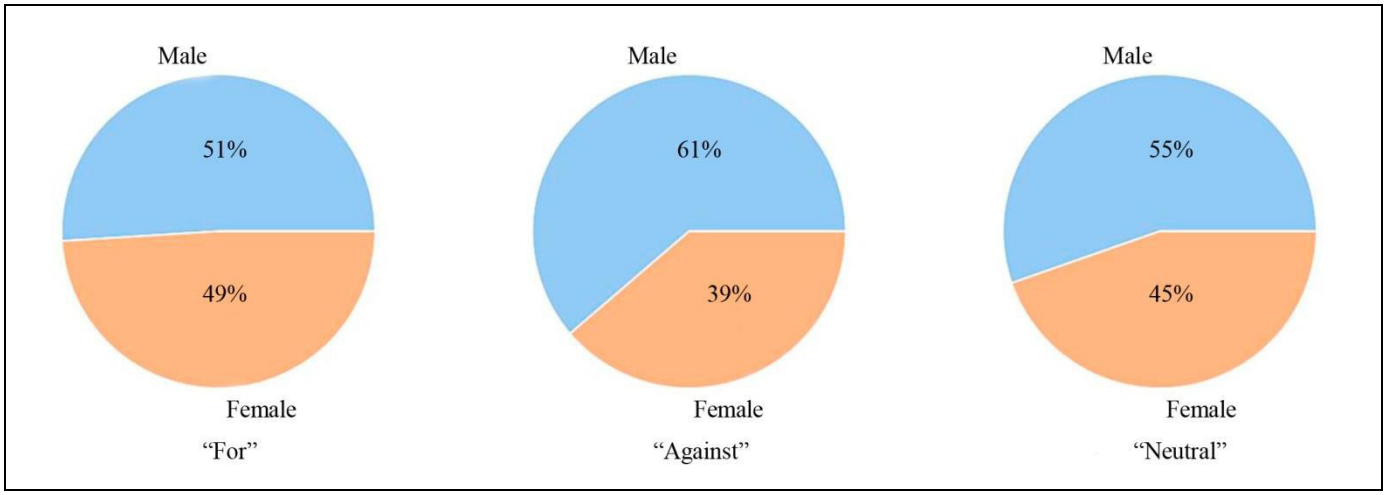


Fig. 1. The distribution of agents with a given position by gender.

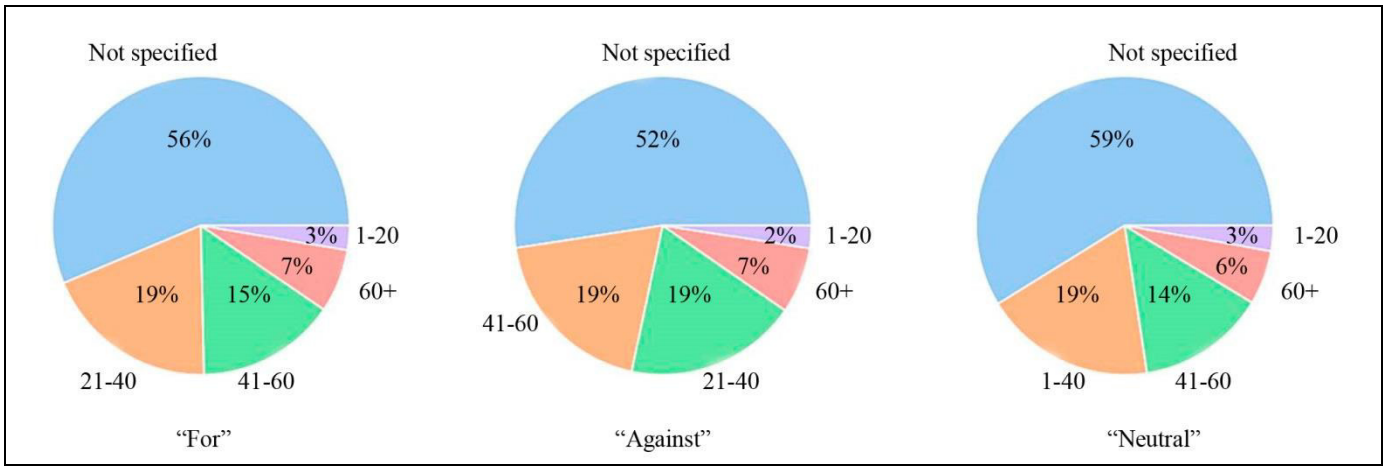


Fig. 2. The distribution of agents with a given position by age category.

The city of residence was specified for 75% of agents; see the distribution in Fig. 3.

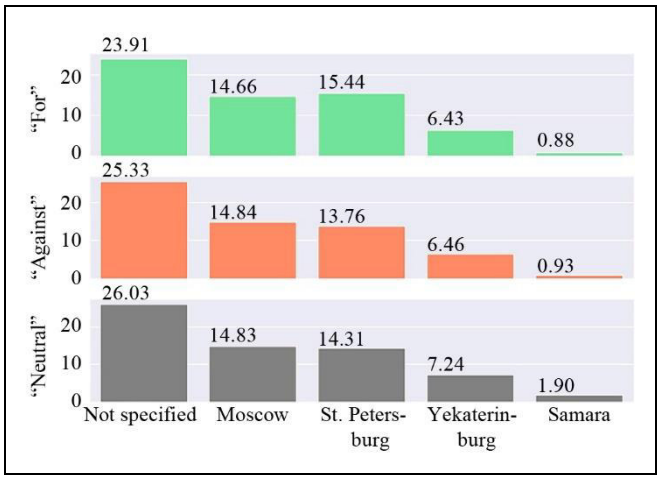


Fig. 3. The distribution of agents with a given position by city.

According to this figure, the first three cities are Moscow, St. Petersburg, and Yekaterinburg. *Among the agents with the "for" position, there were more residents of St. Petersburg; among those with the "against" position, there were more residents of Moscow.*

The opinions and actions of agents. From the study of socio-demographic characteristics, we proceed to the direct analysis of the opinions and actions of the agents with a position.

The distribution of agents by the number of "for" and "against" opinions is shown in Fig. 4. The sample contained *14.4 thousand agents* with a position. On average, an agent with the "for" or "against" opinion made 1.1 comments for wearing masks and 1.5 comments against them during the period under consideration. In other words, *the activity in expressing opinions was low*: *79% of agents commented "for" or "against" at most twice.*

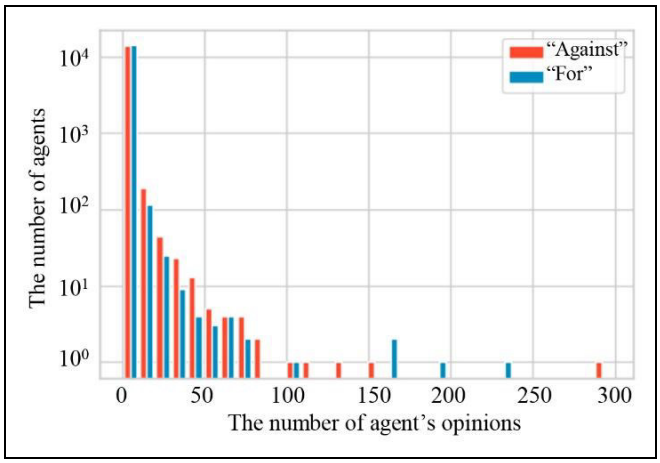


Fig. 4. The distribution of agents by the number of opinions expressed.

About 125 thousand likes (84.5 thousand "against" and 40.8 thousand "for") were given to comments with the "for" or "against" opinion by *44 thousand agents.* (About a third of the likes were given by 6 thousand agents who expressed the "for" or "against" opinion in the comments.) On average, such an agent left 1.9 "against" likes and 0.9 "for" likes. Thus, *likers were not very active as well* (Fig. 5): *79% of the agents performed no more than two "for" or "against" actions.*

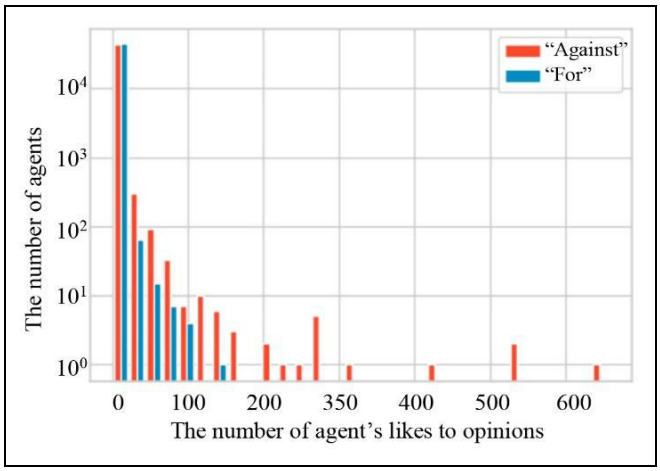


Fig. 5. The distribution of agents by the number of actions.

Of interest is the time interval between successive expressions of the agents' opinions in the comments or the "probability"[6] of expressing an opinion again within a certain period (Fig. 6). As it turned out, if an agent expressed an opinion again, he did so with the following probabilities: 0.5 within a day, 0.7 within a week, and 0.8 within three weeks.

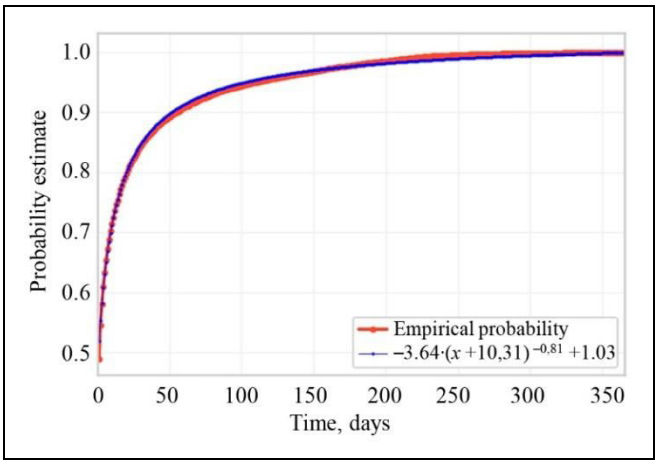


Fig. 6. The estimated probability of expressing an opinion again depending on the time interval. The blue graph is the approximation by the power function.

This result can be easily explained: an agent is involved in discussing new information occasions; as a rule, one occasion is discussed during a day, and the agents involved in the discussion can express their opinions more than once.

3.2 The dynamics of "public" opinion in the network

Let us consider the dynamics of discussions at the macro level. Figure 7 shows the dynamics of the number of agents' opinions on wearing masks (the assessments of comments). The data were smoothened using the 3-day moving average.

On average, the agents posted 42 "for" comments per day (a median of 30, a maximum of 272), 59 "against" comments (a median of 41, a maximum of 419), and 46 neutral/irrelevant comments (a median of 26, a maximum of 409). *The peaks of activity come at the moments of restrictions*. In particular, on March 25, 2020, President Vladimir Putin addressed Russians and announced the introduction of the first off-work period due to COVID-19; in October 2020, repeated restrictions were introduced in response to the growing incidence (e.g., access to entertainment venues was restricted on October 19). Of course, another explanation is possible: *agents' activity was connected with the objective picture of COVID-19 incidence in the Russian Federation.* To test such a hypothesis, we analyzed the incidence dynamics in the Russian Federation based on the *Johns Hopkins University* data [26]; see the graph in Fig. 8.

[6] Here the "probability" is the share of cases falling in a selected time interval (i.e., an estimated probability).

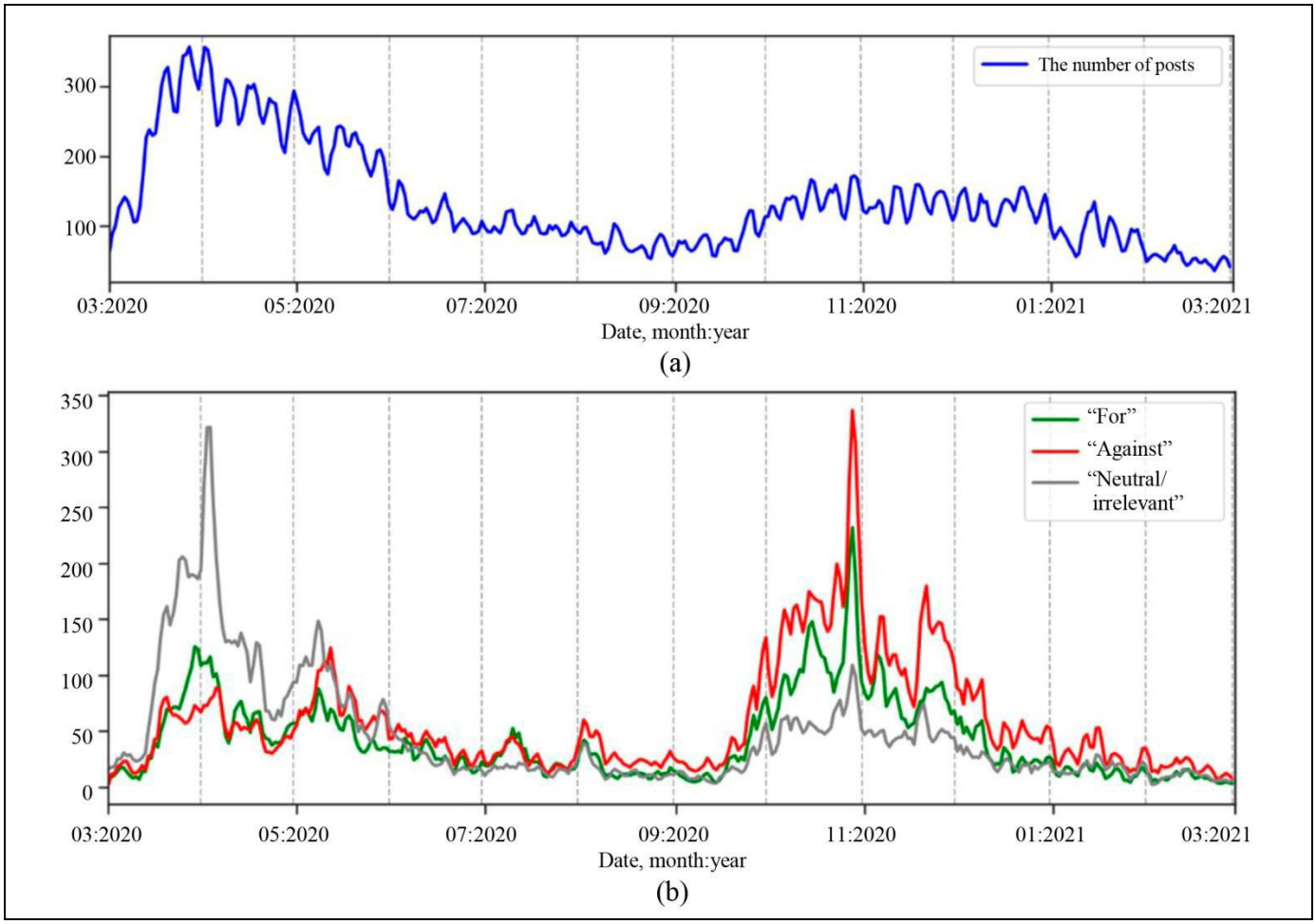


Fig. 7. The dynamics of activity on the issue of wearing masks in *VKontakte*: (a) the number of posts[7] and (b) the number of comments.

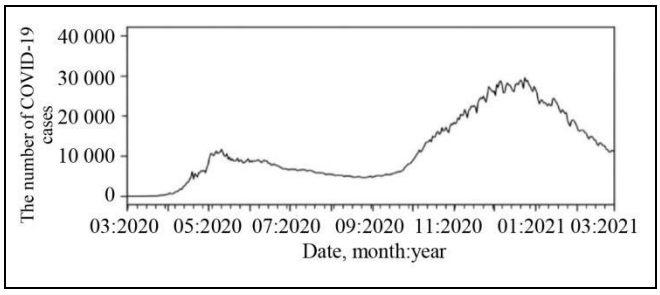


Fig. 8. The dynamics of COVID-19 incidence in the Russian Federation.

On average, there were 11.5 thousand cases per day (a median of 8.8 thousand and a maximum of 29.0 thousand) in the Russian Federation during the period under consideration. Pearson's correlation between incidence and the number of "for" opinions is 0.1 (a maximum of 0.5 is reached at a lag of 45 days, too large for a meaningful explanation); between incidence and the number of "against" opinions, 0.3 (a maximum of 0.7 is achieved at a lag of 45 days); between incidence and the number of "neutral/irrelevant" opinions, –0.3 (–0.2 at a lag of 38 days). Note the correlation between positive and negative (0.9), positive and neutral (0.6), and negative and neutral (0.4) messages. Consequently, *the social network activity on the "mask issue" is most likely indirectly related with COVID-19 incidence. To a higher degree, it is determined by informational events, including the agenda set by public authorities*: e.g., the measures to combat the pandemic.

How did attitudes to wearing masks change over time? As it turned out, *the share of "against" opinions increased* (Fig. 9*a*): by 21% in one year. The share of "against" actions changed even more (Fig. 9*b*): by 23% in one year.

At the same time, the share of "for" and "against" opinions increased in the total number of relevant opinions: it increased by 30% in one year. The share of "for" and "against" actions also increased (by 36%). In other words, we observe *the growing polarization in the network*.

[7] According the random check results, the posts of information sources were neutral on the issue of wearing masks.

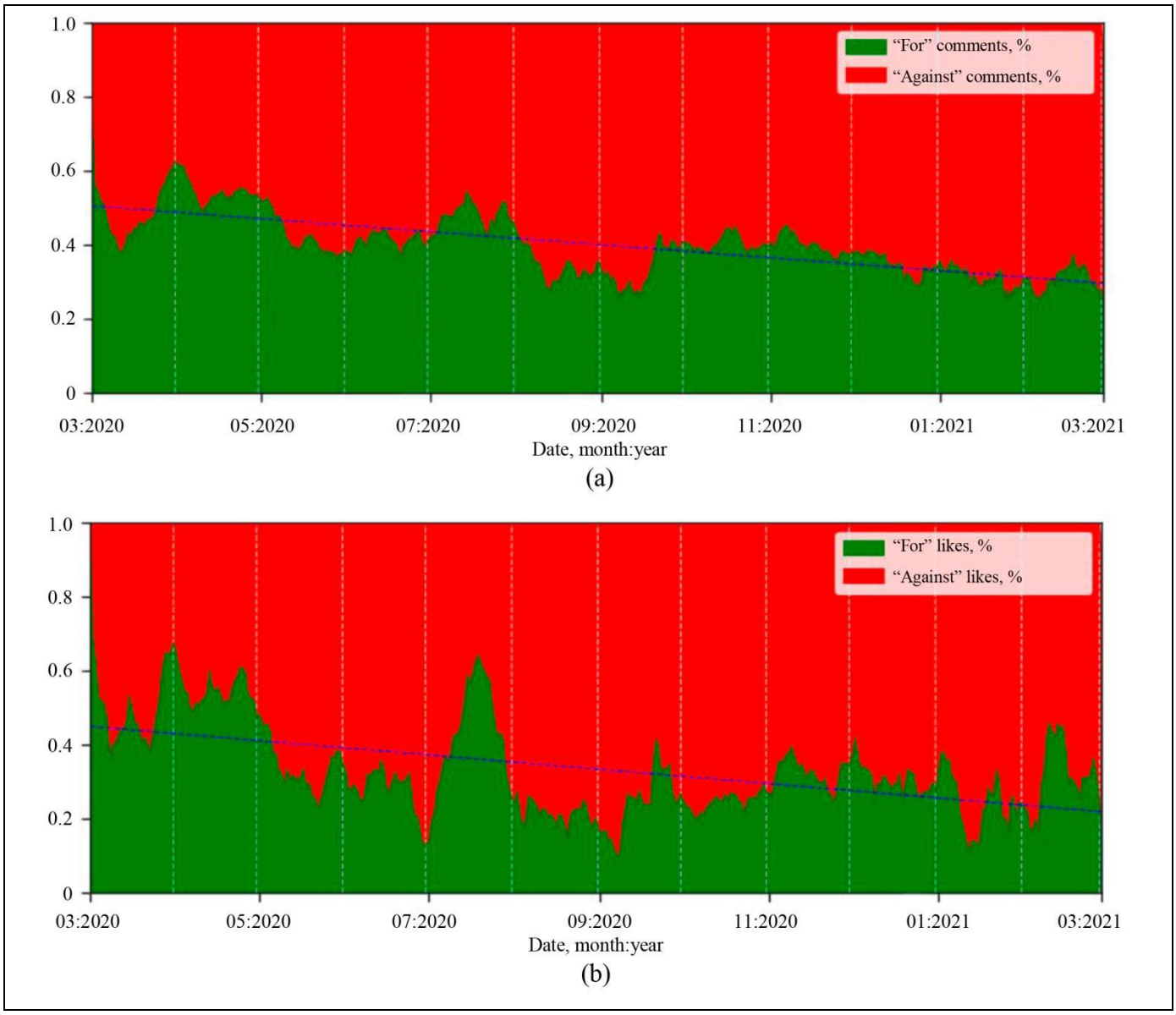


Fig. 9. The shares of: (a) "for" opinions (green area) and "against" opinions (red area) and (b) "for" actions and "against" actions.

3.3 The information interaction of agents

For the agents who responded to the posts of information sources, we construct the following networks of information interactions:

- G, the comments–likes network;
- G_C, the comments network;
- G_L, the likes network.

The network G is connected, it consists of 955 thousand nodes and 5216 thousand interaction links. The network G_C has 878 thousand comment links, and the network G_L has 4522 thousand like links.

The distribution of agents of the network G by degrees is shown in Fig. 10. Note the *power-law* nature of the dependence. The slope of the "straight line" differs for the in-degrees (d^-) and out-degrees (d^+): *a considerable number of agents have high "popularity" (the distribution of d^+); at the same time, there are significantly fewer agents with large "activity" (the distribution of d^-).* The densities of degrees are demonstrated in Fig. 10*b* and *c*. Also, the empirical density was approximated by known heavy-tailed distributions (the power law $f(x) \propto x^{-\alpha}$ and the power law with an exponential cutoff $f(x) \propto x^{-\alpha} e^{-\lambda x}$, as the most appropriate ones). The power law, especially with cutoff, describes well the popularity of agents but not their activity.

What are the peculiarities of interaction between agents with different positions on wearing medical

masks? Let us define an agent's position as the average of the opinions expressed in his actions (see the notations in subsection 3.1). According to Fig. 11, *for the most part, agents take polar positions* (even after eliminating the agents who committed a single act with the "for" or "against" opinion).

The natural question arises: *do agents prefer to interact with like-minded persons?* The answer is important for assessing the informational influence of the environment on opinions in the network.

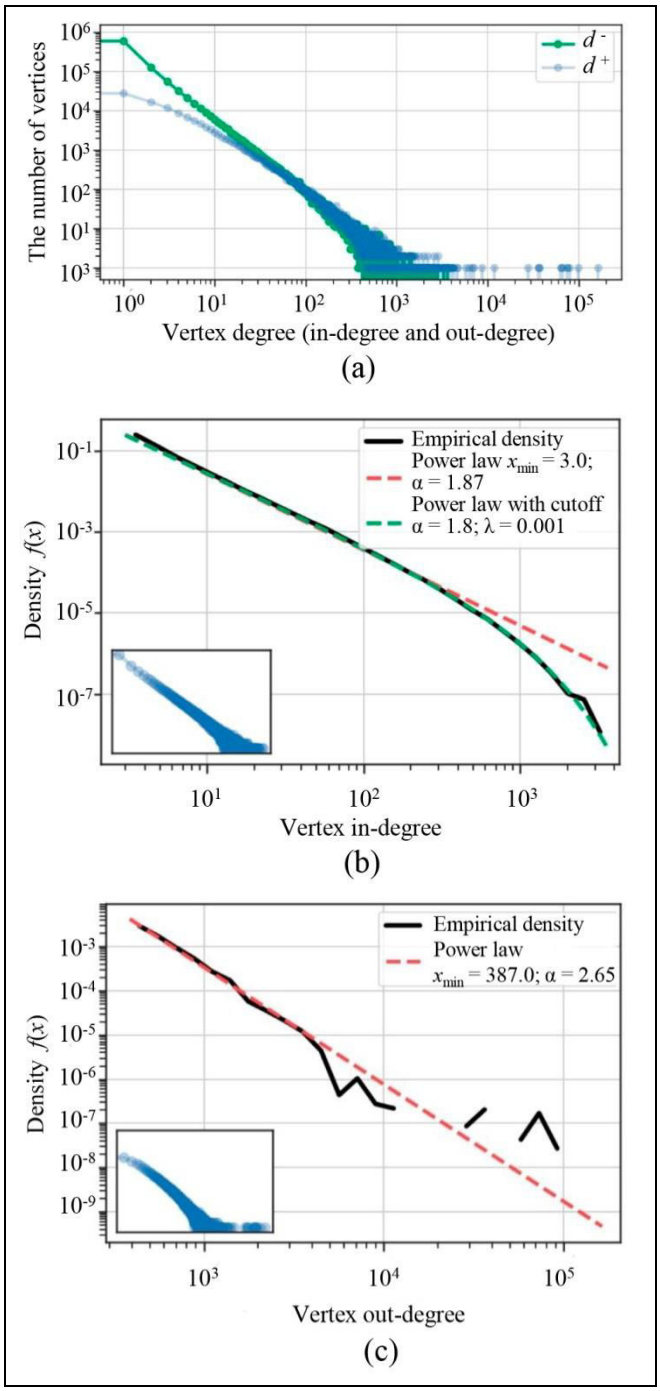


Fig. 10. The distribution of agents in the network: (a) by in-degree d^- and out-degree d^+, (b) by in-degree d^- and (c) by out-degree d^+.

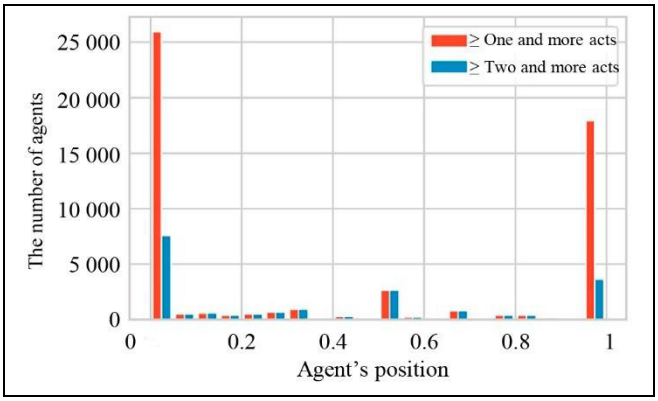


Fig. 11. The distribution of agents by their position.

Table 1 contains the values of *the assortativity coefficient* [27] (its range [–1.0, 1.0]) for the agents who committed at least one "for" or "against" act (52.8 thousand ones) and for the agents who committed at least four "for" or "against" acts (8.5 thousand).

Table 1

The assortativity coefficient for networks

The number of acts	G	G_C	G_L
At least 1	0.21	–0.01	0.25
At least 4	0.24	–0.05	0.31

Therefore, agents (especially active ones) prefer to like agents with a similar position on wearing medical masks. However, such preferences are not pronounced, and there is no particular preference for commenting on agents with a certain position.

Now, we consider the likes network for the agents who committed at least four "for" or "against" acts. In this network, the "for" position is taken by 30% of agents ($p = 0.30$) and the "against" position by 64% of agents ($q = 0.64$). For a randomly chosen edge, the estimated probability that it links agents with different positions is $2pq = 0.38$. At the same time, the share of such edges in the network is 0.24. The inequality $0.24 < 0.38$ confirms the weak assortativity of this network.

Let us calculate probabilities for the "cause-effect" relations in the likes network (Table 2). We introduce the following notations for the events: A_+ (A_-) means that for a randomly chosen link, the agent causing likes has the "for" position (the "against" position, respectively); B_+ (B_-) means that for a randomly chosen link, the agent making likes has the "for" position (the "against" position, respectively).

According to Table 2, the agents with the "against" position prefer to interact with agents with a similar position. (They both influence and are influenced by like-minded persons.) At the same time, for the agents

with the "for" position, the interlocutor's position is not so important.

Table 2

Estimated conditional probabilities for like links

$P(B\|A)$	B_-	B_+
A_-	0.78	0.17
A_+	0.47	0.46

$P(A\|B)$	A_-	A_+
B_-	0.76	0.19
B_+	0.44	0.49

However, we cannot conclude that the agents in the likes network are divided into weakly interacting communities based on their positions: for such a partition, the modularity value [28] is 0.12. The conclusion is confirmed by visualizing the largest connectivity component of this graph; see Fig. 12. For comparison, the partitioning of the network into communities using greedy modularity maximization [29] yields a value of 0.52. (The maximum possible value is 1.0.)

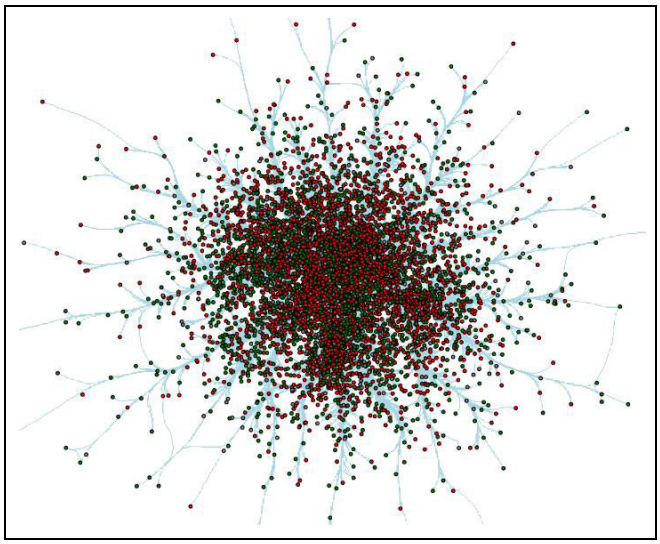


Fig. 12. The graph of likes between agents who committed at least four "for" or "against" acts. (The agents with the "for" position are marked in green whereas those with the "against" position in red.)

Consequently, agents do not form echo chambers (communities of like-minded persons) even when considering only the like links (the highest assortativity coefficient). The agents are influenced by the environment with different positions and can change their position after the influence.

4. THE RELATIONSHIP BETWEEN THE OPINIONS AND ACTIONS OF AGENTS

Let us pose the following questions:

- Are there agents who changed their opinions?
- Does the agent's opinion affect his actions?
- Do the agent's actions affect his opinion?

(See the Introduction.)

These questions are essential to identify the models of opinion/action dynamics. We try to answer them below.

4.1 The agents who changed their opinions

To identify and model the agents who changed their opinions (Question no. 2 in the Introduction), we need to select agents with suitable activity. Consider the agents with the following features:

– Each of them expressed at least one "for" opinion and one "against" opinion on wearing medical masks.

– Each of them expressed his opinion 10 to 100 times. (The weak activity of network agents has been emphasized in subsection 3.1: the majority of agents expressed their opinions and performed actions not more than twice during the period under consideration.)

– Each of them has an open social network profile and at least five friends (required to assess the sociodemographic characteristics of agents and the influence of friends).

These conditions are satisfied for 162 agents (*about* 1% *of the agents with the "for" or "against" opinion*). They will be called *significant agents*. Note that relaxing the second condition does not appreciably increase the number of significant agents (Fig. 13); however, in this case, the data on each agent become insufficient for the purposes of analysis and modeling.

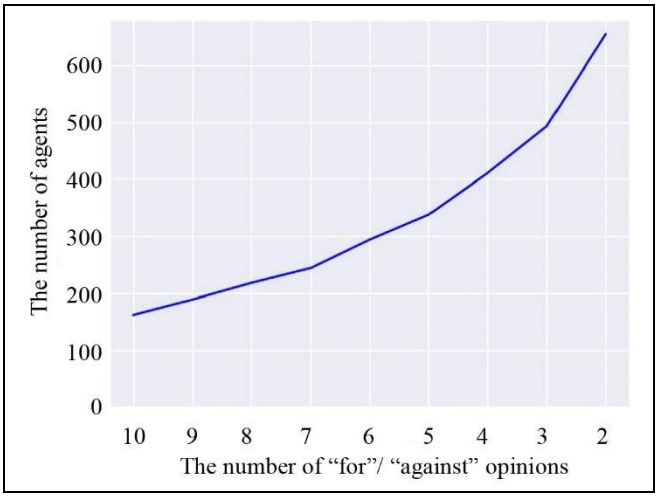


Fig. 13. The distribution of agents by the number of "for" or "against" opinions. (The horizontal line corresponds to the minimum number of opinions for an agent.)

Thus, although there are agents who changed their opinion during the period (some did that twice and more), their share is small. Examples of the opinion

dynamics of significant agents are shown in Fig. 14. (The horizontal axis corresponds to time and the vertical axis to the opinion in the range [–1, 1]; "for" opinions are marked in green and "against" opinions in red.)

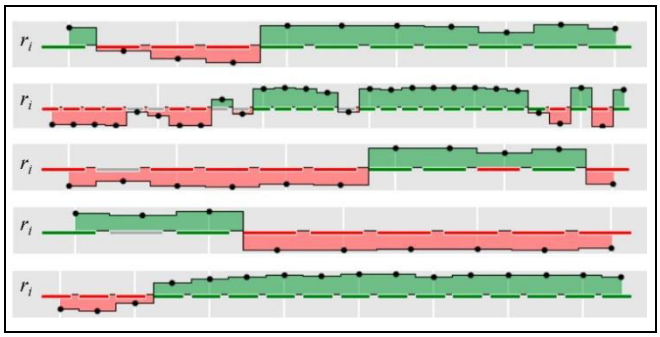


Fig. 14. The opinion dynamics of significant agents: some examples.

We characterize the significant agents: define their socio-demographic characteristics and build networks of links between them (the networks of friendship, comments, and likes).

The socio-demographic characteristics of significant agents. For most of the significant agents (99 agents or 61%), the age was not specified. The same situation is with the initial sample of agents with opinions (the age was not specified for 62% of agents).

The distribution of the other agents by age is presented in Fig. 15. The age of half of the significant agents does not exceed 47 years (38 years for the agents with opinions). The average age is 48 years (42 years for the agents with opinions). Consequently, *the significant agents are older than those with opinions*.

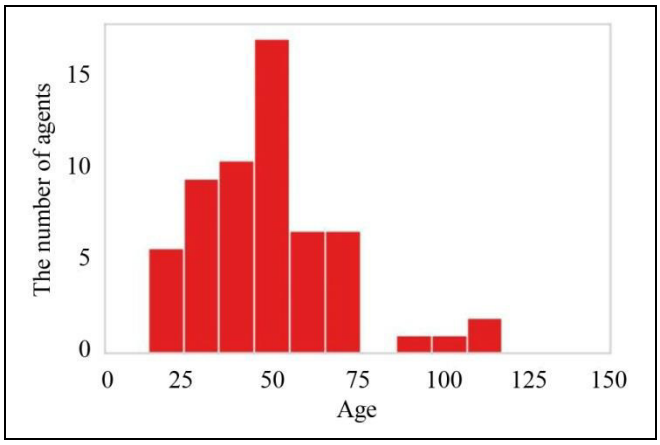


Fig. 15. The distribution of significant agents by age.

According to Fig. 16, the city was not specified for 25% of the significant agents (for 34% of the agents with opinions); 23% of the significant agents specified St. Petersburg (12% of the agents with opinions), 17% of the significant agents specified Moscow (13% of the agents with opinions), and 10% of the agents specified Yekaterinburg (6% of the agents with opinions). Thus, *significant agents prefer to specify the city to a greater extent; for significant agents, the share of their representatives from St. Petersburg and Yekaterinburg is higher.*

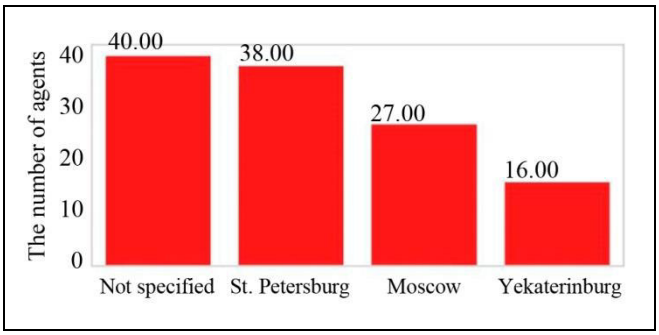


Fig. 16. The distribution of significant agents by the city of residence.

Among the significant agents, the majority belong to males (93 or 57.4%). At the same time, there are 59.6% of males among the agents with opinions. Based on the binomial test results, we do not reject the null hypothesis of equal distributions.

The networks of significant agents. *In the friendship network* of significant agents, there are only 17 links and most of the vertices (138) are isolated (Fig. 17). The green-color vertices correspond to the agents who generally have the "for" opinion and the red-color ones to those with the prevailing "against" opinion.

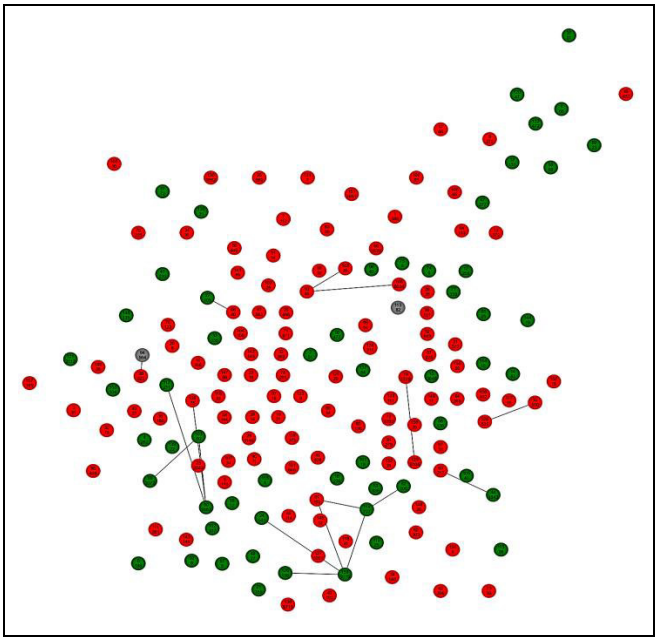


Fig. 17. The friendship network of significant agents.

The assortativity coefficient for the friendship network is 0.32: with some reservation, due to the small number of links, friendship can be assumed an indicator of the similarity of the agents' positions. On average, a significant agent has 432 friends, and half of the significant agents have no more than 113 friends (which is quite close to Dunbar's number).

In the comments network, significant users have 157 friendship links and 45 nodes are isolated (Fig. 18).

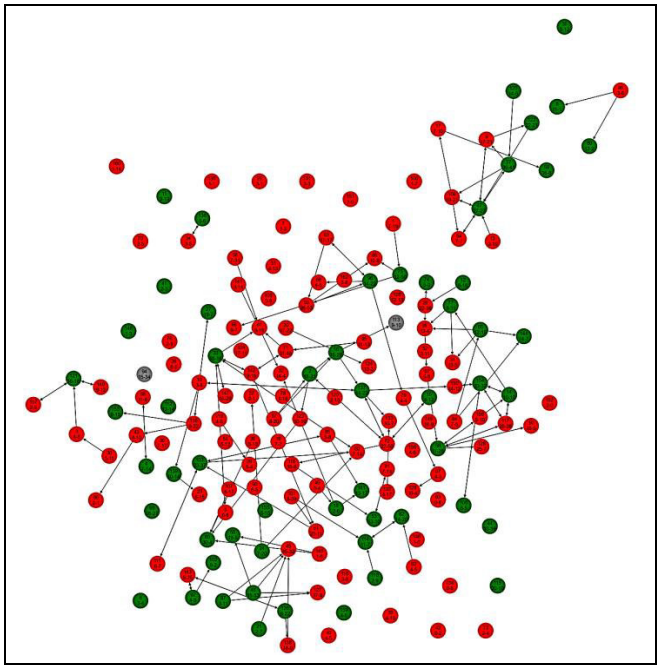


Fig. 18: The comments network of significant agents.[8]

The assortativity coefficient for the network is –0.46: *significant agents with opposite positions prefer to comment on each other.* On average, a significant agent is commented by 10 significant agents and, in turn, he comments on 13 significant agents; every second significant agent is commented by at most 7 significant agents and, in turn, every second significant agent comments on at most 10 significant agents.

In the likes network of significant users, there are 248 friendship links and 37 vertices are isolated (Fig. 19).

The assortativity coefficient for the network is 0.58: *significant agents with a similar position receive likes from each other.* On average, a significant agent receives likes from 69 significant agents and, in turn, he likes 28 significant agents; every second agent receives likes from at most 39 significant agents and, in turn, he likes at most 13 significant agents.

[8] Vertex positions are the same for all networks of significant agents: the friendship network, the comments network, and the likes network.

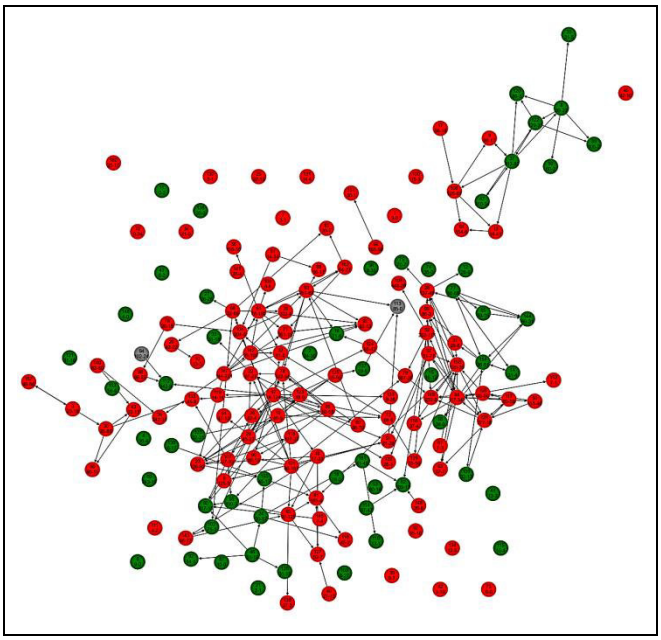


Fig. 19. The likes network of significant agents.

4.2 The influence of opinions on actions

Let us consider the opinion dynamics of agent $i \in N$ with a *position* and consecutive instants $t_m \in T$, $m = \overline{1, M_i}$, of expressing his opinions. (For each m, there exists a comment $a \in \delta_i$ such that $r'(a) \in \{0, 1\}$, $f_k(a) = 1$, $f_t(a) = t_m$.)

We define the set of "for" or "against" actions performed by agent i during the period $\tau = (t_m, t_{m+1}]$ between the expressed opinions with the numbers m and $(m + 1)$:

$$A_i^m = A_i(\tau) = \{a \in \delta_i \mid f_t(a) \in \tau,\ f_k(a) = 2,\ r'(a) \in \{0, 1\}\}.$$

To assess the influence of an agent's opinion on his actions, we introduce the *consistency degree*

$$1 - \frac{1}{M_i} \sum_{m \in \overline{1, M_i}} \left| r_i^{(")m} - \frac{\sum\limits_{a \in A_i^m} r''(a)}{|A_i^m|} \right| \in [0, 1],$$

where $r_i^{(")m}$ is the opinion expressed in a comment a such that

$$r''(a) \in \{0, 1\},\ f_t(a) = t_m,\ f_a(a) = i.$$

In a practical interpretation, *consistency* (see Question no. 1 in the Introduction) reflects how much the agent's actions coincide (correlate) with his opinions.

Figure 20 shows a histogram of the distribution of agents by the consistency degree (for the agents who expressed at least five opinions).

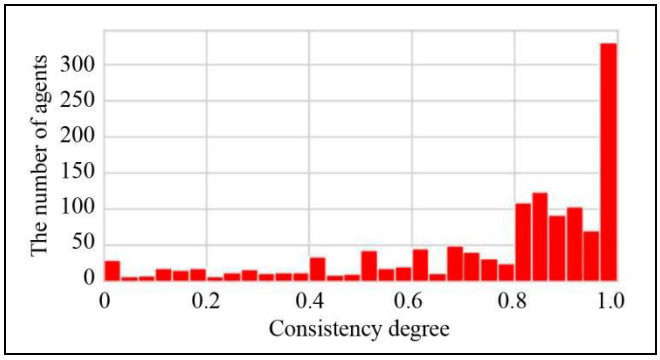


Fig. 20. The distribution of agents by the consistency degree.

The consistency degree averaged over the entire network is 0.76 (when restricting the number of expressed opinions, 0.73). *In general, the agent's actions are "consistent" with his opinion, i.e., the agent's opinion can "influence" his actions.*

4.3 The influence of actions on opinions

Let us assess this influence, thereby partially answering Question no. 5. We define the set of actions performed by agent i during a given period $\tau = [t_m, t_{m+1})$:

$$B_i(\tau) = \{a \in \delta_i \mid r'(a) \in \{0, 1\},\ f_t(a) \in \tau,\ f_k(a) = 2\}.$$

Also, we define the influence of the agent's actions on his opinion:

$$r_{D_i}(\tau) = \frac{\sum\limits_{b \in B_i(\tau)} r(b)}{|B_i(\tau)|} \in [-1, 1].$$

Assume that an opinion change is *significant* if it exceeds the threshold $\epsilon = 0.1$. By analogy with [30], we consider the "probability"[9] of an agent's significant opinion change under the influence of his actions. Let all possible scenarios of expressing the agent's consecutive opinions, $t_m \to t_{m+1}$, be divided into five classes based on his "initial" opinion:

– "strongly against," $r \in [-1, -0.6]$;
– "moderately against," $r \in (-0.6, -0.2]$;
– "weakly expressed position," $r \in (-0.2, 0.2]$;
– "moderately for," $r \in (0.2, 0.6]$;
– "strongly for," $r \in (0.6, 1]$.

For each class, we estimate the probabilities of the following events: (*a*) the agent's opinion will significantly change "towards" his actions and (*b*) the agent's opinion will significantly change in the opposite direction to his actions. Here are the analysis results for two classes, "strongly against" and "strongly for." (The cardinalities of the other classes turned out to be too small.) Figure 21 shows the estimated probabilities of a significant opinion change under the influence of actions: towards actions (blue) and in the opposite direction (red).

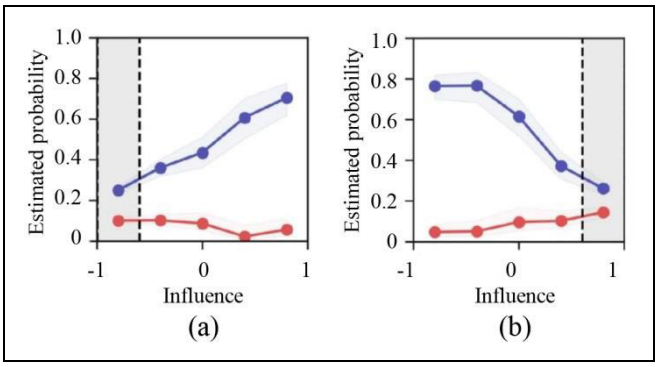


Fig. 21. The estimated probabilities of significant opinion changes for two classes: (a) "strongly against" and (b) "strongly for."

Note. If the agent performed no action between the expressed opinions, the influence is supposed to be 0.

Figure 22 demonstrates the mean and confidence intervals (at a significance level of 0.05) for a significant opinion change due to the influence of actions.

Consequently, if agents change their opinions, they do it most often towards their actions. The greater the difference between the "initial" opinion and the agent's actions is, the higher the probability of opinion change will be (Fig. 21) and the greater magnitude the opinion change towards actions will have (Fig. 22).

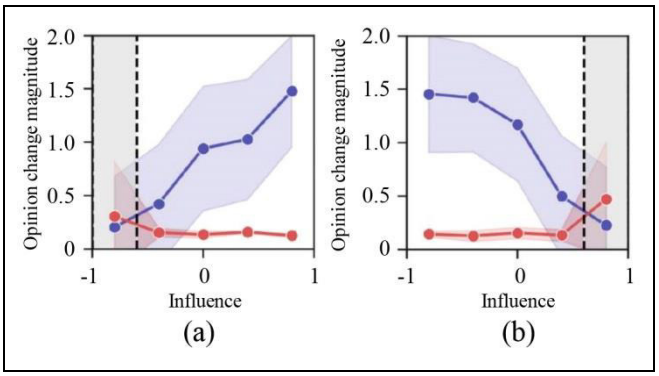


Fig. 22. The magnitude of significant opinion changes for two classes: (a) "strongly against" and (b) "strongly for."

CONCLUSIONS

This paper has presented a primary analysis of the joint dynamics of the opinions and actions of social network agents (*VKontakte* users) on an example of their attitude toward wearing medical masks during the first year of the COVID-19 pandemic.

[9] An interpretation for the share of cases with a significant opinion change.

The opinions of *VKontakte* users have been identified. A satisfactory quality of automatic classification, with an accuracy index of 0.82, has been achieved using deep learning methods.

The network agents with a pronounced position on wearing medical masks have been characterized. As has been discovered, *VKontakte* users are polarized: on the one hand, the share of agents with the "for" position significantly exceeds that of agents with the "against" position; on the other hand, the activity of agents with the "against" position is higher. In general, there is an activity misbalance: the agents are inactive but a small number of agents demonstrated very high activity. If an agent expresses his opinion again, he will do so within a day with probability 0.5. This fact can be explained as follows: agents are involved in discussing new information occasions and the old ones are forgotten.

The dynamics of activity of network agents relevant to wearing masks have been analyzed. The network activity dynamics are characterized by bursts, as a rule, associated with informational events (e.g., the introduction of measures to combat the pandemic). No direct relationship with COVID-19 incidence has been found (most likely, it is implicit). We have found the growing polarization of the network over time (a 30% increase in the number of polarized opinions in one year). The "for"-"against" opinions ratio changed in favor of the negative opinions (a 20% increase in one year).

The networks of information interaction of agents have been examined. In these networks, there are no particular preferences for commenting on agents with a certain position. Agents (especially active ones) prefer to like agents with a similar position regarding wearing masks, but such preferences are not pronounced. However, for the likes network, agents with the "against" position prefer liking agents with a similar position. (They influence like-minded persons and are themselves influenced by them.) At the same time, for agents with the "for" position, the interlocutor's position is not so important. Nevertheless, the agents in the likes network are not in the echo chambers of like-minded persons: the modularity coefficient is too low. This result has been also confirmed by visualizing the network of informational interactions. Hence, agents are exposed to the cross-influence of the social environment and can change their opinions. Therefore, the models of informational influence in social networks should be studied further.

Some important issues have been settled to identify the models of opinion/action dynamics in the future. First, we have confirmed the existence of a small number of agents (called *significant*) who changed their opinion during the period under consideration; see Question no. 2 in the Introduction. They constitute about 1% of the number of agents with "for" or "against" opinions. The analysis of their characteristics has demonstrated the following: there are more males among the significant agents (57%); significant agents are older; the share of significant agents from St. Petersburg and Yekaterinburg is higher compared to those with opinions. Significant agents are weakly connected by friendship links; they prefer to comment on significant agents with opposing positions and like significant agents with similar positions (see Question no. 3). Second, it has been shown that an agent's opinion (his internal state) influences his actions, which are, in turn, "consistent" with the opinion (see Question no. 1). Third, as it has turned out, agents' opinions change towards their actions: the greater the difference between the "initial" opinion and the agent's actions is, the more likely the agent will change his opinion towards his actions and the greater magnitude this change will have (see Question no. 5).

In part II of the study, formal linear models of the joint dynamics of opinions and actions will be identified based on the results obtained (see Questions nos. 5–7). Part III, concluding the study, will be devoted to the identification of binary micro models and the comparison of linear and threshold models (see Questions nos. 4–7).

Acknowledgments. *The authors are grateful to I.V. Kozitsin, V.V. Latynov, A.V. Makarenko, I.V. Petrov, D.V. Ushakov, A.G. Chkhartishvili, and A.A. Shiroky for discussion and constructive remarks.*

REFERENCES

1. Gubanov, D.A., Influence in Social Networks: Formalization Variants, *Large-Scale Systems Control*, 2020, no. 85, pp. 51–71. (In Russian.)
2. Chkhartishvili, A.G., Gubanov, D.A., and Novikov, D.A., *Social Networks: Models of Information Influence, Control and Confrontation*, Cham: Springer, 2019.
3. Gubanov, D.A. and Chkhartishvili, A.G., Meta-Agent and User Influence Levels in a Social Network, *Control Sciences*, 2016, no. 6, pp. 12–17. (In Russian.)
4. Breer, V.V., Novikov, D.A., and Rogatkin, A.D., *Mob Control: Models of Threshold Collective Behavior*, Cham: Springer, 2017.
5. Gubanov, D., A Study of a Complex Model of Opinion Dynamics in Social Networks, *Journal of Physics: Conference Series*, 2021, vol. 1740, pp. 1–6.
6. Allbaracin, D. and Shavitt, S., Attitudes and Attitude Change, *Annu. Rev. Psychol.*, 2018, vol. 69, no. 4, pp. 1–29.
7. Banisch, S. and Olbrich, E., Opinion Polarization by Learning from Social Feedback, *The Journal of Mathematical Sociology*, 2019, vol. 43, pp. 76–103.

8. DeGroot, M., Reaching a Consensus, *Journal of American Statistical Association*, 1974, no. 69, pp. 118–121.
9. Granovetter, M., Threshold Models of Collective Behavior, *The American Journal of Sociology*, 1978, vol. 83, no. 6, pp. 1420–1443.
10. Hunter, J., Danes, J., and Cohen, S., *Mathematical Models of Attitude Change*, Orlando: Academic Press, 1984.
11. Schelling, T., *Micromotives and Macrobehaviour*, New York, London: Norton & Co Ltd, 1978.
12. Xia, H., Wang, H., and Xuan, Z., Opinion Dynamics: A Multidisciplinary Review and Perspective on Future Research, *Int. Journal of Knowledge and Systems Science*, 2011, vol. 2, no. 4, pp. 72–91.
13. Zimbardo, P.G. and Leippe, M.R., *The Psychology of Attitude Change and Social Influence*, Mcgraw-Hill, 1991.
14. Myers, D., *Social Psychology*, 11th ed., McGraw-Hill Education, 2012.
15. Cialdini, R.B., *Influence: The Psychology of Persuasion*, Harper Business, 2006.
16. Pandemic Profiteers: the Business of Anti-vaxx, Center for Countering Digital Hate (CCDH), 2021. URL: https://www.counterhate.com/_files/ugd/f4d9b9_13cbbbef105e459285ff21e94ec34157.pdf.
17. Novikov, D.A. Dynamics Models of Mental and Behavioral Components of Activity in Collective Decision-Making, *Large-Scale Systems Control*, 2020, no. 85, pp. 206–237. (In Russian.)
18. Gubanov, D., Kozitsin, I., and Chkhartishvili, A., COVID-19 Information Consumption and Dissemination: A Study of Online Social Network VKontakte, *Proceedings of the 14th International Conference "Management of Large-Scale System Development"* (*MLSD 2021*), Moscow, 2021, pp. 1–5. URL: https://ieeexplore.ieee.org/document/9600199.
19. Gubanov, D., Kozitsin, I., and Chkhartishvili, A., Face Mask Perception during the COVID-19 Pandemic: An Observational Study of Russian Online Social Network VKontakte, *Advances in Systems Science and Applications*, 2021, vol. 21, no. 3, pp. 91–100.
20. Kuratov, Y. and Arkhipov, M., Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language, *arXiv:1905.07213*, 2019.
21. Babakov, N., Logacheva, V., and Panchenko, A., Beyond Plain Toxic: Detection of Inappropriate Statements on Flammable Topics for the Russian Language, *arXiv:2203.02392*, 2022. DOI: https://doi.org/10.48550/arXiv.2203.02392.
22. Grigoriev, O., Kuznetsova, Y., Nikitina, E., et al., Causative-Emotive Analysis. Part I. Emotional Reactions of Social Networks Users Research, *Psikhologich. Zh.*, 2022, no. 3 (43), pp. 114–121. (In Russian.)
23. Nugamanov, E., Loukachevitch, N., and Dobrov, B., Extracting Sentiments towards COVID-19 Aspects, *CEUR Workshop Proceedings*, Moscow, 2021, pp. 299–312.
24. Pronoza, E., Panicheva, P., Koltsova, O., and Rosso, P., Detecting Ethnicity-targeted Hate Speech in Russian Social Media Texts, *Information Processing and Management*, 2021, vol. 58, no. 6, art. no. 102674.
25. Howe, N. and Strauss, W., *Generations: The History of America's Future, 1584 to 2069*, New York: William Morrow & Company, 1991.
26. Dong, E., Du, H., and Gardner, L., An Interactive Web-based Dashboard to Track COVID-19 in Real Time, *Lancet Inf. Dis.*, 2020, vol. 20(5), pp. 533–534.
27. Newman, M., Mixing Patterns in Networks, *Physical Review E*, 2003, no. 2 (67), p. 026126.
28. Newman, M., Modularity and Community Structure in Networks, *Proceedings of the National Academy of Sciences of the United States of America*, 2006, vol. 103, no. 23, pp. 8577–8696. URL: https://arxiv.org/abs/physics/0602124v1.
29. Clauset, A., Newman, M., and Moore, C., Finding Community Structure in Very Large Networks, *Physical Review E*, 2004, vol. 70, no. 6. DOI: 10.1103/PhysRevE.70.066111.
30. Kozitsin, I., Opinion Dynamics of Online Social Network Users: a Micro-level Analysis, *Journal of Mathematical Sociology*, 2021, pp. 1–41. DOI: https://doi.org/10.1080/0022250X.2021.1956917.

This paper was recommended for publication by F.T. Aleskerov, a member of the Editorial Board.

Received December 28, 2022, and revised March 27, 2023. Accepted April 5, 2023.

Author information

Gubanov, Dmitry Alekseevich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ dmitry.a.g@gmail.com
ORCID iD: https://orcid.org/0000-0002-0099-3386

Novikov, Dmitry Aleksandrovich. Academician, Russian Academy of Sciences; Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ novikov@ipu.ru
ORCID iD: https://orcid.org/0000-0002-9314-3304

Cite this paper

Gubanov, D.A. and Novikov, D.A., Models of Joint Dynamics of Opinions and Actions in Online Social Networks. Part I: Primary Data Analysis. *Control Sciences* **2**, 31–45 (2023). http://doi.org/10.25728/cs.2023.2.4

Original Russian Text © Gubanov, D.A., Novikov, D.A., 2023, published in *Problemy Upravleniya*, 2023, no. 2, pp. 37–53.

Translated into English by *Alexander Yu. Mazurov*, Cand. Sci. (Phys.–Math.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com

DOI: http://doi.org/10.25728/cs.2023.2.5

CREATING FEATURE SPACES AND AUTOREGRESSIVE MODELS TO FORECAST RAILWAY TRACK DEVIATIONS

A.Yu. Vladova

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
Financial University under the Government of the Russian Federation, Moscow, Russia

✉ avladova@ipu.ru

Abstract. Diagnosis of railway tracks reveals the deviations of rail parameters in the plan and profile from their nominal values. If the deviations approach the limit values, the speeds of trains must be reduced. Therefore, forecasting changes in the deviations is a topical problem. Despite the significant amount of diagnostic data collected, railway operators underuse machine learning methods to improve the quality of forecasting. The proposed approach differs from known counterparts as follows. First, the dimensionality of the feature space is increased by calculating the variation of the amplitudes of deviations from the nominal values and two types of areas (the deviation length multiplied by the amplitude and the deviation length multiplied by the variation of the amplitude); subsequently, this space is represented in the 3D matrix form. Second, a set of control parameters is formed; it includes the time and space discretization step, the type of seasonal fluctuations, the number of trend change points, etc. Third, the deviations are forecasted in groups differing in type and position along the track. Forecasting is based on minimizing the empirical risk criterion. As a result, a family of autoregressive models is obtained for each discretization interval along the length of the railway track.

Keywords: time series, diagnosis, software package, discrete technological process.

INTRODUCTION

Railways operate in difficult conditions, i.e., under the effect of natural and climatic factors and loads from passing trains. Railway tracks are diagnosed to measure the track geometry and reveal the deviations of rail parameters in the plan and profile from their nominal values. The revealed deviations are divided as follows:

– track width (narrowing, widening),

– rail level position (misalignments, smooth level variations),

– sag on the right and left rails in the vertical plane (right and left sag),

– rail position in the plan (lining).

The revealed deviations are associated with discrete time instants and track distance and have a nonuniform distribution. If the deviations approach the limit values, the speeds of trains must be reduced. Therefore, forecasting the deviations is a topical problem.

Thus, the goal of this paper is to improve the accuracy of forecasting the deviations of railway tracks. To achieve the goal, we formulate and solve the following tasks: review modern forecasting methods for the sizes of defects (deviations) in extended objects; analyze statistically the deviation parameters as multidimensional data; form the sets of input and output parameters and the combinations of their values; forecast the deviations of railway tracks.

1. A LITERATURE SURVEY

1.1. General analytics

We studied the literature using *Dimensions*, the world's largest linked research database with 135 million publications, 153 million patents, and 7 million grants. The mosaic diagram in Fig. 1 shows the distribution of publications by different fields for the query "*railway degradation*" (in thousand pieces).

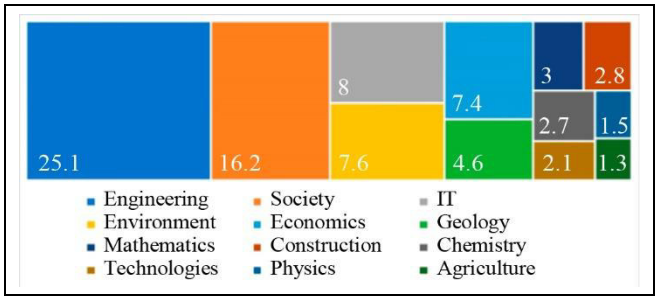


Fig. 1. The distribution of publications (in thousand pieces) by fields for 2012–2022.

With the permanently growing length and operation of railways, the issues of identification and forecasting of their condition are becoming acute. This process is reflected by the increased number of publications and their citations (Fig. 2) over the past 11 years. The characteristic rise of publications and citations is observed in two fields, "Engineering" and "IT"; it seems to be connected with the appearance of new sensors and big data processing methods.

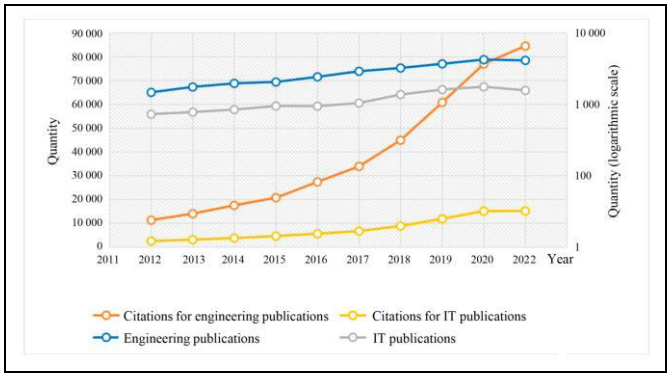


Fig. 2. The dynamics of publications and their citations for the query "*railway degradation*" in the fields "Engineering" and "IT."

1.2. Forecasting railway track deviations

The approaches to forecasting the condition of rail transport systems evolve due to different types of sensors and automatic processing methods for sensor data. The survey [1] was devoted to the condition monitoring of rail transport systems; the authors listed the main areas of development, including the creation of onboard systems based on different types of sensors and corresponding data processing methods. The publication [2] proposed a method to classify car-body vibration signs for the automatic detection of track defects. The causes of the fast degradation of a railway crossing were studied in [3]. The dynamic characteristics of the crossing were assessed using sensor equipment; according to the conclusions, the crossing characteristics degraded fast due to train hunting (the significant effect of wheel impact forces on the rail). The author [4] statistically analyzed diagnostic car data, constructed the probability densities of rail distributions by type, newness, purpose, and class, and clustered the results with transit tonnage recommendations. The paper [5] presented a model for estimating the intensity of railway track wear based on the slope angles of transient curves. According to the literature analysis, forecasting with the multidimensional time series characterizing the technical condition of railway tracks involves two basic types of models [6]:

- autoregressive models, which forecast the values of the time series $\check{z}$ using the past values $z(t),\ldots,z(t-p)$, e.g., by the additive model $\check{z}(t)=a+\varphi_1 z(t-1)+\ldots+\varphi_p z(t-p)+\epsilon(t)$, where φ_p are the weight coefficients and $\epsilon(t)$ is the error;
- regression models of the form $\check{z}(t)=f\left(x_1(t),\ldots,x_n(t)\right)+\epsilon(t)$, which forecast the time series values using a set of features $x_1(t),\ldots,x_n(t)$.

The main disadvantage of all these approaches is the limited use of big data and modern big data processing methods to improve the quality of forecasting the condition of railway tracks. However, in the literature devoted to the analysis of climatic, geotechnical [7, 8], and seismic data [9] as well as astronomical observations, we observe the wide application of classical multivariate parametric models and the Fourier transform (on the one hand) and the relatively recent wavelet transforms, canonical correlations, and trajectory matrices (on the other hand). Furthermore, these models can be adopted in optimization problems to find an optimal set of parameters [10].

1.3. Time series forecasting models implemented in machine learning libraries

We analyzed machine learning methods for time series forecasting on the example of Darts and GreyKite, the open source libraries. Darts contains the implementations of several regression models (ARIMA, Exponential Smoothing, Prophet, forecasting based on the Fast Fourier transform (FFT), etc. [11]). GreyKite provides time series forecasting based on Silverkite, Prophet, and ARIMA models, especially suitable for series with trend change points or seasonal fluctuations [12]. See Table 1 for a qualitative analysis of Silverkite, Prophet, and ARIMA, three modern models.

Table 1

Qualitative analysis of models

Criterion	Silverkite	Prophet	ARIMA
The speed of calculations	High	Low	Medium
The accuracy of forecasting	Very good	Good	Satisfactory
Interpretability	Good	Very good	Good
The ease of use	Satisfactory	Good	Very good
Visibility	Good	Very good	Satisfactory

After the analysis, we selected Prophet [13], an additive regression model with adjustable components. It has the form

$$\check{z}(\mathrm{t}) = g(t) + s(t) + h(t) + \epsilon(t) \qquad (1)$$

with the following notations: t is time; $z(t)$ is the factual time series value; $\check{z}(t)$ is the forecasted time series value; $g(t)$ is the trend component described by a piecewise linear, piecewise logistic, or smooth function; $s(t)$ is the seasonal component (periodic changes), estimated using the partial Fourier sum; $h(t)$ is the component responsible for trend reversal instants (e.g., repair schedule); finally, $\epsilon(t)$ is the error containing the information neglected by this model.

The model allows managing different parameters (the seasonality of the time series, the number of trend change points, and the time step) and is stable to missed data and trend shifts. As a rule, it handles data outliers well.

Adjusting properly the components in equation (1) will ensure an acceptable quality of forecasting at the model output. This quality is estimated through the mean absolute error in percentage [14]:

$$\mathrm{MAE}\,\% = \frac{\sum_{t=1}^{N} \left| z(t) - \check{z}(t) \right|}{\check{z}(t)}, \qquad (2)$$

where N denotes the number of time series points.

In addition, Prophet visualizes the confidence intervals [8] to establish the range of the output feature.

Thus, the problem under consideration has the following statement: on a training sample $\{x_t : t = 1, ..., h\}$, find the control parameter vector U of the model $A(X, U)$ by minimizing the empirical risk (2). The vectors X and U take values from given subsets of the Euclidean spaces E^n and E^r, respectively: $X \subset E^n$, $U \subset E^r$. Constraints can be imposed on X and U: $g_i(X, U) \geq 0,\ i = 1, ..., m$.

2. FORECASTING THE PARAMETERS OF DEVIATIONS

2.1. Statistical analysis of initial data

The initial data occupy 8.9 Mb, are distributed in 10 columns (features), and contain 101 thousand rows. Among them, 91 rows have missed data; see Table 2 below.

Table 2

Initial data format

Assigned index	Feature	Range	The number of unique values
1D	Kilometer	[1, 650]	645
	Picket, m	[1, 1105]	1076
2D	Year	[2018, 2021]	4
	Month	[1, 12]	12
	Day	[1, 31]	29
3D	The code of deviation	[2065, 2161]	17
	Deviation	[Left sag, Wid]	18
Features	Amplitude, mm	[6, 1543]	144
	Nominal amplitude, mm	[10, 1535]	156
	Length, mm	[1, 308]	151
	The degree of danger	[1, 4]	4

According to the analysis results, the data are unbalanced:

- by year (Fig. 3*a*). There were as many deviations in 2020 as years 2018, 2019, and the first two months of 2021 combined;
- by the type of deviations (Fig. 3*b*). The distributions by year and type are presented in Fig. 3*d*. In 2020, the number of diagnosed deviations significantly increased; in 2021, the types of deviations increased, from 11 in 2018 to 18 in 2021. Moreover, the most frequent deviations changed from year to year: sag (Left sag, Right sag), misalignment (Mis), narrowing (Nar), widening (Wid), and smooth level variation (Lev);
- by the railway track length. Significantly more deviations were detected at the railway hubs compared to the track distance on average (Fig. 3*c*).

Due to the small number of some types of deviations, it is possible to combine them into a group or to generate additional data using bootstrap methods.

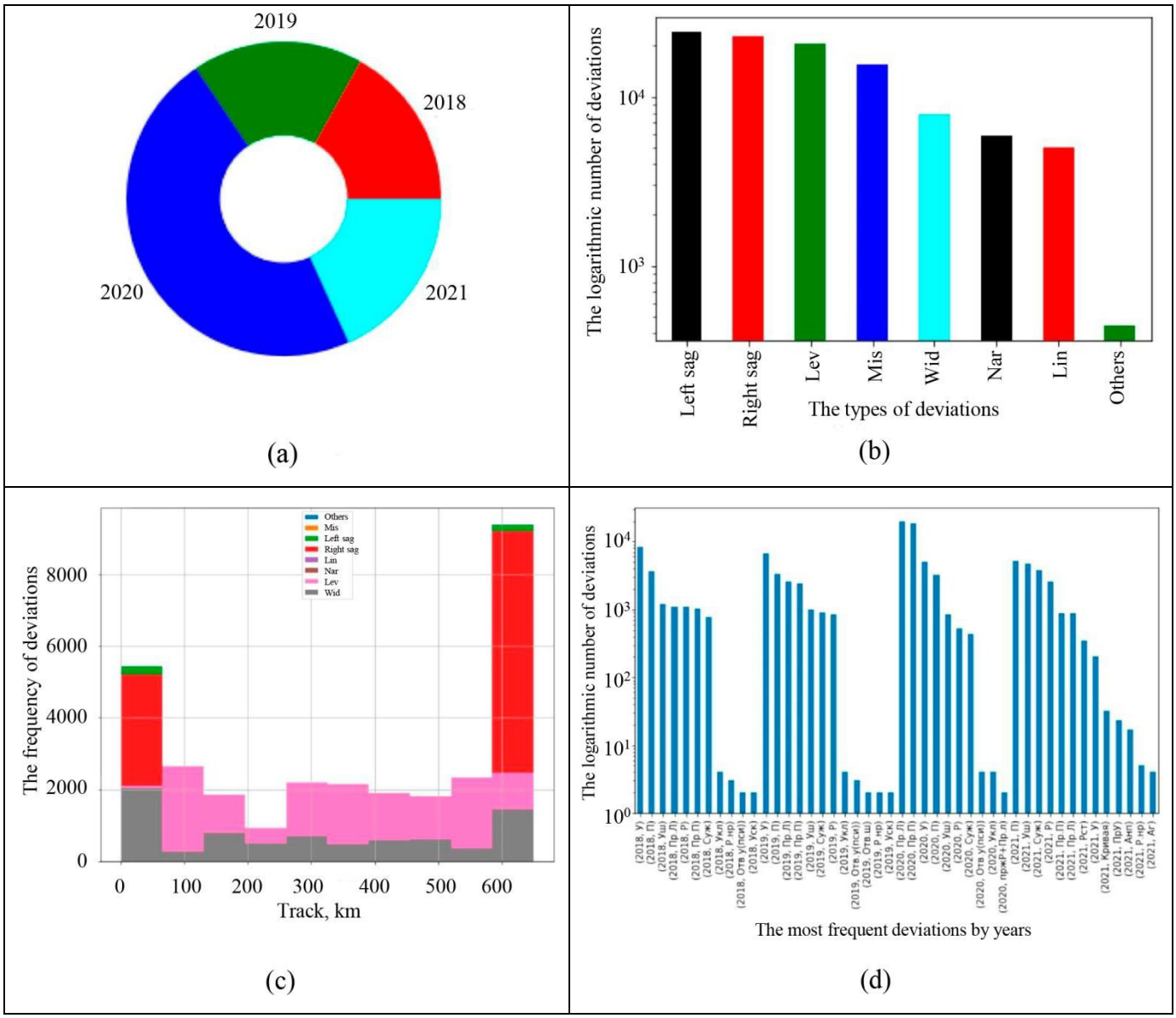


Fig. 3. Unbalanced data: (a) by year, (b) by the type of deviations, (c) by distance, and (d) by year and the type of deviations. In Figs. 3*b*–3*d*, Lin denotes lining.

2.1.1. Identify trend and seasonality

The presence of a trend and seasonality was checked using the upsampling and downsampling operations by varying the discretization step of the time grid:

• The trend for decreasing the amplitudes of deviations is traced by calculating their median values for each year. However, the lengths of the deviations first decreased and then sharply increased (Fig. 4*a*); the trend change point corresponds to year 2020. Note that "Danger," an aggregate parameter, is indifferent to changes in the median amplitude and the length of deviations.

• Downsampling by time (bimonthly data) and transition to the logarithmic scale (Fig. 4*d*) showed that the amplitudes of deviations have the highest variations in July. These changes suggest the presence of seasonality in the data. The locally optimal lengths of deviations, to some extent, follow those of the amplitudes with a time lag but have no pronounced seasonality. Also, no seasonal fluctuations were graphically identified for the degree of danger.

2.1.2. Increasing the dimensionality of the feature space

Since the time is represented by three integer features ("Year," "Month," and "Day"), the calculated feature "Time" was formed on their basis. The combination of two features, "Kilometer" and "Picket," characterizes the location of a deviation and forms the

new feature "Distance." To increase the stationarity of the time series, the differences between the features "Amplitude" and "Nominal Amplitude" were calculated and the attribute "Variation" was created. For a more complete description of the attribute "Variation" with a planar rather than point value, we introduced two aggregate features, "Area_A" and "Area_V" (the deviation length multiplied by the amplitude and the deviation length multiplied by the variation of the amplitude, respectively). In addition, 11 infrequent deviations were combined into the feature "Others" to increase the number of time series values. Fig. 5 demonstrates the peculiarities of the new features:

- the differences in the spread of deviations: only positive values for "Widening," "Misalignment," and "Sag"; only negative values for "Narrowing" with a significant degree of danger; a large dispersion for "Smooth level variations" (Fig. 5*a*);
- the clusters of the feature "Area_A" values by the types of deviations when passing to the logarithmic scale (Fig. 5*b*).

Hence, it is necessary to use different forecasting models [15, 16] for each type of deviations or to normalize their values.

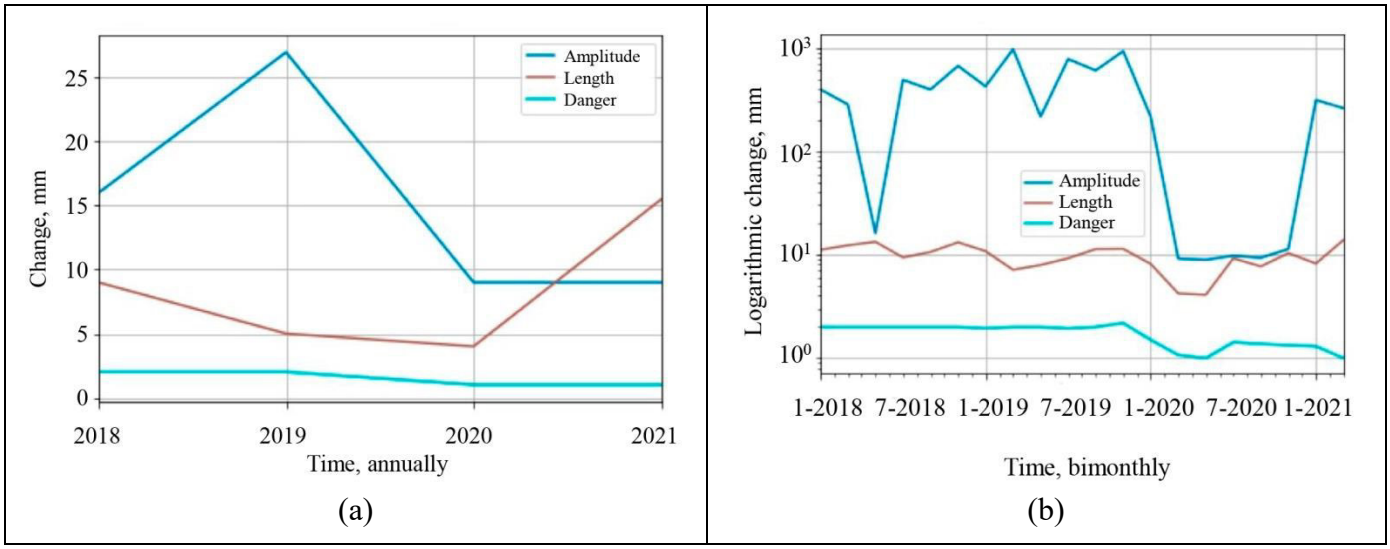


Fig. 4. The identification of (a) trend and (b) seasonality in the features "Amplitude," "Deviation length," and "The degree of danger."

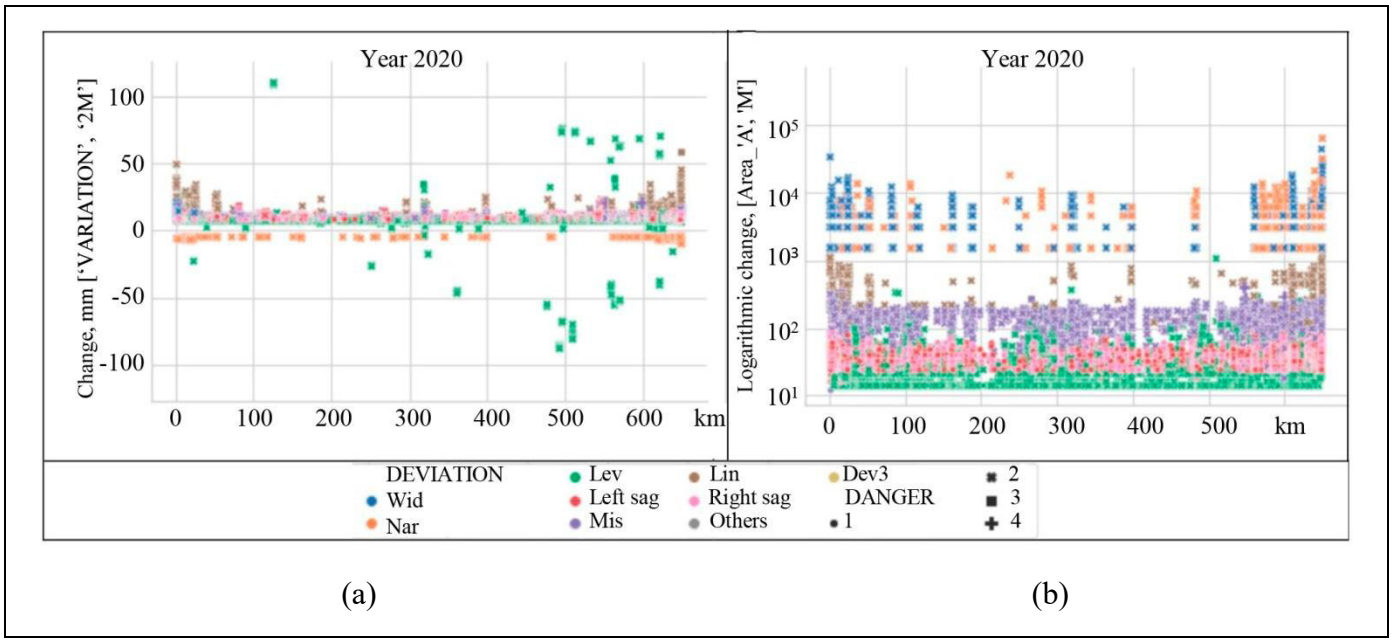


Fig. 5. Analysis of the dispersion by the type of deviation: (a) the spread of values and (b) value clusters.

2.1.3. Representing the data in the 3D matrix form

Each type of deviations may have specific dynamics (the law of variation), and each kilometer of railway tracks may have geotechnical peculiarities. Therefore, in the forecasting models, the length and amplitude of deviations, their nominal values as well as the aggregate features were written as a 3D matrix with the following axes:

– the location along the railway track,

– the time (the instant of detecting a deviation),

– the code of deviation.

A fragment of the transformed data is demonstrated in Fig. 6.

2.1.4. Probability densities

Probability densities were constructed on different time grids (with averaging from several weeks to a quarter) for the initial values of lengths, amplitudes, the variations of amplitudes, the areas of each type of deviations, and the degree of deviation. As it turned out, the values of the features "Variation" and "Area_V" are closest to the Gaussian distribution (Fig. 7).

The probability densities of the other features correspond to a two-tailed distribution. Therefore, their values were normalized into the interval [0, 1] by the formula

$$x_{i,\,\mathrm{norm}} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \qquad (3)$$

where $x_{i,\,\mathrm{norm}}$ denotes the normalized value of a feature, $x_{\min}$ and $x_{\max}$ are its smallest and largest values, respectively, and x_i is the ith initial value of the feature.

Thus, the statistical analysis of the parameters of deviations as multidimensional data revealed the following properties: the data are unbalanced by year and the type of deviations, and trends and local minima appear when varying the discretization step of the data by year, month, and quarter. As a result, the least frequent types of deviations were grouped; the dimensionality of the feature space was increased by adding the position of deviations and the time instants of their detection and by calculating the variation of the deviation amplitudes from the nominal values and the corresponding areas of deviations (by the length, amplitude, and amplitude variation); the resulting feature space was represented in the 3D matrix form.

			Km	Amplitude	Normative	Type	Year	Danger	Length	Variation	Area_A	Area_V
Code	Track	Time										
2081	1469	2018-01-29	1	1529.0	1520	Wid	2018	2	4	9.0	6116.0	36.0
	1614	2018-01-29	1	1530.0	1520	Wid	2018	2	2	10.0	3060.0	20.0

Fig. 6. A fragment of the transformed data

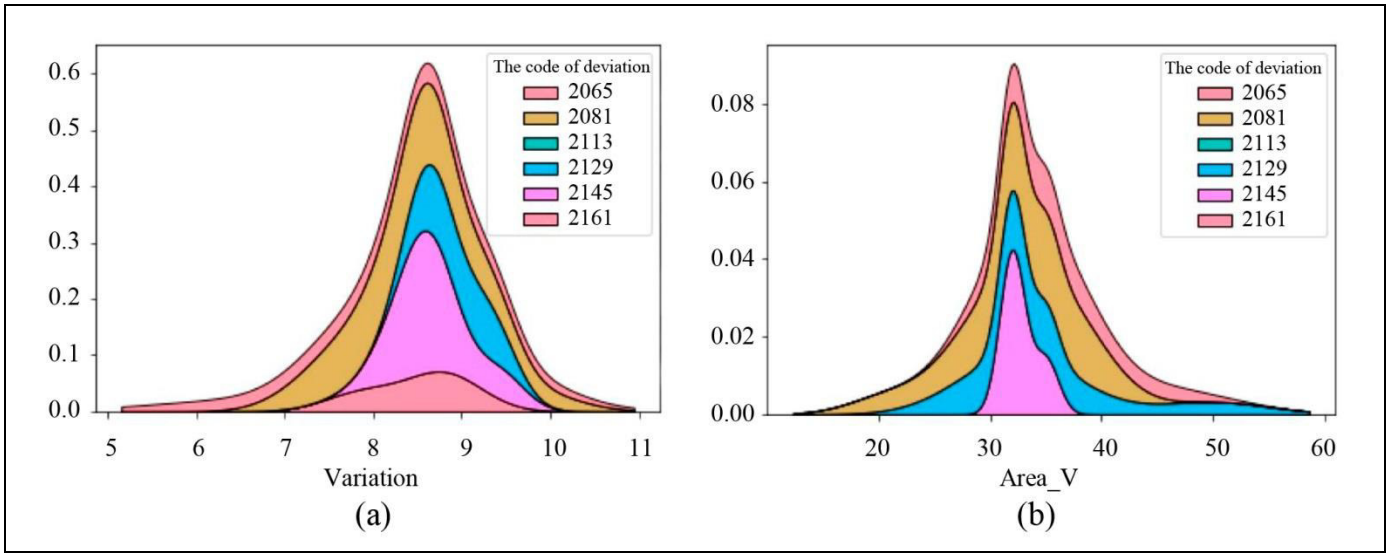


Fig. 7. The probability densities of feature values: (a) "Variation," averaged on the six-week scale, and (b) "Area_V," averaged on the seven-week scale.

2.2. The method for forecasting the parameters of railway track deviations

The proposed method for forecasting the parameters of railway track deviations includes three stages as follows. The first stage includes statistical analysis to group the rare types of deviations and form the model parameter vector X (the distance and time, the variation of amplitudes from the nominal values, and the areas of deviations). Then, the feature values are indexed along the three axes (distance, the code of deviation, and time) to form time series for each kilometer and the type of deviation.

In the second stage, the feature space is increased by adding the normalized and logarithmic lengths, amplitudes, the variations of amplitudes, areas, and the degrees of danger). Then, the control parameter vector U is constructed, including the time discretization step, seasonality, and the number of trend change points. After that, the constraints $g_i(U)$ are formed considering the following parameters:

– data depth and forecasting step: one year; three, two, or one quarter; two or one month; five, three, or two weeks;

– the type of seasonality: annual, quarterly, monthly, or weekly;

– the number of trend change points, according to the number of repairs;

– the type of seasonality model (additive or multiplicative);

– the rate of seasonal fluctuations (the number of terms in the partial Fourier sum).

In the third stage, under sufficiently many deviations in the group characterized by the type of deviation and the track kilometer, the model A (1) is trained in each group with estimating the empirical risk MAE% (2). As a result, the optimal values of the parameter vector U are determined for each group in terms of the minimum empirical risk criterion.

The practical significance of this method consists in the possibility to gain new knowledge and optimize railway transport.

2.3. Implementation

Table 3 shows the sets of model parameters corresponding to the best quality of forecasting.

The mean absolute error MAE% ranges from 2.6% to 8.4% for the recommended model parameters. According to the calculation results, we draw the following conclusions:

– Quarterly and annual grids significantly degrade the quality of forecasting.

– The best forecasts are for “Misalignment,” “Lining,” “Left sag,” “Widening,” and “Others” (the aggregate of 11 rarely encountered deviations).

– At most two trend change points should be selected.

– For the five-week grid, the best values of MAE% are achieved in the case of two trend change points.

– The lowest values of MAE% are achieved when considering the weekly, monthly, and quarterly seasonal fluctuations.

– Taking the logarithm of the feature values worsens the quality of forecasting.

– Normalizing the feature values into the interval [0, 1] improves the quality of forecasting.

Table 3

The results of forecasting

MAE%	Time step	Deviation	Variation	The number of trend change points
2.68	1 month	Misalignment	AREA_A	0
3.03	3 weeks	Lining	LENGTH	0
3.27	3 weeks	Widening	LENGTH	2
3.89	3 weeks	Left sag	VARIATION	0
4.36	1 month	Misalignment	AMPLITUDE	1
4.45	3 weeks	Others	LENGTH	2
4.60	5 weeks	Misalignment	VARIATION	2
5.34	1 month	Misalignment	AMPLITUDE	1
5.34	1 month	Right sag	AMPLITUDE	2
5.74	3 weeks	Others	AREA_V	0
5.74	3 weeks	Widening	AREA_V	0
6.59	1 month	Misalignment	DEGREE	2
7.90	5 weeks	Lining	LENGTH	2
8.39	3 weeks	Others	AREA_A	0
8.39	3 weeks	Widening	AREA_A	0

– Centering the values (subtracting the mean) and normalizing them (dividing by the dispersion) and using multiplicative regressors for seasonality worsens the quality of forecasting.

Figure 8*a* shows the one-month forecast for the feature "Misalignment"; Fig. 8*b*, the three-week forecast for the feature "Widening." The black dots are the factual values, the blue line corresponds to the model values, and the blue domain is the uncertainty corridor. Figures 8*c* and 8*d* present the identified trends; Figs. 8*d*–8*h*, the variations of the seasonal component, monthly and quarterly (Figs. 8*e* and 8*g*), and weekly

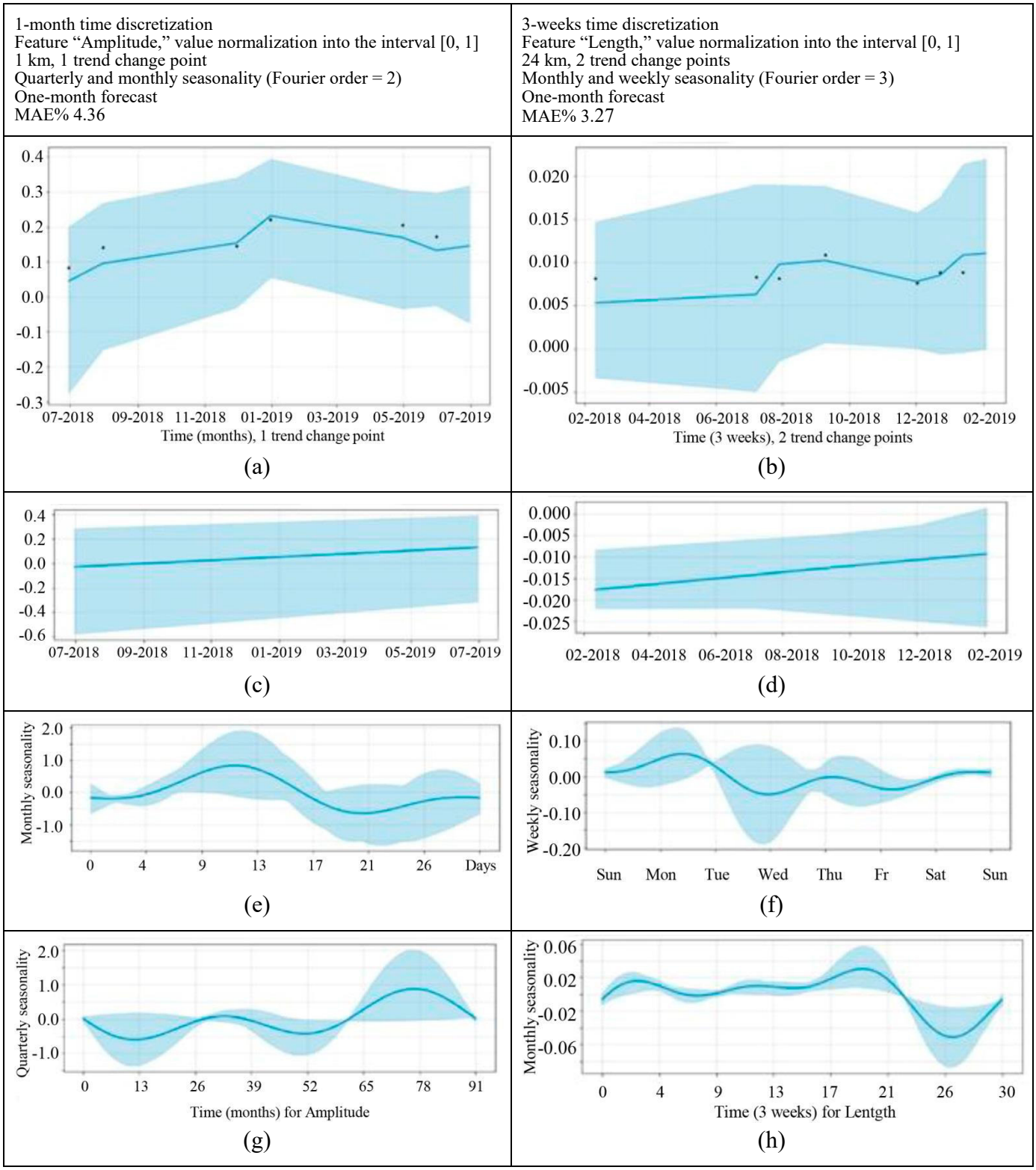


Fig. 8. The results of forecasting: (a), (b) model-based; (c), (d) trend; (e)–(h) seasonality.

and monthly (Figs. 8*f* and 8*h*). For both models, the uncertainty corridors for the seasonal and trend components are calculated as well.

According to the one-month forecast, the length and amplitude of deviations are increasing. The last point of the training sample is the local optimum point. The growth of amplitudes for the feature "Misalignment" falls on the last month in the quarter. This is probably due to repair work.

To implement the method, we developed a program in the Python language. The program forms a family of forecasting models for different groups of the types of deviations and the kilometers of roadway tracks on the features. It can be used to optimize railway transportation, which is of practical significance.

The Python language and the Google Colaboratory platform were applied to manipulate, visualize, and study the data. The statistical analysis results were visualized using Matplotlib, Pandas, and Seaborn libraries. The models were trained using SciPy, Sklearn, and Prophet libraries.

CONCLUSIONS

According to the literature survey, machine learning methods have proven their effectiveness in identifying and forecasting the condition of objects under a fairly long history of observations.

The statistical analysis carried out in this paper allows:

– revealing the number of deviations on any given time interval for any part of the railway track;

– comparing the sums of the variations of deviations from the nominal values for any railway track section at different time instants;

– normalizing the feature values into a certain interval;

– establishing data unbalancedness for different time intervals and types of deviations;

– synthesizing control parameters and features (time and distance, normalized and logarithmic lengths, amplitudes, the variations of amplitudes, areas, and the degrees of danger).

The parameters of railway track deviations have been statistically analyzed and the feature space has been supplemented by the variation of the amplitudes of deviations from the nominal values and the areas of deviations and variations.

The proposed method for forecasting the parameters of railway track deviations allows:

– defining the control parameter vector U;

– forming the constraints $g_i(U)$;

– obtaining a family of forecasting models by the types of deviations and railway track sections in terms of the minimum error (relative risk) criterion.

The control parameter vector of the forecasting model considers the type of seasonal fluctuations (from weekly to annual), the type of their description (additive or multiplicative), the rate of seasonality change (the number of terms in the partial Fourier sum), and the number of trend change points (in the general case, the number of time series points minus one).

To increase the accuracy of forecasting, including deviations with a shallow detection history, we have proposed to apply upsampling along the railway track (one and a half kilometers, two kilometers, etc.).

Data Availability. *A code fragment and a data fragment (three years of homogeneous measurements on two trace sections) that confirm the presented results can be found on the author's GitHub account: https://github.com/avladova/Railway-track-deviations or the author's website http://vladova.ru/About.*

REFERENCES

1. Kostrzewski, M. and Melnik, R., Condition Monitoring of Rail Transport Systems: A Bibliometric Performance Analysis and Systematic Literature Review, *Sensors*, 2021, vol. 21, no. 14, art. no. 4710.
2. Tsunashima, H., Condition Monitoring of Railway Tracks from Car-Body Vibration Using a Machine Learning Technique, *Applied Sciences*, 2019, vol. 9, art. no. 2734.
3. Liu, X. and Markine, V.L., Train Hunting Related Fast Degradation of a Railway Crossing-Condition Monitoring and Numerical Verification, *Sensors*, 2020, vol. 20, no. 8, art. no. 2278.
4. Vladova, A.Yu., Identification of the Railway Track Technical State, *Proceedings of 2021 14th International Conference "Management of Large-Scale System Development" (MLSD)*, Moscow, 2021, pp. 1–5.
5. Dubitskii, I.S., Enin, A.V., and Vladova, A.Yu., Analysis of Railway Track Wear Dynamics, *Trudy 14-oi mezhdunarodnoi konferentsii "Upravlenie razvitiem krupnomasshtabnykh sistem"* (Proceedings of the 14th International Conference "Management of Large-Scale System Development" (MLSD-2021)), Moscow, 2021, pp. 979–985. (In Russian.)
6. Lukashin, Yu.P., *Adaptivnye metody kratkosrochnogo prognozirovaniya vremennyh ryadov* (Adaptive Methods for Short-Term Time Series Forecasting), Moscow: Finansy i Statistika, 2003. (In Russian.)
7. Vladova, A.Yu., Harmonic Analysis of Temperatures of the Frozen Soil of Right-of-Way of the Linear Object, *Occupational Safety in Industry*, 2017, no. 7, pp. 25–30. (In Russian.)
8. Vladova, A.Yu., Remote Geotechnical Monitoring of a Buried Oil Pipeline, *Mathematics*, 2022, vol. 10, no. 11, art. no. 1813.
9. Lyubushin, A.A., *Analiz dannykh sistem geofizicheskogo i ekologicheskogo monitoringa* (Analysis of Geophysical and

Environmental Monitoring Systems Data), Moscow: Nauka, 2007. (In Russian.)
10. Valeev, S.G., *Regressionnoe modelirovanie pri obrabotke dannykh* (Regression Modeling in Data Processing), Kazan: FEN, Tatarstan Academy of Sciences, 2001. (In Russian.)
11. Herzen, J., Lässig, F., Piazetta, S.G., Neuer, T., et al., Darts: User-Friendly Modern Machine Learning for Time Series, *Journal of Machine Learning Research*, 2022, vol. 23, pp. 1–6.
12. Hosseini, R., Yang, K., Chen, A., and Patra, S., A Flexible Forecasting Model for Production Systems, *ArXiv: 2105.01098v1*, 2021, DOI: https://doi.org/10.48550/arXiv.2105.01098.
13. Taylor, S. and Letham, B., Forecasting at Scale, *The American Statistician*, 2018, vol. 72, no. 1, pp. 37–45.
14. Vandeput, N., Forecast KPI: RMSE, MAE, MAPE & Bias Towards Data Science, URL: https://towardsdatascience.com/forecast-kpi-rmse-mae-mape-bias-cdc5703d242d. (Accessed February 4, 2023.)
15. Hartomo, K. and Nataliani, Y., A New Model for Learning-Based Forecasting Procedure by Combining k-means Clustering and Time Series Forecasting Algorithms, *PeerJ Computer Science*, 2021, vol. 7, no. 2, art. no. e534. DOI: https://doi.org/10.7717/peerj-cs.534.
16. Vivas E., Allende-Cid H., and Salas R., A Systematic Review of Statistical and Machine Learning Methods for Electrical Power Forecasting with Reported MAPE Score, *Entropy*, 2020, vol. 22, no. 12, art. no. 1412.

This paper was recommended for publication by A.S. Mandel, a member of the Editorial Board.

Received November 30, 2021, and revised March 27, 2023. Accepted April 13, 2023.

Author information
Vladova, Alla Yur'evna. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia; Financial University under the Government of the Russian Federation, Moscow, Russia
✉ avladova@ipu.ru

Cite this paper
Vladova, A.Yu., Creating Feature Spaces and Autoregressive Models to Forecast Railway Track Deviations. *Control Sciences* **2**, 46–55 (2023). http://doi.org/10.25728/cs.2023.2.5

Original Russian Text © Vladova, A.Yu., 2023, published in *Problemy Upravleniya*, 2023, no. 2, pp. 54–64.

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com

DOI: http://doi.org/10.25728/cs.2023.2.6

INTER-ORBITAL SPACECRAFT TRANSFER: TRAJECTORY DESIGN BY ITERATING PARAMETER VALUES WITHIN A DATA GRID

E.V. Savvina

✉ petrakowae@mail.ru

Abstract. This paper considers the problem of designing an optimal inter-orbital spacecraft transfer. We present a computational algorithm and modeling results of the optimal transfer trajectory between near-Earth elliptical orbits for a spacecraft with a chemical booster and fixed thrust. The trajectory design procedure includes four stages as follows: a) formation of the primary ranges of initial approximations for typical optimization problems; b) iterative integration to find the domains of convergence for a typical variational problem; c) determination of the optimum for each problem statement within the accepted ranges and its implementation by calculating the final conditions residuals; d) analysis of the results obtained. We use numerical methods of mathematical analysis and mathematical programming. The risk of "overstepping" the potentially optimal result is minimized by varying the accuracy at different stages of calculations. Based on the results, we improve the primary solution of the reference problem statement, identify the domains of convergence of solutions, and obtain the sets of initial approximation vectors ensuring convergence in the considered problems for further analysis. The results of this study can be used to develop further and refine the algorithm for selecting optimal initial approximations for different optimization problems (including spacecraft trajectory optimization as a typical one).

Keywords: optimal control, spacecraft trajectory optimization, maximum principle, mathematical modeling, nonlinear programming.

INTRODUCTION

Currently, there exist many methods for solving optimal control problems. However, Pontryagin's maximum principle [1] is one of the most widespread approaches to dynamic optimization. This method yields optimality conditions, including the cases when optimal control is on the admissible domain boundary. Also, it allows deriving all necessary conditions for the variational calculus problem, reducing the original problem to a boundary value problem of differential equations [1]. Like other methods [2], the maximum principle requires an initial approximation for the parameter values, and the correct choice of initial approximations provides faster convergence and successful determination of an optimum. At the same time, the choice problem is connected with the branching of optimal solutions and the high sensitivity of the residuals of the boundary value problem to its parameter variations [3]. Moreover, despite the possibility to create formal estimation algorithms under a given initial approximation, one should follow intuition and a priori knowledge to select good approximations [2, 4].

Modern literature offers some practical methods facilitating the choice of initial approximations, such as the homotopy of maximum thrust [5], edge normalization [6], edge estimation by a reference trajectory [7], and others. In special cases, some of the developed methods turn out useful. For example, we mention approximate rephrasing solutions, which were developed to obtain initial approximations for indirect methods [7–10]. Relevant studies focus on particular problems; solving some of them gives new (auxiliary) methods for finding initial approximations for relevant problems. Nevertheless, most of the modern approaches still involve the trial-and-error method, often more effective than subtle counterparts. It consists in choosing initial approximations based on a priori knowledge and intuition [2–4, 8].

Thus, more and more efforts of the global research community are applied to find algorithms that will be effective in choosing correct initial approximations for optimal control problems; for example, see [2, 4–6, 8].

This paper implements the first stage of research, intended to identify the relationships between the vector components ensuring convergence for typical inter-orbital spacecraft transfer optimization problems. The first part of the study identifies the domains of convergence for typical optimization problems within accepted ranges. It reduces to forming initial approximation vectors ensuring the best solutions within a data grid. Expectedly, this study will contribute to refining initial approximation choice algorithms for typical problems.

1. PROBLEM STATEMENT

This paper considers the problem of designing an optimal inter-orbital transfer trajectory for a spacecraft.

1.1. The general formulation and parameters of the optimization problem

We consider a spacecraft on a given initial near-Earth orbit. The spacecraft includes a chemical booster with some known characteristics. This booster must transfer the spacecraft to a working near-Earth orbit with specified characteristics.

It is required to find a rational transfer scheme between the orbits. As an optimality criterion, we choose the spacecraft mass inserted into the working orbit: the mass is maximized. The transfer time is not limited.

We fix the following parameters and conditions for all cases under consideration: the spacecraft mass on the initial orbit is 5000 kg; the thrust of the unregulated rocket engine of the booster is 5 kN; the number of engine ignitions is arbitrary; the specific impulse is 330 s; the spacecraft transfer scheme is limited to one revolution; the orbits belong to the same plane; the apsidal lines of the orbits coincide; the gravity field is Newtonian.

We vary the following parameters: the perigee altitude of the initial orbit and its apogee altitude; the perigee altitude of the final orbit and its apogee altitude.

The transfer scheme characteristics are as follows:

– the start point of the spacecraft on the initial orbit,

– the number of active and passive sections on the transfer trajectory,

– the duration of active and passive sections of the trajectory and their location on the transfer trajectory (in other words, the time instants of engine ignition and cutoff),

– the pitch angle program on each active section,

– the end point on the final orbit.

1.2. Spacecraft transfer model

The mathematical model of the spacecraft motion includes the vector of its phase coordinates with the following components: the radial velocity V_r, the normal velocity V_n, the radius r, and the polar angle β. Recall that the sequence of active and passive sections is not fixed, and we design an optimal law of engine ignition and cutoff. Therefore, it is reasonable to add the spacecraft mass m to the listed variables, and the resulting vector of the phase variables (further called the phase vector) takes the form

$$z = \begin{pmatrix} V_r \\ V_n \\ r \\ \beta \\ m \end{pmatrix}. \tag{1}$$

The spacecraft motion is described by the system of differential equations

$$\begin{aligned} \dot{V}_r &= \frac{P\sin(\vartheta)}{m}\delta - \frac{\mu}{r^2} + \frac{V_n^2}{r}, \\ \dot{V}_n &= \frac{P\cos(\vartheta)}{m}\delta - \frac{V_r V_n}{r}, \\ \dot{r} &= V_r, \\ \dot{\beta} &= \frac{V_n}{r}, \\ \dot{m} &= -q\delta. \end{aligned} \tag{2}$$

The notations are as follows:

P is the engine thrust (unregulated, a known value);

ϑ is the true anomaly;

μ is the Earth's gravitational parameter;

q is the mass flow rate of the engine (a known value);

δ is the thrust function taking only two values: $\delta = 1$ (ignition) and $\delta = 0$ (cutoff);

φ is the pitch angle of the spacecraft (the angle between the thrust vector and the local horizon);

$\delta(t)$ and $\varphi(t)$ are the control functions to be optimized.

In the first stage of the analysis, we fix the motion conditions in the initial orbit perigee as the initial conditions:

$$\begin{aligned} V_r(t_0) &= 0, \\ V_n(t_0) &= \sqrt{\frac{\mu}{p_0}(1+e_0)}, \\ r(t_0) &= \frac{p_0}{1+e_0}, \\ \beta(t_0) &= 0, \\ m(t_0) &= m_0. \end{aligned} \tag{3}$$

Here, t_0 is the start time, which can be set to 0 (the time is counted from the start), and p_0 and e_0 are the focal parameter and the eccentricity of the initial orbit, respectively. The subscript "*f*" in the relations below denotes belonging to the final motion conditions. We count the angular range (the polar angle) from the apsidal line of the initial orbit.

The spacecraft transport problem arising in the first stage is as follows: for the set of initial conditions (3), find the control functions $\delta(t)$ and $\varphi(t)$ and the transfer time t_f under which the spacecraft will reach the phase space point

$$V_r(t_f) = 0,$$

$$V_n(t_f) = \sqrt{\frac{\mu}{p_f}(1-e_f)},$$

$$r(t_f) = \frac{p_f}{1-e_f},$$

$$\beta(t_f) = \pi$$

with the minimum fuel consumption, $m(t_f) \to \max$.

1.3. The mathematical optimization problem

We introduce an auxiliary function (Hamiltonian). It can be treated as the scalar product of two vectors: the right-hand sides of the motion equations and the conjugate variables. The vector of conjugate variables, further called the conjugate vector, has the same dimension as the phase vector; each component of the conjugate vector corresponds to some phase variable. In other words, the dimension of the phase vector is 5; see (1).

The conjugate vector has the form

$$\lambda = \begin{pmatrix} \lambda V_r \\ \lambda V_n \\ \lambda r \\ \lambda\beta \\ \lambda m \end{pmatrix}.$$

The Hamiltonian is given by

$$H = \lambda V_r\left(\frac{P\sin(\vartheta)}{m}\delta - \frac{\mu}{r^2} + \frac{V_n^2}{r}\right)$$
$$+\lambda V_n\left(\frac{P\cos(\vartheta)}{m}\delta - \frac{V_r V_n}{r}\right) + \lambda r V_r + \lambda\beta\frac{V_n}{r} - \lambda m q\delta.$$

According to the maximum principle, the chosen control law maximizes the Hamiltonian, i.e., the optimal control functions ($\delta(t)$ and $\varphi(t)$) can be found from the maximum conditions for the Hamiltonian.

In addition, it is possible to show that

$$\cos(\vartheta_{\text{opt}}) = \frac{\lambda V_n}{\lambda V},\ \sin(\vartheta_{\text{opt}}) = \frac{\lambda V_r}{\lambda V},$$

where

$$\lambda V = \sqrt{\lambda V_r^{\,2} + \lambda V_n^{\,2}},$$

$$\delta_{\text{opt}} = \begin{cases} 1 \text{ if } \Psi > 0 \\ 0 \text{ if } \Psi < 0, \end{cases}$$

$$\Psi = \frac{P}{m}\lambda V - \lambda m q \quad \text{or} \quad \Psi = \frac{W}{m}\lambda V - \lambda m.$$

Here, Ψ denotes the engine switching function and W is the exhaust velocity. The subscript "opt" means that the corresponding relations are derived by maximizing the Hamiltonian.

The obtained pitch angle program is as follows:

$$\text{tangag} = \left\{ \begin{array}{l} \arccos\left(\dfrac{\lambda V_n}{\sqrt{(\lambda V_n)^2 + (\lambda V_r)^2}}\right) \text{ if } \lambda V_r > 0 \\ -\arccos\left(\dfrac{\lambda V_n}{\sqrt{(\lambda V_n)^2 + (\lambda V_r)^2}}\right) \text{ if } \lambda V_r \le 0 \end{array} \right\}.$$

Considering the optimal control laws (the pitch angle program and the optimal thrust function), the equations of the phase variables on the optimal trajectory take the form

$$\begin{aligned} \dot{V}_r &= \frac{P\lambda V_r}{m\lambda V}\delta_{\text{opt}} - \frac{\mu}{r^2} + \frac{V_n^2}{r}, \\ \dot{V}_n &= \frac{P\lambda V_n}{m\lambda V}\delta_{\text{opt}} - \frac{V_r V_n}{r}, \\ \dot{r} &= V_r, \\ \dot{\beta} &= \frac{V_n}{r}, \\ \dot{m} &= -q\delta_{\text{opt}}. \end{aligned} \qquad (4)$$

Due to the maximum principle, the conjugate variables satisfy the system of differential equations

$$H = \lambda V_r\left(\frac{P\sin(\vartheta)}{m}\delta - \frac{\mu}{r^2} + \frac{V_n^2}{r}\right)$$
$$+\lambda V_n\left(\frac{P\cos(\vartheta)}{m}\delta - \frac{V_r V_n}{r}\right) + \lambda r V_r + \lambda\beta\frac{V_n}{r} - \lambda m q\delta,$$

$$\frac{d\lambda_i}{dt} = -\frac{\partial H}{\partial z_i}.$$

Consequently,

$$\frac{d\lambda V_r}{dt}=-\frac{\partial H}{\partial V_r}=\lambda V_n\frac{V_n}{r}-\lambda r,$$
$$\frac{d\lambda V_n}{dt}=-\frac{\partial H}{\partial V_n}=-\lambda V_r\frac{2V_n}{r}+\lambda V_n\frac{V_r}{r}-\lambda\beta\frac{1}{r},$$
$$\frac{d\lambda r}{dt}=-\frac{\partial H}{\partial r}=\lambda V_r\left(-\frac{2\mu}{r^3}+\frac{V_n^{\,2}}{r^2}\right)+\lambda V_n\left(-\frac{V_rV_n}{r^2}\right)+\lambda\beta\frac{V_n}{r^2},\quad(5)$$
$$\frac{d\lambda\beta}{dt}=-\frac{\partial H}{\partial\beta}=0.$$

The corresponding boundary value problem of the maximum principle is as follows: find values of the conjugate vector components at the start point, $\lambda_{Vr}(t_0)$, $\lambda_{Vn}(t_0)$, $\lambda_r(t_0)$, $\lambda_\beta(t_0)$, and $\lambda_m(t_0)$, and the transfer time t_f (six unknowns in total) such that

$$H(t_0)=0,$$
$$V_r(t_f)=0,$$
$$V_n(t_f)=\sqrt{\frac{\mu}{p_f}}(1-e_f),$$
$$r(t_f)=\frac{p_f}{1-e_f},$$
$$\beta(t_f)=\pi,$$
$$\lambda m(t_f)=1.$$

In the second stage of the analysis, the start and end points of the spacecraft transfer trajectory are floating. To implement this requirement, we introduce the transversality conditions.

1.4. The transversality conditions at the start and end points of the transfer trajectory

The transversality condition expresses the perpendicularity of the conjugate vector to all tangent vectors of the boundary manifold.

If the phase variables at a boundary point (first, the start point) are a function of some chosen parameter (in the case under consideration, the true anomaly of the initial orbital point, υ_0), the tangent vector of the initial manifold has the components

$$\left[\frac{d}{d\upsilon_0}(V_{r0}),\frac{d}{d\upsilon_0}(V_{n0}),\frac{d}{d\upsilon_0}(r_0),\frac{d}{d\upsilon_0}(\beta_0),\frac{d}{d\upsilon_0}(m_0)\right].$$

Calculating the derivatives, we obtain the vector

$$\left[\frac{e_0\cos(\upsilon_0)}{\sqrt{p_0}},\frac{-e_0\sin(\upsilon_0)}{\sqrt{p_0}},\frac{p_0e_0\sin(\upsilon_0)}{(1+e_0\cos(\upsilon_0))^2},1,0\right].$$

The optimality conditions for the start point on the initial orbit are given by the perpendicularity of the conjugate vector and this tangent vector. The perpendicularity condition can be written as

$$\lambda V_r(t_0)\frac{e_0\cos(\upsilon_0)}{\sqrt{p_0}}-\lambda V_n(t_0)\frac{e_0sin(\upsilon_0)}{\sqrt{p_0}}+\lambda r(t_0)\frac{p_0e_0sin(\upsilon_0)}{(1+e_0\cos(\upsilon_0))^2}+\lambda\beta(t_0)\cdot1+\lambda m(t_0)\cdot0=0.$$

Hence,

$$\lambda\beta(t_0)=-\lambda V_r(t_0)\frac{e_0\cos(\upsilon_0)}{\sqrt{p_0}}+\lambda V_n(t_0)\frac{e_0\sin(\upsilon_0)}{\sqrt{p_0}}-\lambda r(t_0)\frac{p_0e_0\sin(\upsilon_0)}{(1+e_0\cos(\upsilon_0))^2}.$$

Similarly, the optimality condition for the transfer end point (the optimality of the final angular distance) has the form

$$\lambda\beta(t_f)=-\lambda V_r(t_f)\frac{e_f\cos(\beta_f)}{\sqrt{p_f}}+\lambda V_n(t_f)\frac{e_f\sin(\beta_f)}{\sqrt{p_f}}-\lambda r(t_f)\frac{p_fe_f\sin(\beta_f)}{(1+e_f\cos(\beta_f))^2}.$$

2. A DATA GRID TO FIND INITIAL APPROXIMATIONS

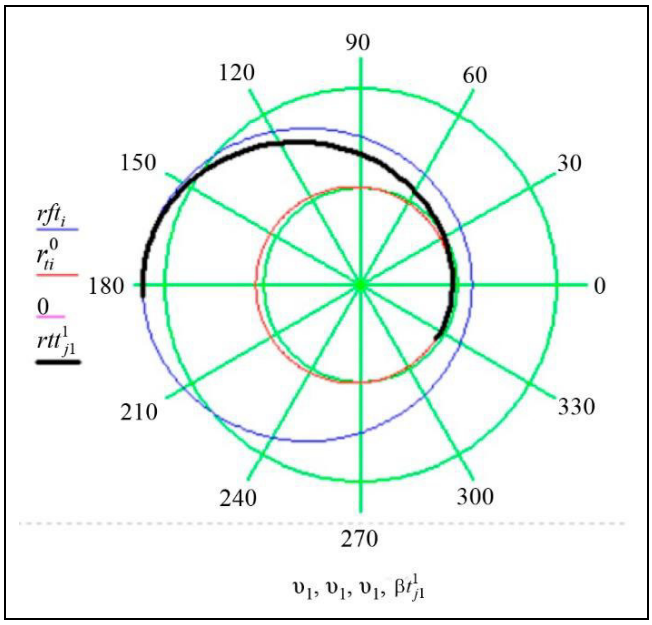
The algorithm for further study (the trajectory design procedure) consists of four stages:

- calculation of optimal intra-orbital transfer trajectories with different parameters for the spacecraft with a chemical booster and a fixed thrust to identify and refine the ranges of initial approximations for typical problems. It produces a data grid for an exhaustive search over the initial values vector;
- iterative integration to find the domains of convergence and reach an optimum of the variational problem;
- determination of the optimum for each problem statement within the accepted ranges and its implementation by calculating the final conditions residuals.
- analysis of the results obtained.

For the primary choice of the range of initial approximations, we solve an optimization problem with the following orbit data: 500-km perigee altitude and 1500-km apogee altitude (the initial orbit), and 2000-km perigee altitude and 10 000-km apogee altitude (the final orbit). For these data, the Bard method [2] yields the following initial approximations (with a final spacecraft mass of 3203.788 kg):

$$\begin{pmatrix} \lambda m(t_0) \\ \lambda V_r(t_0) \\ \lambda V_n(t_0) \\ \lambda r(t_0) \\ tf \\ \upsilon_o \\ \beta_f \end{pmatrix} = \begin{pmatrix} 0.6 \\ -0.065851324495908 \\ 1.3618066619686777 \\ 1.4193211140743078 \\ 6.803580030230557 \\ -0.6054269338992805 \\ 3.14159265358979 \end{pmatrix}.$$

The aircraft transfer scheme after calculating the residuals is presented in the figure below. The notations are as follows: rtt^1_{j1} is the transfer trajectory obtained with integrating by the Runge–Kutta method of the fourth order with a variable step; βt^1_{j1} is the change in the polar angle βt^1 on this trajectory.



The optimal transfer scheme (choice of initial values).

This solution is taken as a reference to form the range for each of the unknown variables. To determine such a range, we add the values for the left and right boundaries at the initial time instant; they are calculated so that, according to the generated initial data vectors, the sizes of the domains of convergence will be sufficient to identify the relationships between the parameter values and the degree of convergence of the results. Note that the ranges should be not very large to reduce the risk of "overstepping" the possible solution during integration (to avoid critical steps).

Thus, we choose the following ranges to find good approximations for the unknown variables at the initial time instant:

$$\lambda V_{r0} = -0.079, ..., -0.058,$$
$$\lambda V_{n0} = 1.291, ..., 1.431,$$
$$\lambda r_0 = 1.361, ..., 1.491,$$
$$\lambda m_0 = 0.3, ..., 0.7,$$
$$\upsilon_0 = -0.61, ..., -0.60405.$$

Over 4 145 000 variations of the initial approximation vectors were checked for each of the five problems in the selected range to identify the relationships between the parameter values and convergence.

The data grid (see Table 1) was loaded into Python.

3. THE ITERATIVE ALGORITHM FOR SOLVING THE SET OF TYPICAL OPTIMIZATION PROBLEMS

3.1. The iterative algorithm for solving the optimization problem

The algorithm was implemented in MathCad in two stages as follows.

The first stage is to find the preliminary domains of convergence areas by iterating parameter values (performing an exhaustive search) within the data grid. Note that the solution accuracy is set to 10^{-3}. This moderate value reduces the risk of "overstepping" the potential solution and allows avoiding the multiply increasing number of vectors to be checked (in the case of four and more decimal places). In the second stage, we return to the required accuracy of 10^{-14} and refine the preliminary solution by calculating the residuals.

These stages are discussed in detail below.

3.1.1. The first stage (preliminary solution)

The first step of the algorithm is to enter the data and reduce them to a common dimensionless form. We select five sets of input parameters to demonstrate the algorithm:

1. 400-km perigee altitude and 1400-km apogee altitude (the initial orbit), and 1900-km perigee altitude and 9900-km apogee altitude (the final orbit).

2. 400-km perigee altitude and 1400-km apogee altitude (the initial orbit), and 2000-km perigee altitude and 10 000-km apogee altitude (the final orbit).

Table 1

The data grid for the exhaustive search when integrating by the Runge–Kutta method of the fourth order

V_r	V_n	R	λV_r	λV_n	λr	υ_0	λm
–0.0394875	1.05887075	0.942166729	–0.079	1.291	1.361	–0.61	0.3
–0.0394875	1.05887075	0.942166729	–0.079	1.291	1.361	–0.61	0.4
–0.0394875	1.05887075	0.942166729	–0.079	1.291	1.361	–0.61	0.5
–0.0394875	1.05887075	0.942166729	–0.079	1.291	1.361	–0.61	0.6
–0.0394875	1.05887075	0.942166729	–0.079	1.291	1.361	–0.61	0.7
–0.0394677	1.05888457	0.942154435	–0.079	1.291	1.361	–0.6097	0.3
–0.0394677	1.05888457	0.942154435	–0.079	1.291	1.361	–0.6097	0.4
–0.0394677	1.05888457	0.942154435	–0.079	1.291	1.361	–0.6097	0.5
–0.0394677	1.05888457	0.942154435	–0.079	1.291	1.361	–0.6097	0.6
–0.0394677	1.05888457	0.942154435	–0.079	1.291	1.361	–0.6097	0.7
–0.0394479	1.05889838	0.942142147	–0.079	1.291	1.361	–0.6093	0.3
–0.0394479	1.05889838	0.942142147	–0.079	1.291	1.361	–0.6093	0.4
–0.0394479	1.05889838	0.942142147	–0.079	1.291	1.361	–0.6093	0.5
–0.0394479	1.05889838	0.942142147	–0.079	1.291	1.361	–0.6093	0.6
–0.0394479	1.05889838	0.942142147	–0.079	1.291	1.361	–0.6093	0.7

3. 400-km perigee altitude and 1400-km apogee altitude (the initial orbit), and 2100-km perigee altitude and 10 100-km apogee altitude (the final orbit).

4. 500-km perigee altitude and 1500-km apogee altitude (the initial orbit), and 2000-km perigee altitude and 10 000-km apogee altitude (the final orbit).

5. 600-km perigee altitude and 1600-km apogee altitude (the initial orbit), and 2000-km perigee altitude and 10 000-km apogee altitude (the final orbit).

All varying characteristics are entered iteratively in a loop with a counter from 0 to 4 (five problem statements). Due to a very significant load on the computational system (exhaustive search within the data grid and iterative calculations), we divided the program by stages into the following subprograms:

- a program with preliminary calculations, which outputs the results in a separate Excel file;
- a program with basic calculations in the loop;
- a program that visualizes the results.

The input data for the problem include:

- the Earth's gravitational parameter (398 600 $\frac{\text{km}^3}{\text{s}^2}$) and the Earth's radius (6371 km);
- the initial spacecraft mass (5000 kg), the thrust of the chemical rocket engine (5000 N), and its specific impulse (330 s·g);
- the perigee and apogee altitudes of the initial and final orbits as well as the angle between their apsidal lines;
- the elements of the initial and final orbits (the perigee radius rp and the apogee radius ra, the semi-major axis A, the energy constant h, the eccentricity e, and the focal parameter p);
- the orbital equation, i.e., the length of the spacecraft radius vector as a function of the true anomaly:

$$r = \frac{p}{1 + e \cdot \cos(\vartheta)}.$$

Next, the entered dimensional quantities are converted to dimensionless form. Dimensionless quantities reduce the CPU load by replacing thousands of kilometers with the normalized values. Also, the number of input arguments is reduced.

The mathematical model is the equations of the planar motion of the spacecraft. The orbital coordinate system is used to analyze the spacecraft velocity (the radial V_r and transversal V_n components). The spacecraft position is considered in the polar coordinate system: the principal axis x is directed along the radius vector of the apsidal point of the initial orbit. In this case, r is the length of the radius vector and b is the polar angle.

The angle θ in the equations is the pitch angle of the spacecraft measured from the local horizon line. All variables in the system of differential equations under consideration are dimensionless.

See Section 1, system (2), for the initial mathematical model. The final model based on the maximum

principle is represented by the system of ten first-order ordinary differential equations (4) and (5).

Integration involves the Rkadapt tool of MathCad. The initial approximations for Rkadapt are found by exhaustive search within the data grid; see Section 2. The data grid is presented in Table 1. In this stage of calculations, the solution accuracy is set to 10^{-3}.

In the first stage, the preliminary check of the solutions (insertion into the final orbit) is visual by the graphs of the results since the final conditions are not considered and the required accuracy is not observed. The algorithm fixes all sets of vectors without convergence and distributes all solutions with convergence into three groups for further analysis:

– insertion into the final orbit (for each case),

– insertion into an orbit above the final one,

– insertion into an orbit below the final one.

Note that the final spacecraft mass on the orbit (the transfer optimality criterion) is calculated but not considered due to insufficient accuracy.

Thus, the main result of the first stage is the set of initial value vectors ensuring insertion into the final orbit (for each specific case) and the sets of initial value vectors for further analysis.

3.1.2. The second stage (optimal solution)

In the second stage, the algorithm returns to the required accuracy 10^{-14} and operates the sets of candidate vectors (the ones ensuring insertion into the final orbit).

To find the exact solution and then the optimum, we developed a program calculating the final conditions residuals at a floating point of inserting into the final orbit. The program requires the transversality conditions derived in Section 2.

The program outputs the vector of residuals, which is used to find an exact solution: equating their values to 0 allows obtaining the factual initial variables, the final transfer time, and the polar angle characterizing the transfer end point.

In this stage, the spacecraft mass at the transfer end point is analyzed and the optimum (the solution of the problem) is identified. In addition, the effect of start point variations within the data grid on the final spacecraft mass (on the final orbit) is assessed.

The result obtained within the data grid is the vectors ensuring the optimal transfer between the orbits. For problem statements 1–5, they are combined in Table 2. The notations are as follows: T_F is the transfer time; β_F is the polar angle characterizing the transfer end point; m_F is the spacecraft mass on the final orbit. Here, the subscript "1" indicates the parameters at the initial time instant with the best result by the final aircraft mass criterion.

Thus, the implemented approach improved the result for the "reference" problem (statement 4) by 365 g compared to the Bard method [2].

3.2. Analysis and discussion of the results

Five typical problems were calculated in a loop. The domains of convergence were identified and the best solutions satisfying the optimality condition were obtained within the grid (the ones with the maximum aircraft mass on the final orbit). The risk of "overstepping" the potentially optimal result was reduced using variable accuracy in different stages of the calculations.

The results—the vectors of the desired variables for each case—were written in Excel tables using MathCad. They will be estimated and analyzed by statistical methods in a Python program.

With the proposed approach to calculations and accuracy, the initial result for the "reference" problem was improved by 365 g without leaving the ranges for each value.

Table 2

The optimal transfer characteristics in problem statements 1–5 (analysis within the data grid)

Characteristics	Statement 1	Statement 2	Statement 3	Statement 4	Statement 5
ϑ_0^1	–0.608528181266	–0.687262955509	–0.58694862445	–0.58063271537992	–0.67460422765
λV_r^1	–0.065265825615	–0.075257944004	–0.06564653582	–0.06466045327036	–0.06339073139
λV_n^1	1.3623340777782	1.3653869345017	1.364158466294	1.36474044140864	1.357648817782
λr^1	1.4200011568480	1.4207257561587	1.421069918027	1.42243156356652	1.40764863211
T_F	6.7321648922778	7.0228415355241	12.30599404742	12.48386539989280	5.855669428846
β_F	3.1311300732552	3.166796288751	5.127686975088	5.33807657900554	2.859452408788
m_F	3227.339 kg	3182.552 kg	3181.12 kg	3204.153 kg	3247.748 kg

Thus, this paper demonstrates the effectiveness of exhaustive search within the data grid when refining and improving the initial result in optimal inter-orbital spacecraft transfer problems. This paper is the first (preliminary) stage of research aimed at identifying the mathematical relationships between the vector components ensuring convergence for typical problems. Calculations were carried out in parallelized programs in MathCad 15 and Python 3.9 mainly based on a Core i5 1035G1 processor. The total running time of the programs was 8 hours.

CONCLUSIONS

In this paper, we have studied and extended the capabilities of mathematical programming with application to typical optimization problems (optimization of the spacecraft transfer trajectory between near-Earth elliptical orbits). In addition, we have demonstrated an effective approach to finding optimal initial approximations for the variational problem of inter-orbital spacecraft transfer optimization with the minimum mass flow criterion.

The results are as follows:

- Over 4 145 000 variations of the initial approximations were checked for each problem in MathCad and Python programs to identify relationships between the parameter values and convergence.
- The primary solution of the reference variational problem of inter-orbital spacecraft transfer optimization was improved within the considered ranges.
- Five typical problems were iteratively solved in a loop in MathCad and their parameter sets were examined.

The results of this study can be used to develop further and refine an algorithm for selecting optimal initial approximations for different optimization problems (including spacecraft trajectory optimization as a typical one). Expectedly, they will simplify the solution of such problems and will contribute to the refinement and development of the corresponding mathematical apparatus.

REFERENCES

1. Vedyakova, A.O., Milovanovitch, E.V., Slita, O.V., and Tertychny-Dauri, V.Yu., *Metody teorii optimal'nogo upravleniya* (Methods of Optimal Control Theory), Moscow: ITMO University, 2021. (In Russian.)
2. Bard, Y., *Nonlinear Parameter Estimation*, New York–London: Academic Press, 1979.
3. Petukhov, V.G., Optimization of Interplanetary Trajectories for Spacecraft with Ideally Regulated Engines Using the Continuation Method, *Cosmic Research*, 2008, vol. 46, no. 3, pp. 219–232.
4. Kitrell, J.R., Mezaki, R., and Watson, C.C., Estimation of Parameters for Nonlinear Least Squares Analysis, *Industrial & Engineering Chemistry*, 1965, vol. 57, pp. 18–27.
5. Hofmann, C., and Topputo, F., Embedded Homotopy for Convex Low-Thrust Trajectory Optimization with Operational Constraints, *Proceedings of 2022 AAS/AIAA Astrodynamics Specialist Conference*, Charlotte, NC, USA, 2022, pp. 1–16.
6. Jiang, F., Baoyin, F., and Li, J., Practical Techniques for Low-Thrust Trajectory Optimization with Homotopic Approach, *Journal of Guidance, Control and Dynamics*, 2012, vol. 35, no. 1, pp. 245–258.
7. Wu, D., Wu, C., Lin, F., et al., Analytical Costate Estimation by a Reference Trajectory-Based Least-Squares Method, *Journal of Guidance, Control and Dynamics*, 2022, vol. 45, pp. 1–9.
8. Wu, D., Cheng, L., Gong, S., and Baoyin, H., Approximate Time-Optimal Low-Thrust Rendezvous Solutions Between Circular Orbits, *Aerospace Science and Technology*, 2022, vol. 131, part A, art. no. 108011.
9. Bevilaqua, R., Analytical Guidance Solutions for Spacecraft Planar Rephasing via Input Shaping, *Journal of Guidance, Control and Dynamics*, 2014, vol. 37. DOI: 10.2514/1.G000008.
10. Wu, D., Wu, Ch., Lin, F., and Baoyin, H., An Atlas of Optimal Low-Thrust Rephasing Solutions in Circular Orbit, *arXiv:2209.07418v1*, 2022. DOI: 10.48550/arXiv.2209.07418.

This paper was recommended for publication by L.B. Rapoport, a member of the Editorial Board.

Received January 23, 2023, and revised March 21, 2023. Accepted April 13, 2023.

Author information

Savvina, Elena Valer'evna. Applicant for a degree, Moscow, Russia
✉ petrakowae@mail.ru

Cite this paper
Savvina, E.V., Inter-orbital Spacecraft Transfer: Trajectory Design by Iterating Parameter Values within a Data Grid. *Control Sciences* **2**, 56–63 (2023). http://doi.org/10.25728/cs.2023.2.6

Original Russian Text © Savvina, E.V., 2023, published in *Problemy Upravleniya*, 2023, no. 2, pp. 65–74.

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com

ON THE 110TH ANNIVERSARY OF ACADEMICIAN BORIS N. PETROV'S BIRTH

March 11, 2023, marked the 110th anniversary of Academician Boris Nikolaevich Petrov's birth. He was an outstanding scientist and scientific organizer in the field of automatic control. B.N. Petrov's R&D activities were closely connected with Trapeznikov Institute of Control Sciences (the USSR, then Russian, Academy of Sciences; before 1969, the Institute of Automation and Remote Control, the USSR Academy of Sciences), where he rose from the engineer to the world-renowned scientist. From 1947 to 1951, he was Director of the Institute.

Boris Nikolaevich was remarkable for his encyclopedic knowledge and wide range of research interests. Actively working in the general theory of automatic control, he always selected the most relevant problems.

He developed the structural transformation method for the block diagrams (schemes) of automatic systems and a corresponding mathematical apparatus, the algebra of structural transformations. B.N. Petrov is a founder of the theory of invariant control systems. His multifaceted investigations on invariance theory resulted in new principles and structures for various types of combined systems.

In 1957, Boris Nikolaevich led the research in the theory, design, and development of searchless self-adaptive systems (adaptive systems with a model). Under his guidance and participation, adaptive control systems for several classes of rockets by Chief Designer I.S. Seleznev were designed and developed, for the first time in the USSR.

B.N. Petrov's studies of nonstationary (time-varying) and multivariate systems, as well as his contributions to sensitivity theory and control algorithm design as an inverse dynamics problem, are widely known.

Boris Nikolayevich was a talented teacher. He began tutoring at Ordzhonikidze Moscow Aviation Institute in 1944 by creating a new course of lectures on the automation of motors and propellers. The significance of that course went beyond the technical narrative: in his lectures, B.N. Petrov delivered to the students the most important and recent results in the theory of automatic control of those years.

In 1954, the Institute was entrusted by a government decree to lead R&D works on a propulsion control system for the intercontinental ballistic missile developed by S.P. Korolev. Boris Nikolaevich took responsibility for the ideology of a fundamentally new class of systems—terminal fuel control systems for liquid-propellant engines—that significantly increased the rocket's power output by sharply reducing the guaranteed fuel reserves.

The development of thrust control systems for liquid-propellant engines and the synchronization of tank emptying for complex-architecture rockets were acute problems. They were accompanied by many challenges on the pathway of creating fundamentally new systems that started from scratch, without any background history, prototype systems, and literature sources.

The results of R&D works by B.N. Petrov and the Institute's employees during collaboration with academician S.P. Korolev, OKB-1 Chief Designer, and other outstanding Soviet scientists and engineers in the field of space rocket technology contributed significantly to developing onboard control systems of

launch vehicles, particularly the launch of the first artificial Earth satellite and Yu.A. Gagarin's pioneering space flight.

The results obtained by Boris Nikolayevich and his team were of a fundamental character. The control systems on their basis became an integral part of all large liquid-propellant rockets developed by chief designers S.P. Korolev, M.K. Yangel, V.N. Chelomey, and V.F. Utkin.

B.N. Petrov's ideas were further refined and applied in modern R&D works of the Institute in the field of rocket and space technology. They were embodied in terminal control systems for a new generation of launch vehicles and upper stages for space and defense applications (modernized Soyuz-2, the Angara family, Sarmat, Soyuz-5, Amur, and oxygen-hydrogen heavy-class upper stages (KVTK)).

Since 1956, an important area of B.N. Petrov's work was the theory and control systems for artificial Earth satellites. Boris Nikolayevich participated in the following activities: creating pre-damping systems for gravitationally stabilized Earth satellites, control systems for communication satellites and direct TV broadcasting satellites in the geostationary orbit, and several multi-seat manned orbital aircraft; developing automatic stations; injecting the world's first artificial moon satellite into a near-moon orbit.

In the last years of his life, B.N. Petrov headed the Intercosmos Council of the USSR Academy of Sciences. He was in charge of the successful development and implementation of major international space programs. Among them, note the Soyuz-Apollo project involving researchers, engineers, and designers from the USSR and the USA. Petrov personally contributed to solving numerous organizational, scientific, and technical problems of the project.

Under Boris Nikolaevich's guidance, large professional teams grew up. His scientific school successfully develops topical problems of modern control theory.

B.N. Petrov wrote about 200 journalistic and popular science articles on major scientific problems in automation, computer engineering, experiment automation, and program management of space research. He supported everything new and promising in science and repeatedly emphasized the importance of developing a mathematical or abstract systems theory. As he said, this theory extends the horizons of control science.

Boris Nikolaevich was not only a great researcher but also an outstanding scientific organizer. In 1953, he was elected Corresponding Member of the USSR Academy of Sciences; in 1960, Academician of the Academy. Since 1963, B.N. Petrov was the Academician-Secretary of the Department of Mechanical Engineering and Control Processes (the USSR Academy of Sciences); in 1979, he was elected Vice-President of the USSR Academy of Sciences.

B.N. Petrov was entitled the Hero of Socialist Labor. He was awarded the Order of Lenin (5 times), the Order of the October Revolution, the Labor Red Banner Order, the Red Star Order, as well as the USSR Lenin and State Prizes.

Boris Nikolaevich's activities were widely recognized in other countries. He was Full Member of the International Academy of Astronautics and Foreign Member of the Czechoslovak, Hungarian, Bulgarian, and Polish Academies of Sciences. Moreover, he was awarded several foreign orders and the Gold Medal of the French National Center for Space Research.

In August 1980, an untimely death took B.N. Petrov away, full of creative energy. In November 1980, the Government issued a decree on perpetuating his memory. The Presidium of the USSR Academy of Sciences established the Petrov Gold Medal (since 1993, the Petrov Prize), awarded for outstanding research in the theory and systems of automatic control and experimental research in space exploration.

The name of Boris Nikolaevich Petrov will forever remain in the annals of the national science of control and astronautics.

Employees of Trapeznikov Institute
of Control Sciences RAS,
the Editorial Board and Editorial Office of the Journal

Cite this paper
On the 110th Anniversary of Academician Boris N. Petrov's Birth. *Control Sciences* **2**, 64–65 (2023).

Original Russian Text © 2023, published in *Problemy Upravleniya*, 2023, no. 2, pp. 75–76.

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com