St. Petersburg State Univeristy

Manuscript copyright

Uzhva Denis Romanovich

ADAPTIVE MESO-SCALE CONTROL OF COMPLEX MULTI-AGENT NETWORKED DYNAMICAL SYSTEMS

1.2.3. Theoretical informatics, cybernetics

Thesis for the degree of Candidate of Physical and Mathematical Sciences

Translation from Russian

Scientific supervisor:
Doctor of Physical and Mathematical Sciences, Professor
Oleg Nikolayevich Granichin

St. Petersburg
2025

Contents

Introduction

Recently, computing devices have undergone significant changes due to the increasing intellectualization of various areas of human activity, be it manufacturing, medicine or, for example, logistics. To solve large-scale problems, it is necessary to create complex information systems with a large number of computers interacting with each other, since it is often not possible to solve current problems with the resources of a single machine within required time.

At the same time, information processes are becoming more and more voluminous—problems of so-called Big Data and the "curse of dimensionality" arise. Moreover, often excessively voluminous data is subject to noise and disturbances, which further complicates their analysis. Big data arises in many areas of human activity: for example, in medicine we have to deal with the analysis of three-dimensional images of the brain and other organs [13] (C. Austin), [102] (A A. Tahmassebi), in agriculture there is also a need for optimal coordination of the work of a large number of agricultural machinery [91] (P. Ribarics), and in particle physics the problem of data processing in experiments ion collisions amounting to petabytes are solved simultaneously by a large-scale network of computers distributed throughout the world [2,3] (NA61/SHINE CERN collaboration). Working with such data becomes possible only by combining the efforts of many computing devices into a single network for distributed collaboration. In the context of control and information processing tasks distributed computing has a number of useful features: scalability (the solution to the problem can be accelerated by the additive introduction of new devices into a common network), fault tolerance (if several devices fail, the tasks assigned to them are redistributed to other computers) and security (attack on some computing nodes will not bring down the entire system, since such nodes can be temporarily eliminated without significant damage to the operation of the entire system as a whole). These features give rise to more and more relevant research aimed at creating technologies for distributed analysis of big data: [30] (M. H. DeGroot), [82] (M. Pease), [65]

(N. A. Lynch), [72] (S. Nakamoto), [79] (D. Ongaro), [41] (M. J. Fischer).

The problem of big data is typical not only for information systems in the classical sense as a network consisting of desktop computers. There is another trend: the number of all kinds of embedded mobile devices that allow computerization of physical objects is growing. Integrated into a network, "smart" objects form the so-called "Internet of Things" — a network of cyber-physical systems (from the English cyber-physical systems). For example, today the development of unmanned vehicles is becoming more and more active, which opens up space for new solutions to the problem of "balancing" transport traffic on the roads. These solutions can be divided into two classes: centralized and decentralized [33] (G. Egger). In the first case, cars report their coordinates to some data center, which plays the role of a coordinator who makes decisions on balancing traffic of the entire system. In the second, there is no such center, and machines, with the help of local interactions, come to a consensus that satisfies the solution to the problem. Today, centralized hierarchical strategies are more preferable because they provide direct and clear system monitoring, which is important for stable operation. However, simplicity comes at the price of inevitable costs for communications between computers and data centers, which on a large scale can lead to a lack of communication channel capacity [93] (F. Rossi), [85] (S. Rashid). In other words, the naturally limited bandwidth of the channels also limits the scalability of the system, which makes such a solution unsuitable for working with big data. At the same time, the described disadvantage is practically absent in the decentralized strategy: depending on the load, new nodes can be attached to those computers that have relatively few connections with others. In this case, there is no need to connect all nodes into a fully connected network, since, by influencing the state of neighbors (hereinafter, neighbors mean nodes communicating with each other), one way or another, each computer in the system will be indirectly connected with the others. Such systems are usually called *multi-agent systems* (MAS), and the nodes in this case are called (intelligent) agents [118] (M. Wooldridge), [40] (M. Falco), [114] (O. Vinyals). The study, creation and use of multi-agent systems has become commonly referred to as *multiagent technologies* (MT). MT is based on a decentralized approach to problem solving, in which dynamically updated information in a distributed network of intelligent agents is processed not in some center, but directly on the agents themselves based on their local observations together with locally available information from neighbors [46] (O. N. Granichin). At the same time, both resource and time costs for communications in the network, as well as the time for processing and decision-making in the center of the entire system (if it still exists) are significantly reduced.

Today, multi-agent technologies have not yet become widespread due to insufficient knowledge regarding MAS. However, multi-agent algorithms such as Particle Swarm Optimization (PSO) [14] (M. Babanezhad), [103] (A. Tharwat), gravitational search [34] (L. V. Enikeeva), [96] (H. Shehadeh), stochastic diffuse search [4] (M. Al-Rifaie) or the ant algorithm [8] (N. O. Amelina), [7] (K. S. Amelin) are already successfully used in the industry to solve optimization problems. The main obstacle in the development of such algorithms is the irreducibility of the behavior of individual agents to the behavior of the whole system [45] (J. Goldstein), [62] (P. Lodge), which cybernetics is called the phenomenon of *emergence*. In this regard, today we are still at the stage of considering relatively simple problems of achieving consensus states, for example, balancing the load on computing nodes [16] (M. Bandupadhyay). However, the growing need for decentralized solutions and, as a consequence, increasing interest in multi-agent technologies inevitably lead to a discussion of projects of complex swarm robotic systems for medicine and industry [90] (A. Requicha), as well as physics and mechanics [59] (T.A. Khantuleva).

A major role in the mathematical formalization of multi-agent systems was played by studies using the control and optimal decision making theories, based on the works [104] (J. N. Tsitsiklis), [112] (T. Vicsek), [88, 89] (W. Ren), [77, 78] (R. Olfati-Saber), [29] (F. L. Lewis), [28, 35, 48, 52] (O. N. Granichin), [47] (V. I. Gorodetsky), [115] (P. O. Skobelev), [43] (A. L. Fradkov), [73] (A. Nedic), [76] (G. Notarstefano), [81] (F. Pasqualetti), [92] (F.

Rossi), [19,20] (F. Bullo), [22,83,117] (C. Cassandras), [97] (Y. Shimizu), [87] (Z. Volkovich), as well as dimensionality reduction control [105] (V. I. Utkin), [98] (L. Fridman). The main alternative approach to the formalization of multi-agent systems is modeling using automata [42] (M. Flasiński), [69] (A. Mohammed), [99] (E. Silvia). However, for modeling cyber-physical systems specifically, the first approach is the most convenient, since changes in the states of cyber-physical agents (for example, unmanned aerial vehicles) occur in the physical world according to the laws of mechanics.

Among the most developed methods of MAS control, two approaches can be distinguished: global control, where each agent is given the same behavior algorithm, independent of the actions of the agents themselves [109] (K. Vamvoudakis), and control with local feedback, where each agent has its own algorithm for interaction with the others, depending on the state of neighboring agents [43] (A. L. Fradkov), [8–11] (N. O. Amelina), [5,7] (K. S. Amelin). In the first case, the control scheme is quite simple to understand and, more importantly, to operate. However, for simplicity we have to sacrifice the variability of the set of reachable states. The theory of control with local feedback is currently not as deeply developed as in the case of global control, but has great potential due to the substantial variability of possible technological solutions. Despite the promising prospects, local control has a big problem: in large-scale systems, if it is necessary to change the control strategy (for example, when adapting to a new road in the problem of balancing transport traffic), you will have to "pointwise" contact each agent individually, which will entail large communication and computing costs on the part of the data center.

In recent years, the popularity of research into the phenomenon of clustering in MAS has been growing [23–25] (E. F. Camacho), [67] (M. Mattioni). The clustering process can occur spontaneously: if the individual goals of the agents coincide, then together they can form a certain structure (subsystem), perceived as a whole. Spontaneous grouping is a very common phenomenon in nature (union of animals into herds [61] (N. E. Leonard), synchronous work of brain areas [1] (H. Acebron (J. Acebron)), [68] (V. I. Sysoev), [94]

(M. Sadilek), and in human and intellectual activity (unification of settlements into states, servers into clusters, discovery of common patterns in data [80]) (J. Oyelade).

Just as clustering occurs in nature, this phenomenon is also common in groups of robots. In fact, clustering can be used to manage entire groups of agents — in this case, decentralization moves from the level of individual agents to the level of interacting clusters. With a large number of agents in the system, it is quite difficult to trace all the variety of interactions between them, but at the same time, clustering allows you to effectively group agents, which allows you to reduce the number of control actions and, thus, simplify the system. Moreover, since individual groups of agents are isolated from others, their trajectories in state space have a sparse representation. This allows you to use data compression methods when transmitting information about the MAS to the data center, for example, to make decisions about changing the local and cluster management strategy.

Thus, the relevance of the cluster management paradigm is supported by the numerous examples of demand in industry and science mentioned above. However, in the main works devoted to the phenomenon of clustering of multi-agent systems, the emphasis is on the connections between agents, which, due to the violation of fully connectedness or weighted connectivity, lead to desynchronization at the subsystem level. In this regard, there is motivation to develop a new method for modeling and effective adaptive control of multi-agent systems, taking into account both internal (networked, local) and external factors in the emergence of clusters.

The goal of the study is research and development of algorithms for adaptive control of complex large-scale multi-agent network dynamic systems at the cluster meso-scale. To achieve this goal, the following tasks were set and solved:

1) develop a model of information and control processes in complex multi-agent networked dynamical systems to identify characteristic patterns in their behavior, which often lead to clusterization within the system;

2) develop a new approach to controlling complex multi-agent networked dynamical systems, based on the synthesis of control action at a new meso-scale, corresponding to clusters emerging in the system;

3) explore data compression methods and develop on their basis an algorithm for compressed representation of a complex multi-agent networked dynamical system with clusters in a reduced-dimensional space to encode the structure of the system without significant losses and transition to the meso-scale.

Research methods. The dissertation employs methods from information theory, control theory, optimization theory, graph theory; randomized algorithms, linear matrix inequalities, and numeric simulations are used.

Main results. The following scientific results were obtained during the work:

1) a new approach to modeling information and control processes in complex multi-agent networked dynamical systems is proposed and justified, describing time-varying clustering in dynamic networks of elementary control objects;

2) developed a method for controlling complex multi-agent networked dynamical systems with clustering, in which the synthesis of control action occurs in a space of reduced dimensionality, the effectiveness of the developed method in comparison with classical approaches was demonstrated;

3) developed an approach to encoding sparse information in complex multi-agent networked dynamical systems with clusters based on the "compression identification" method, and demonstrated the connection between agent clustering and the sparseness of the system representation.

Scientific novelty. All the main scientific results of the dissertation are novel.

Theoretical value and practical significance. The theoretical value of the results lies in the development and substantiation of new methods for modeling and controlling complex network multi-agent systems based on the synthesis of a control action based on a compressed representation of this system in a low-dimensional space; in a new approach to compressing large sparse data arising in large-scale networked systems, and in the connection between dynamic clustering and agent sparseness in the system's shared state space. The randomized method of encoding sparse data, "identification by compression", formed the basis of a new approach to encoding the state of a complex system with clusters, allowing one to form effective observations of the main features of the system and synthesize control with high computational efficiency.

The proposed methods and approaches can be used to solve a number of practical problems. In particular, for effective load balancing in computer networks, managing large-scale networks of autonomous unmanned vehicles, spacecraft and nano-machines, as well as in problems of applied physics, geological and meteorological predictions, modeling of biological swarm systems and social modeling.

Research validation. The results of the dissertation were presented at seminars of the Department of System Programming of the Faculty of Mathematics and Mechanics of St. Petersburg State University, at the international "Summer Student Program at JINR" (Dubna, Russia, July 1-August 10, 2018), conferences Science And Progress (St. Petersburg, Russia, November 12-14, 2018), XIV Workshop on Particle Correlations and Femtoscopy (WPCF 2019) (Dubna, Russia, June 3-7, 2019), 18th National Congress on Cognitive Research, Artificial Intelligence and Neuroinformatics (CAICS 2020) (Moscow, Russia, October 10-16, 2020), XIVth Multiconference on Management Problems (MCPU 2021) (Divnomorskoe village, Gelendzhik, Russia, September 27–October 2, 2021), at the 61st IEEE Conference on Decision and Control (CDC 2022) (Cancun, Mexico, December 6-9, 2022), XXIV All-Russian Conference of Young Scientists "Navigation and Traffic Control" (with international participation) (St. Petersburg, Russia, March 15–18, 2022), 22nd IFAC World Congress (IFAC 2023) (Yokohama, Japan, July 9-14, 2023).

The results of the dissertation were used in grant work RSF 17-72-20045 "Application of machine learning methods and Bayesian Gaussian process in modeling the properties of hadron collisions at high energies", YBN202 0095061 "Compressing Sensing based image processing for improved perceptual quality in extremely low light conditions", RSF 16-19-00057 "Adaptive control with predictive models with a variable state space structure with application to network motion control systems and automation of medical equipment", RSF 21-19-00516 "Multi-agent adaptive control in network dynamic systems with application to groups of robotic devices under uncertainty".

Publication of results. The main research results are reflected in the works [6, 36, 49–51, 95, 106–108]. The applicant has published 9 scientific papers, 4 of which were published in journals indexed in the Web of Science and Scopus databases, 3 in conference proceedings indexed in the Web of Science and Scopus databases, and 2 were published in publications included in the RSCI.

The works [6, 36, 49–51, 95, 106–108] were written in collaboration. In the work [95] D. R. Uzhva is responsible for the development of neural network models for data analysis, conducting and visualizing the results of statistical analysis, co-authors — the general formulation of the problem. In the work [49] D. R. Uzhva is responsible for the proof of the theorem and simulation modeling, the co-authors are responsible for the general formulation of the problem and the choice of solution methods. In the work [51] D. R. Uzhva is responsible for the proof of the theorem on the effective control of nonlinear systems and simulation modeling, the co-authors include the general formulation of the problem and the choice of solution methods. In the work [108] D. R. Uzhva describes the approach to effective management of clusters in complex systems, the co-authors describe the general formulation of the problem, the choice of solution methods. In the work [50] D. R. Uzhva is responsible for the development of the theory of control of complex systems, the formulation and proof of the theorem, the co-authors are the general formulation of the problem. In the work [107] D. R. Uzhva is responsible for the development of an algorithm for encoding big data in a complex network system and simulation modeling, the co-authors are the general formulation of the problem. In the work [6] D. R. Uzhva is responsible for the formulation of clustering models in multi-agent systems, formulations and proofs of theorems on clustering in a nonlinear formulation, co-authors — general formulation of the problem, simulation modeling. In the work [36] D. R. Uzhva is responsible for the theoretical result, the formalization of the problem, the co-authors are the general formulation of the problem and simulation modeling.

Compliance with the specialty passport. The dissertation corresponds to the passport of the scientific specialty 1.2.3. "Theoretical informatics, cybernetics" and the field "6. Mathematical theory of optimal control, including optimal control under conflict conditions".

Structure and volume of the dissertation. The dissertation consists of an introduction, three chapters, a conclusion, a list of references, and 120 sources. The text spans 95 pages and includes 9 figures and 5 tables.

Brief content of the work

The **introduction** substantiates the relevance of the dissertation work, formulates the goal, sets the research objectives and briefly outlines the main results.

In **first chapter** a description of the problem of modeling complex large-scale multi-agent networks under conditions of external disturbances and communication interference is given. Section 1.1 outlines intuitive ideas about complex systems as difficult to predict multi-agent systems (MAS) with a large number of weakly coupled elementary agents. The intuitive foundations of clustering in complex systems are considered as a process of pattern formation, thanks to which it is possible to organize the system and describe it in an effective way without the need to contact each agent directly. In section 1.2, a model of cluster flows is proposed for a formal description of complex systems and the processes of cluster formation in them. The notations used in this work are first introduced, then a model of a continuous dynamic system with disturbances and interference in observations, as well as the introduction of cluster control, is formulated. This system has been discretized for implementation in software agents. Section 1.3 contains conclusions about the first chapter.

In **chapter two** a method is proposed for the formation of a new mesoscopic (cluster) control based on the cluster flow model, applied to the mesoscopic subsystems of the original system united in a coalition. Section 2.1 provides a general classification of control strategies based on the presence of feedback, the achievability of the control goal, as well as a classification of models by control action scale. By analyzing the advantages and disadvantages of the presented classes, the work further uses feedback strategies optimized by the amount of control (labor input), as well as models with meso-scale patterns and "unknown but limited" interference Section 2.2 demonstrates the advantage of the meso-level control strategy over micro- and macro-level strategies. A method for controlling complex multi-agent networked dynamical systems with clustering is proposed. In section 2.3, using the developed methodology for forming meso-scale control, the nonlinear model of Kuramoto oscillators is analyzed. For this model, conditions are presented for the formation of clusters in the system under which cluster management can be introduced. Section 2.4 contains conclusions about the second chapter.

In **chapter three** A universal method "identification by compression" is proposed, which is capable of efficiently encoding and recovering sparse signals. Methods for using this method to identify clusters in a complex system are proposed. Section 3.1 describes the general methodology for "compression identification". Conditions for efficient reconstruction of the original signal from its compressed representation are formulated, and randomized compression strategies are discussed. Section 3.2 demonstrates the connection between sparsity and clustering in a multi-agent system as between the processes of pattern formation in the system state vector. A theorem on transforming the system state vector into a sparse form is formulated and proven. Section 3.3 describes the developed software — a meso-scale control simulator with recognition by compression for generating a compressed representation of the system. The principles of quantization of a system for recognizing clusters are formulated, an assessment of complexity and software requirements are given. In section 3.4, the developed software was tested on the previously analyzed nonlinear model of Kuramoto oscillators. Experiments were carried out to measure the accuracy of cluster determination. In section 3.5, the developed software was tested on a large-scale multi-agent system of 100,000 agents. The limits of noise intensity at which high-quality restoration of the system state is possible are demonstrated. Section 3.6 contains conclusions on the third chapter.

The **conclusion** formulates the main results of the dissertation.

Main scientific results

Below are the main new scientific results of the author with links to the relevant publications and author's contribution:

- a new approach to modeling information and control processes in complex multi-agent networked dynamical systems is proposed and justified, describing time-varying clustering in dynamic networks of elementary control objects:
  - model of information-control processes in complex multi-agent network dynamic systems with clustering arising in them, see [6, 36, 49–51, 107, 108] (the personal contribution of the author of the dissertation is at least 80%);
  - modeling and analysis of nonlinear multi-agent network dynamic systems, see [6, 49, 51] (all analytical and numerical calculations were performed personally by the author of the dissertation, the total contribution is at least 80%);

- developed a method for controlling complex multi-agent networked dynamical systems with clustering, in which the synthesis of control action occurs in a space of reduced dimensionality, the effectiveness of the developed method in comparison with classical approaches was demonstrated:
  - meso-scale method of adaptive control of complex multi-agent networked dynamical systems with clustering, see [6, 36, 50, 107, 108] (the personal contribution of the author of the dissertation is at least 80%);
  - optimal meso-scale control of complex multi-agent network dynamic systems, see [6, 107] (all analytical calculations were performed personally by the author of the dissertation, the total contribution is at least 80%);
- developed an approach to encoding sparse information in complex multi-agent networked dynamical systems with clusters based on the "compression identification" method, and demonstrated the connection between agent clustering and the sparseness of the system representation:
  - approach to encoding sparse information in complex multi-agent clustered systems based on the "recognition by compression" method, see [6, 36, 51, 106–108] (the personal contribution of the author of the dissertation is at least 80%);
  - simulations of complex multi-agent network dynamic systems with clustering using sparse agent data coding, see [49, 51, 107] (all analytical and numerical calculations were performed personally by the author of the dissertation, the total contribution is at least 80%);
  - coding of sparse magnetometric data based on the "recognition by compression" method, see [106] (the contribution of the author of the dissertation is 100%);

– classification of events in physics experiments with high-energy hadron collisions using deep neural networks, see [95] (all numerical calculations were performed personally by the author of the dissertation).

Findings and arguments of the dissertation to be defended

Below are the main findings and arguments submitted by the author for defense:

- model of information-control processes in complex multi-agent networked dynamical systems with clustering arising in them; features of clustering in linear and nonlinear multi-agent dynamic systems; conditions for the emergence of a cluster structure in complex systems;

- meso-scale paradigm of adaptive control of complex multi-agent networked dynamical systems with clustering; optimal adaptive control of clustered multi-agent systems;

- relationship between sparseness of data in models describing complex multi-agent networked dynamical systems and clustering; analytical demonstration of the possibility of encoding complex clustered systems using sparse data compression methods; numerical demonstration of the efficiency of encoding complex systems with meso-scale control.

Chapter 1

Model for describing information and control processes in complex systems

To model complex multi-agent networked dynamical systems, such systems must first be defined. In their definition lies the problem of their description, which motivates the study of means for their formalization.

1.1 The concept of a complex multi-agent networked dynamical system

When solving problems of automating life processes, there is always a need to make predictions about the future behavior of our interpretations of these processes. Without the assumption of the presence of cause-and-effect relationships between natural phenomena, it would be impossible to make correct predictions with a high probability, and therefore, the apparatus of dynamic systems constructed from meaningful assumptions about the dependence of events from the future on events occurring in the past and present is used throughout.

Thus, as we study patterns in the dynamics of various natural and artificial systems, we find that analyzing them to predict future behavior can be extremely difficult. For example, consider the problem of weather forecasting. In most cases, it is safe to say that the weather one hour from now $W_{\text{future}} = T(W_{\text{present}})$, where T is a transformation formalizing the cause-and-effect relationship, will be very similar to the current weather W_{present}, allowing us to view future events as transformed events occurring at the present moment in time. However, as we move into the future, the transformations T will overlap: $T^n(W_{\text{present}})$, where T has been reapplied to W_{present} n times, corresponding to a prediction n hours ahead. In this case,

$$W_{\text{future}} = T^n(W_{\text{present}})$$

becomes less and less like the present. Moreover, if we "choose" current weather $\widetilde{W}_{\text{present}}$ to be slightly different from W_{present} (i.e. $\widetilde{W}_{\text{present}} \approx W_{\text{present}}$), then for large values of $n \gg 1$ (e.g., when predicting the weather a week ahead), the same series of transformations will lead to some future weather

$$\widetilde{W}_{\text{future}} = T^n(\widetilde{W}_{\text{present}}),$$

which may be very different from W_{future}. In other words, $\widetilde{W}_{\text{present}} \approx W_{\text{present}}$ does not necessarily lead to $\widetilde{W}_{\text{future}} \approx W_{\text{future}}$, which is observed in practice: the longer the time horizon of weather prediction, the less accurate it will be. Such behavior, sensitive to initial conditions, is associated with the nature of T, determined by the interaction between various internal components and external disturbances (e.g., air molecules, the Sun) in the climate. It is worth noting that the interactions themselves can be quite simple: for example, today physics is able to describe the interactions between air molecules quite accurately. However, the number of components and the relationships between them, despite the simplicity of the interaction rules, makes it impossible to obtain and calculate an accurate model of T over a sufficiently long time interval using limited resources. In turn, the emergent behavior mentioned in the introduction, which consists in the irreducibility of the resulting observable behavior (the weather at the current moment in time) to the behavior of the components of the system and its surrounding parts (air molecules, the Sun, the geological and ecological situation on the planet), is also characteristic of the described system. Thus, the problematic of describing the dynamics of the weather system stems from the impossibility of accurately describing even its static picture at the current moment in time. Further in the dissertation, such systems are called "complex multi-agent networked dynamical systems": applying the principles of multi-agent systems, multi-agent control and data coding theory, a framework for efficient management of them is formulated. This framework bypasses the "curse of dimensionality" typical for them, using new rules of cluster control and simplification of models due to the sparse representation in the state space caused by clustering. The current chapter is about the first necessary step: building a model of complex multi-agent networked dynamical systems.

1.1.1 Modeling of complex multi-agent networked dynamical systems

As noted above, the method of modeling using the apparatus of dynamic systems allows one to intuitively clearly imagine systems developing over time according to cause-and-effect relationships. This type of modeling can be divided into three classes:

1. Discrete-time simulation [86] (C. Ravazzi), [99] (E. Silva).

2. Continuous time simulation [101] (S. H. Strogatz), [12] (V.I. Arnold), [43] (A. L. Fradkov).

3. Modeling using the apparatus of field theory (with a continuum index of agents) [113] (V. Vilasini).

According to the first class, a complex system consists of a finite number of elementary autonomous units (*agents*), each of which has its own individual state, evolving iteratively over time. The corresponding system model can be expressed by a difference equation; As an example, consider the autoregressive model:

$$x_t^i = c + \sum_{k=1}^{K} \theta_k x_{t-k}^i + w_t^i, \tag{1.1}$$

where x_t^i is the state of agent i at time t, c is a constant, $\theta_1, \ldots, \theta_p$ is the system parameters and w_t^i — external disturbance. When $p = 1$, the result is a so-called Markov process, subject to the "out of memory" problem, since x_t^i depends only on its previous iteration x_{t-1}^i. The model (1.1) can be expanded by adding nonlinearity and relationships between several neighboring agents. In addition, we can consider probabilistic versions of automata models based on the formalism of Markov chains [99] (E. Silva). The class of discrete modeling can also include Poincare and Lorentz "maps" [101] (S. H. Strogatz), despite the fact that such models are traditionally associated with dynamical systems. Thus, we can say that the discrete approach is convenient for describing stochastic discrete processes, which is relevant in the context of large-scale systems with a large number of digital agents.

Despite the fact that discrete modeling uses an iterative approach to describe the evolution of a system and, accordingly, such models have a simple implementation in software environments, cyber-physical systems can often depend continuously on time. In this case, the dynamics are modeled by a system of ordinary differential equations:

$$\dot{x}^i(t) = f_i(x^i(t), u^i(t), \xi^i(t)), \tag{1.2}$$

where $x^i(t) \in \mathbb{R}^{n_i}$ — agent state vector $i \in \mathcal{N} = \{1, \ldots, N\}$ (n_i — number variables necessary to describe the agent's state), $u^i(t)$ is a local control action, and $\xi^i(t) \in \mathbb{R}^{m_i}$ is a disturbance. Moreover, this method of modeling systems is not subject to the above-mentioned phenomenon of memory shortage due to greater freedom in choosing the time intervals over which the state function $x^i(t)$ is defined. Continuous dynamic systems are more convenient for analytical research due to the well-developed apparatus of mathematical analysis.

Finally, the field theory approach can be considered the most general (derived from the equation (1.2) in the $N \to \infty$ limit) since it is able to model the continuum cardinality of agents:

$$\dot{x}(\lambda, t) = f(\lambda, x(\lambda, t), u(\lambda, t), w(\lambda, t)), \tag{1.3}$$

where the agents are now “renumbered” (or rather localized) at the point $\lambda \in \mathbb{R}$ or its equivalent subset (that is, also with continuum cardinality).

Modern cybernetics is mainly focused on modeling discrete and continuous dynamic systems, while modeling based on field theory apparatus, as in the equation (1.3), still has few applications, and theoretical methods for analyzing field models are relatively poorly developed and less accessible to researchers and engineers in the field of control theory than the first two categories of models. Consequently, the main emphasis will be placed further on dynamic systems with a large, but finite and integer number of agents.

1.1.2 Clustering in complex systems

In the equation (1.2), the control input u regulates the behavior of the system by establishing rules for changing the state of the agent depending on the states of neighboring agents and other environmental factors. Artificial complex systems are often required to change their global state in a controlled manner so that the rules u lead to a given global goal. Thus, before moving directly to systems management, it is worth exploring possible management goals. According to [43] (A. L. Fradkov), five types of goals can be distinguished:

1. Stabilization (reduction of all states of agents x to a constant vector x_*):

$$\lim_{x \to +\infty} x(t) = x_*.$$

2. Tracking (reducing the states of agents to the function $x_*(t)$):

$$\lim_{x \to +\infty} |x(t) - x_*(t)| = 0.$$

3. Excitation of oscillations:

$$\lim_{x \to +\infty} G(x(t)) = G_*$$

   for some scalar function $G(x)$.

4. Synchronization:

$$\lim_{x \to +\infty} |x^i(t) - x^j(t)| = 0.$$

5. Modifications of the limit set (qualitative changes in the system, for example, modifications of the types of bifurcations).

However, this classification is applicable primarily to fairly simple systems, primarily single-element ones. As for multi-agent systems, the goal of synchronization is usually of greatest interest, since it is associated with the emergence of patterns that open up opportunities to reduce the dimension of the system and, as a result, complexity. Indeed, if all agent states converge into one synchronous dynamic trajectory, the entire system can be controlled as a single group of identically behaving components, which requires only one control input instead of N for each agent.

Although the prospect of controlling a complex system using synchronization may seem tempting, in practice complete synchronization almost always cannot be achieved. Most often, in many artificial and natural complex systems, the so-called *cluster synchronization* (also called clustering) is observed, according to which agents are synchronized in groups: system components from the same group are synchronized, but those belonging to different groups are not. For example, cluster synchronization occurs in human brain activity, assuming that the brain can be accurately represented by a nonlinear coupled oscillator model [1] (J. Acebron).

As noted in the introduction, cluster synchronization mainly occurs in systems with incomplete connectivity between agents and due to external disturbances that can affect the connectivity and state of agents. During clustering, a set of synchronous manifolds is formed (in reality or in the model) on which the trajectories of agents fall. Let us denote the number of such cluster varieties as s, then in practice the following relationship between the number of agents N and the number of clusters s is often true:

$$N \gg s > 1. \tag{1.4}$$

The empirical inequality (1.4) motivates the need to study the phenomenon of clustering in complex systems in order to develop simple strategies for system control at the cluster level. To manage complex multi-agent systems with significant uncertainties and non-trivial network topology leading to clustering, a model of *cluster flows* is further formulated, the foundations of which are outlined in [84] (O. N. Granichin, A. V. Proskurnikov) . The model is based on the idea of combining equations of dynamics of agents from one subsystem into an equation of the aggregated state of some characteristic of the subsystem: for example, cluster centroids.

1.2 Cluster flows model

Next, we denote the vectors in bold or Greek letters (if this is a vector of system parameters, noise or disturbances), for example $\mathbf{x}$ or θ; whereas the indices of vectors, matrices, and ordered sets are represented as superscripts, as in $\mathbf{x}^i$, A^{ij}, or $\mathcal{N}^i$. Time indices are written as subscripts for discrete systems (e.g. $\mathbf{x}_t$) and in brackets for continuous ones ($\mathbf{x}(t)$). A positive definite matrix A is denoted by $A \succ 0$, and a positive semidefinite matrix B is denoted by $B \succeq 0$. We denote the identity matrix as $\mathbf{I}$, whose dimensions are chosen according to its application, or $\mathbf{I}_{n\times n}$ if the dimension must be specified for clarity. We also denote the null matrix as either $\mathbf{O}$ or $\mathbf{O}_{n\times n}$ if the dimension must be specified. In the same way, the unit vector is denoted as $\mathbf{1}$ or $\mathbf{1}_n$ and the zero vector is denoted as $\mathbf{0}$ or $\mathbf{0}_n$.. Everywhere below $|\cdot|$ — the cardinality of a set or the absolute value of a number. The notation $\otimes$ represents the Kronecker product. The block matrix A consists of blocks denoted by $[A^{ij}]^{i,j\in\mathcal{N}}$, where $\mathcal{N}$ is the corresponding set of indices. A block vector $\mathbf{x}$ consists of its lower-dimensional projections onto a standard basis, denoted by $[\mathbf{x}^i]^{i\in\mathcal{N}} = \mathrm{col}(\mathbf{x}^1,\ldots,\mathbf{x}^{|\mathcal{N}|})$. The spectral norm of the matrix A is denoted as $\|A\|$.

1.2.1 Concepts from graph theory

Let $\mathcal{G} = (\mathcal{N},\mathcal{E})$ be a directed graph, where $\mathcal{N} = \{1,\ldots,N\}$ is a set of vertices, and $\mathcal{E} \subseteq \mathcal{N}\times\mathcal{N}$ is a set of edges. Let $(j,i) \in \mathcal{E}$ if there is a directed edge from node j to node i. The latter means that node j can transmit data to node i. For node $i \in \mathcal{N}$ a set of neighbors is defined as $\mathcal{N}^i = \{j \in \mathcal{N} \mid (j,i) \in \mathcal{E}\}$. We denote the indegree of a vertex $i \in \mathcal{N}$ is equal to $|\mathcal{N}^i|$, and $i \notin \mathcal{N}^i$. It is also assumed that the topology of the graph can change over time, which is modeled by a subgraph $\mathcal{G}$, that is, $\bar{\mathcal{G}}_t = (\mathcal{N}_t,\mathcal{E}_t)$, where $\mathcal{N}_t \subseteq \mathcal{N}$ and $\mathcal{E}_t \subseteq \mathcal{E}$. Let $b_t^{ij} > 0$ be the weight of an edge $(j,i) \in \mathcal{E}_t$, and $b_t^{ij} = 0$ whenever $(j,i) \notin \mathcal{E}_t$. Let $\mathcal{B}_t = [b_t^{ij}]$ be the weighted adjacency matrix (or simply the connectivity matrix) of the graph $\bar{\mathcal{G}}_t$. We will assume that the weight b_t^{ij} is the cost of transmitting data through the edge $(j,i) \in \mathcal{E}_t$. The weighted indegree $i \in \mathcal{N}$ is defined as $\deg_i^+(\mathcal{B}_t) = \sum_{j=1}^n b_t^{ij}$, the maximum weighted indegree among all nodes contained in the graph $\bar{\mathcal{G}}_t$ as $\deg_{\max}^+(\mathcal{B}_t)$. We also denote the diagonal matrix of indegrees of all nodes as $\mathcal{D}(\mathcal{B}_t) = \mathrm{diag}_n(\mathrm{col}(\deg_1^+(\mathcal{B}_t),\ldots,\deg_n^+(\mathcal{B}_t)))$. Then $\mathcal{L}(\mathcal{B}_t) = \mathcal{D}(\mathcal{B}_t) - \mathcal{B}_t$ is the "Laplacian" of the graph $\bar{\mathcal{G}}_t$.

Similarly, for continuous systems, the concepts of a time-varying subgraph $\bar{\mathcal{G}}(t) = (\mathcal{N}(t)),\mathcal{E}(t)))$ are introduced.

1.2.2 Modeling agent group dynamics

As before, we denote the set of agents as $\mathcal{N} = \{1, \dots, N\}$, where N is their number. In classical approaches to MAS control, the dynamics of a set of interacting agents is characterized by the following system of differential equations:

$$\dot{\mathbf{x}}^i(t) = \mathbf{f}^i(t, \mathbf{x}^i(t), \mathbf{u}^i(t), \mathbf{U}(t), \xi^i(t)), \tag{1.5}$$

where $\mathbf{x}^i(t) \in \mathbb{R}^{n^i}$ is the state vector of agent $i \in \mathcal{N}$; $\mathbf{u}^i(t)$ — microscopic control, describing how local interactions between agents affect their state; $\mathbf{U}(t)$ is a macroscopic control that affects all agents equally; $\xi^i(t) \in \mathbb{R}^{m^i}$ — external disturbance. Next, the model (1.5) will be modified by including a new control action acting on individual clusters in the system.

In practice, it is often not possible to extract the agent's state $\mathbf{x}^i(t)$ in its pure form. For example, imagine a car system, where $\mathbf{x}^i(t)$ will be its coordinates and speed in $\mathbb{R}^6$, as well as the entire set of states of its parts: engine temperature, oil pressure, etc. Despite the variety of measuring instruments available to the driver, it is not possible to measure the condition of absolutely all parts, and therefore the driver, as a controller that forms the control input (by pressing the pedal and rotating the steering wheel), acts according to only part of these measurements. In models, measurement is represented by agent output functions.

Definition 1 (System output). *The function $\mathbf{g}^i(\mathbf{x}^j(t), \eta^{ji}(t))$ is called the output of agent j, received by agent i, if $\mathbf{g}^i : \mathbb{R}^{n^j} \times \mathbb{R}^{m^{ji}} \mapsto \mathbb{R}^l$, where l does not depend on i and j; $\eta^{ji}(t)$ — communication interference between agents i and j.*

According to the ideas contained in [84] (O. N. Granichin, A. V. Proskurnikov), in multi-agent systems, it is useful to introduce two outputs: one is used to ensure synchronization of agents if necessary, the other is used to monitor the state of an agent. The first is used for internal communication protocols between network nodes, the second is used for external or internal observation of agent's state, according to which a cluster pattern is formed. Let $\mathbf{y}^{ji}(t) = \mathbf{g}^i(\mathbf{x}^j(t), \eta^{ji}(t))$ $j \in \mathcal{N}^i(t)$ be outputs agent j from neighborhood i used for agent observation. By "communication" between agents it is assumed that the state of an agent i (at time t) is changed based on the outputs $\mathbf{y}^{ji}(t)$. In practice, these outputs can be transmitted from j to i or displayed by agent j to the environment, being then recognized by agent i. Mathematically, the transfer rules are defined in the local control $\mathbf{u}^i$ (see equation (1.5)):

$$\mathbf{u}^i(t) \equiv \mathbf{u}^i\left(t, \{\mathbf{y}^{ji}(t)\}_{j \in \mathcal{N}^i(t) \cup i}\right), \tag{1.6}$$

where $\mathbf{u}^i(\cdot)$ is a vector function of outputs $\mathbf{y}^{ji}(t), j \in \mathcal{N}^i(t) \cup i$. Equation (1.6) is called *communication protocol* in the sense that it contains rules for controlling $\mathbf{u}^i(t)$ based on the outputs $y^{ji}(t)$ received by agent i. Thus, we formulate the following definition of a multi-agent networked dynamical system:

D e f i n i t i o n 2 (Multi-agent networked dynamical system)**.** *A triple of objects consisting of: 1) a family of agents (see equation* (1.5)*); 2) an interaction graph $\mathcal{G}$ with dynamically changing topology of subgraphs $\bar{\mathcal{G}}(t)$ and 3) the communication protocol defined in equation* (1.6) *is called multi-agent networked dynamical system.*

Further in the dissertation, multi-agent networked dynamical systems are denoted by the set of their agents — as $\mathcal{N}$, corresponding to the set of agents of the system.

Let $\mathbf{z}^i(t) = \mathbf{h}^i(x^i(t), \eta^i(t))$ be the output i used to measure synchronization, $\eta^i(t)$ is communication interference between agent i and an external measuring device. As described in the introduction, agents can synchronize not only globally, but also across different clusters. In modern control theory, as a rule, clustering is modeled by partitioning a set of agents $\mathcal{N}$ into a number of disjoint subsets. Such subsets are usually called *coalitions* [24] (E. Camacho).

D e f i n i t i o n 3 (Coalition structure)**.** *Time-dependent coalition* *structure over the set* $\mathcal{N}$*, denoted by* $\mathcal{CS}(t) = \{\mathcal{C}^1(t), \ldots, \mathcal{C}^{s(t)}(t)\}$*, where* $1 \leq s(t) \leq N$ *is the number of coalitions, is a partition of* $\mathcal{N}$ *into disjoint subsets, satisfying the following conditions:*

- *coalition* $\mathcal{C}(t) \in \mathcal{CS}(t)$ *is not empty:* $\mathcal{C}(t) \notin \emptyset$*;*
- *coalitions cover the entire system:* $\cup_{\mathcal{C}(t) \in \mathcal{CS}(t)} \mathcal{C}(t) = \mathcal{N}$*;*
- *coalitions do not intersect:* $\forall \mathcal{C}(t), \overline{\mathcal{C}}(t) \in \mathcal{CS}(t)$*,* $\mathcal{C}(t) \neq \overline{\mathcal{C}}(t) : \mathcal{C}(t) \cap \overline{\mathcal{C}}(t) = \emptyset$*.*

The number of agents in the coalition $\mathcal{C}^\alpha(t)$ *is denoted as* $|\mathcal{C}^\alpha(t)|$*. Everywhere below, coalitions are indexed by Greek letters, for example,* $\alpha \in \mathcal{S}(t) = \{1, \ldots, s(t)\}$*.*

The novelty of the result of the dissertation lies in the effective control of cluster structures based on the aggregated states of clusters. To form such a control, the standard definition of the coalition structure is not enough, and therefore the definition of cluster synchronization from [84] (O. N. Granichin, A. V. Proskurnikov) is presented below.

Definition 4 (Cluster synchronization)**.** *Let* $\Delta^{ij}(t) = \|\mathbf{z}^i(t) - \mathbf{z}^j(t)\|$ *denote the difference of states* $\mathbf{z}^i(t), \mathbf{z}^j(t)$*, where* $\| \cdot \|$ *— norm equivalent to the* ℓ_2*-norm. Let also* $\mathcal{CS}$ *be a coalition structure over the set* $\mathcal{N}$*. Then the multi-agent network experiences cluster synchronization at* $\delta \geq \varepsilon \geq 0$ *if for* $\mathcal{C}, \overline{\mathcal{C}} \in \mathcal{CS}, \mathcal{C} \neq \overline{\mathcal{C}}$ *is fair*

1. $\Delta^{ij} \leq \varepsilon$ *for all* $i, j \in \mathcal{C}$*;*
2. $\Delta^{ij} > \delta$ *for all* $i \in \mathcal{C}$*,* $j \in \overline{\mathcal{C}}$*.*

For $\delta = \epsilon = 0$*, "full" cluster synchronization takes place.*

In the dissertation, it is proposed to modify this definition, since it assumes the ability to find unclassified points $\mathbf{z}^j$ (outliers) for which $\varepsilon < \Delta^{ij} \leq$

δ for any $i \neq j$ and $\delta > \varepsilon$. Such points will never be assigned to any coalition $\mathcal{C}$, but at the same time, by Definition 3 all agents are assigned to the coalition. Thus, the Definitions of 3 and 4 from various sources contradict each other. It is also worth noting that by definition, a cluster is a structure in which *even the most distant* agents are located at a distance of at least ε. Therefore, the communication graph between agents within a cluster, defined by their respective distance Δ^{ij} (which can be related to the communication radius as $(j, i) \in \mathcal{E} \iff \Delta^{ij} \leq \varepsilon$) will be fully connected. In general, such a feature will require solving a complex computational clique problem (search of complete subgraphs) [58] (R. Karp) to successfully identify clusters. Thus, taking into account the above-mentioned problems, the cluster flow model proposes a new definition of cluster synchronization that does not contradict the Definition of coalition structure 3.

Definition 5 (Cluster synchronization)**.** *Let* $\Delta^{ij}(t) = \|\mathbf{z}^i(t) - \mathbf{z}^j(t)\|$ *denote the difference of observations* $\mathbf{z}^i(t), \mathbf{z}^j(t)$*, where* $\|\cdot\|$ *— norm equivalent to the* ℓ_2*-norm. Let also* $\mathcal{C}$ *be a coalition structure over the set* $\mathcal{N}$*. The graph* $\mathcal{G}_{\mathcal{CS}(t)} = (\mathcal{N}, \mathcal{E}_{\mathcal{CS}(t)})$ *is assigned the edges* $(i, j) \in \mathcal{E}_{\mathcal{CS}(t)} \iff \Delta(t)^{ij} \leq r^i(t)$ *for some* $r^i(t) \geq 0$ *(called the communication radius) and nodes* $i \in \mathcal{N}$*. Then the multi-agent network experiences cluster synchronization if for* $\mathcal{C}(t), \overline{\mathcal{C}}(t) \in \mathcal{CS}(t), \mathcal{C}(t) \neq \overline{\mathcal{C}}(t)$ *is true:*

1. *All* $i, j \in \mathcal{C}(t)$ *are in one connected component* $\mathcal{G}_{\mathcal{CS}(t)}$*;*

2. *All* $i \in \mathcal{C}(t)$ *and* $j \in \overline{\mathcal{C}}(t)$ *are in different connected components* $\mathcal{G}_{\mathcal{CS}(t)}$*.*

For $r^i(t) = 0$*, "full" cluster synchronization takes place.*

In other words, any closed $r^i(t)$-ball around $\mathbf{z}^i(t)$ either does not contain points (then the corresponding coalition $\mathcal{C}(t) = \{i\}$ trivial), or contains some or all observation points of other agents within the same coalition $\mathcal{C}(t)$. In this case, full synchronization means the collection of all agents within the cluster into one point. This approach is also similar to density-based clustering methods such as Density-Based Spatial Clustering for Noisy Applications (DBSCAN) [39] (M. Ester), and therefore has similar range of applications.

Finally, a new control input

$$\bar{\mathbf{u}}^{\alpha}(t) \equiv \bar{\mathbf{u}}^{\alpha}\left(t, \bar{\mathbf{z}}^{\alpha}(t)\right) \tag{1.7}$$

can be the added into the system described by the equation (1.5), where $\bar{\mathbf{z}}^{\alpha}(t)$ is some aggregated (generalized) state of the cluster α, and $i \in \mathcal{C}^{\alpha}(t)$. For example, the average value of the outputs $\mathbf{z}$ in the coalition $\mathcal{C}^{\alpha}(t)$ can be used as an aggregated state:

$$\bar{\mathbf{x}}^{\alpha}(t) = \frac{1}{|\mathcal{C}^{\alpha}(t)|} \sum_{i \in \mathcal{C}^{\alpha}(t)} \mathbf{x}^{i}(t).$$

Equation (1.7) allows you to formulate control for $s \ll N$ clusters, which applies equally to all agents in a single cluster. Thus, a new system is obtained, which is used further within the framework of the cluster flow model:

$$\dot{\mathbf{x}}^{i}(t) = \mathbf{f}^{i}(t, \mathbf{x}^{i}(t), \mathbf{u}^{i}(t), \bar{\mathbf{u}}^{\alpha}(t), \mathbf{U}(t), \xi^{i}(t)) \tag{1.8}$$

with added “cluster” control $\bar{\mathbf{u}}^{\alpha}(t)$. Methods for synthesizing the proposed control input for clusters will be discussed in the second chapter of the dissertation.

1.2.3 Discrete multi-agent systems

Equation (1.8) represents the general continuous case, but discrete systems:

$$\mathbf{x}^{i}_{t+1} = \mathbf{f}^{i}_{t}(\mathbf{x}^{i}_{t}, \mathbf{u}^{i}_{t}, \mathbf{U}^{i}_{t}, \xi^{i}_{t+1}), \tag{1.9}$$

in which the agent’s state changes iteratively. In this case, the disturbance ξ^{i}_{t+1} is modeled as a variable that is ahead of the current state and control, which reflects the essence of the disturbance as an unpredictable interference in the operation of the network. All the above definitions are also valid for discrete systems: to reformulate it is enough to include the time dependence in the index of a function or variable.

1.3 Summary

The first chapter proposes a cluster flow model to describe clustering in complex multi-agent systems. The practical value of such a model lies in the potential simplification of the description of a complex system by moving to a low-dimensional space, where some aggregated characteristics of clusters will be used as system states, which will be described in the third chapter. A formal description of clustering in the system is proposed as a time-varying partition of the MAS into subsystems.

Further, in the second chapter, the cluster flow model will be used to formulate and solve control problems, as well as demonstrate the advantages of the new cluster control paradigm over more conventional ones using the example of models of the same system, but in different dimensions.

Chapter 2

Control of complex multi-agent networked dynamic systems at the cluster meso-scale

The previous chapter described various types of goals for managing complex systems, as well as a cluster flow model based on the goal of cluster synchronization. The current chapter reveals possible approaches to the synthesis of cluster control inputs for both classical linear systems and nonlinear ones using the example of a model of coupled oscillators. First, the general principles of synthesis of optimal control for classical linear systems are revealed, then a modification of the classical control theory is proposed using a model of cluster flows and cluster control.

2.1 Classification of control strategies

To develop a cluster control strategy, it is initially necessary to analyze existing classes of strategies to correctly demonstrate the advantages of the proposed approach.

2.1.1 Classification by the presence of feedback

One of the simplest, yet widely used in practice, ways to model and implement the control action $\mathbf{u}$ is to construct the corresponding function $\mathbf{u}^i(t)$ for each agent i, which depends only on time. As an example, consider a linear model without disturbances [57] (R. E. Kalman):

$$\dot{\mathbf{x}}^i(t) = A^i\mathbf{x}^i(t) + B^i\mathbf{u}^i(t), \tag{2.1}$$

where A^i and B^i are state transition and control transformation matrices, and $\mathbf{u}^i(t) \not\equiv 0$. By introducing such a control function $\mathbf{u}^i(t)$, we obtain a heterogeneous system whose state $\mathbf{x}^i(t)$ changes independently of any functions of its current state, except linear. In what follows, this approach to control will be called a program or open-loop strategy to emphasize its independence from the current state of the system.

Program strategies are extremely simple to implement because they can be calculated in advance without burdening the system with additional calculations to correct the direction to the terminal state. However, often, especially for synchronizing multi-agent systems, the dependence of the control input $\mathbf{u}$ only on time, as in the equation (2.1), turns out to be insufficient. For example, local voting [8] (N. O. Amelina) and nonlinear models of Kuramoto oscillators [1] (J. Acebron), [94] (M. Sadilek), discussed in detail later in the current chapter, demonstrate useful and quite complex behavior that stimulates synchronization in the system due to the dependence of control inputs on the state of the system, even despite the simplicity of the control functions themselves. In this regard, it is further proposed to consider control strategies with *feedback* (thereby emphasizing the dependence of the control action on the angets' states) as a class of functions of the form

$$\mathbf{u}^i\left(t, \mathbf{x}^i(t), \{\mathbf{x}^j(t)\}_{j\in\mathcal{N}^i(t)\cup i}\right),$$

depending, in addition to time, also on the states of the agent i and the set of its neighbors.

As was mentioned in the first chapter (see equation (1.6)), it is often impossible to directly know the state of agents, and therefore observational feedback control can be identified as a separate class:

$$\mathbf{u}^i\left(t, \mathbf{x}^i(t), \{\mathbf{y}^{ji}(t)\}_{j\in\mathcal{N}^i(t)\cup i}\right).$$

Thus, three classes of control strategies are formulated, varying in the way the control function itself is modeled.

1. Program strategy.
2. Strategy with state feedback.
3. Strategy with observational feedback.

2.1.2 Classification by feasibility

Strategies can also be classified in terms of the *feasibility* of the desired control action. Indeed, it is impossible to apply unlimited force to instantly achieve the desired state of the system; such a force will always be limited by the capabilities of either the influencing object or the capacity of the communication channel between the controller and the system: $|\mathbf{u}| < \infty$. In practice, similar reasoning is also true for the states of the system $|\mathbf{x}(t)| < \infty$ and observations $|\mathbf{y}(\mathbf{x}(t))| < \infty$.

For the convenience of analytical derivation of solutions, stronger restrictions are often considered: for example, it is often assumed that the control inputs belong to the class of Lipschitz continuous functions [44] (G. Galbraith). A function $x(t)$ is said to be Lipschitz continuous if there exists a constant $K > 0$ (the so-called Lipschitz constant) such that

$$|\mathbf{x}(t_1) - \mathbf{x}(t_2)| \le K|t_1 - t_2| \tag{2.2}$$

for all real-valued t_1 and t_2 (or for all t_1 and t_2 on the time interval under consideration). From the equation (2.2) we can conclude that state- or observation-dependent Lipschitz continuous control inputs $\mathbf{u}$ will be limited by the magnitude of these states or observations, which leads to a simplification of many proofs in the control theory, since, as will be seen later, quality functionals often include a combination of a control function and a state (or observation).

In other cases (for example, control action optimization in a Hilbert space), $\mathbf{x}$, $\mathbf{y}$, or $\mathbf{u}$ may be required to be bounded in the sense of $L_p(0,T)$ norm:

$$\int_0^T |\mathbf{u}|^p dt < \infty$$

to control $\mathbf{u}$, where $T > 0$—the so-called time horizon of system control, which can be either finite or infinite, depending on the task (in the infinite case, you should be especially careful to ensure the convergence integral).

Finally, methods of stability analysis and optimization for control synthesis have recently attracted great interest. When analyzing stability, the task is to search for control classes $\mathbf{u}$ that lead to stable states of the system (or the task is to classify input signals $\mathbf{u}$ according to their ability to bring the system to a stabilized state) [64] (A. M. Lyapunov). At the same time, when solving optimization problems, a search is carried out for those $\mathbf{u}$ that would most quickly lead the system to a stable variety [57] (R. E. Kalman), [32] (J. Doyle) under the above-described restrictions on the classes of control functions. Despite the fact that these two branches of research into the control of complex systems were initially quite clearly separated, they are now closely related to each other. Moreover, today the prospects for solving many complex problems of optimal and, in particular, stabilizing control are opening up thanks to sufficient computing power available for numerical optimization methods, which are the only possible solution to the optimization problem for complex systems that are difficult to analyze analytically.

To solve the optimal control problem, quality functionals $J(u)$ are formulated, depending on the state (or observation) and control. Minimizing the distance to the target together with optimization or bounding of the applied effort leads to an *optimal control strategy*. In a situation of limited resources, as a rule, in practice one of the following types of $J(u)$ is minimized to synthesize an optimal control strategy.

1. Time performance cost when control input is limited:

$$J_1(u) = \operatorname*{argmin}_{\mathbf{u}} T(\mathbf{y}, \mathbf{u}),$$

   where T is the total time spent as a function of monitoring the state of the system $\mathbf{y}$ and the limited control input $\mathbf{u}$.

2. The magnitude of the control input expressing the "expended effort", given that time is fixed and limited:

$$J_2(u) = \operatorname*{argmin}_{\mathbf{u}} \int_0^T \phi(t, \mathbf{u}(y))dt,$$

   where ϕ is some function of the control input $\mathbf{u}$ and time t.

3. The values of the control input and output together:

$$J_3(u) = \operatorname*{argmin}_{\mathbf{u}} \int_0^T \phi(t, \mathbf{y}(t), \mathbf{u}(y))dt, \tag{2.3}$$

   where ϕ is some function of observing the state of the system $\mathbf{y}$, the control input $\mathbf{u}$ and time t.

In the dissertation, further the third type (2.3) is considered, in which resource costs are optimized through control inputs and outputs. As a control goal, we choose to bring the state to the origin and assume that the system is controllable (satisfies the Kalman criterion), and also $\mathbf{x}(T) = \mathbf{y}(\mathbf{x}(T)) = 0$. For a system of one agent it from the classical control theory is known that the control action $\mathbf{u}(\mathbf{y}) = -K\mathbf{y}(t)$ with observation feedback with the transformation matrix K (the search for which is the formal optimization goal) is optimal. It is easy to verify this intuitively by substituting the resulting control function into the functional J_2: such a function will iteratively bring the state closer to zero by subtracting the observation vector correlating with it.

2.1.3 Classification by multiagent subsystem scale

The first chapter of the dissertation proposes a cluster flow model that combines three classes of scale control strategies:

1. Local (micro-scale, microscopic) control, different for each agent.
2. *Cluster (meso-scale, mesoscopic)* control, different for all individual clusters.
3. Global (macro-scale, macroscopic) control, the same for all agents in the system.

For the above classification of control by scale of action, a feedback scheme is proposed (see Fig. 2.1). In this diagram, the action of the control inputs is demonstrated in parallel: the system is simultaneously affected by three controls with different scales. However, as noted earlier, in practice it is often sufficient to use one type of control due to resource limitations, while on a micro scale control turns out to be accurate, but resource-intensive; at the meso-scale—presumably optimal (in terms of the trade-off between resource costs and accuracy in achieving the target state) with a small number of clusters; at the macro-scale—easy to implement, but crude, since it is not capable of making structural changes to the system.

Local and global strategies are considered traditional and the most studied, while cluster strategies are the least studied due to the non-trivial approaches to their study: in particular, it is necessary to resolve the issue of "online" searching for clusters in the system, which will be discussed in the third chapter. However, due to the emerging compromise between the complexity of the calculation and the accuracy of system control, it is precisely the meso-scale control that is emphasized in the dissertation. The next section analytically demonstrates the advantage of cluster (meso-scale) control

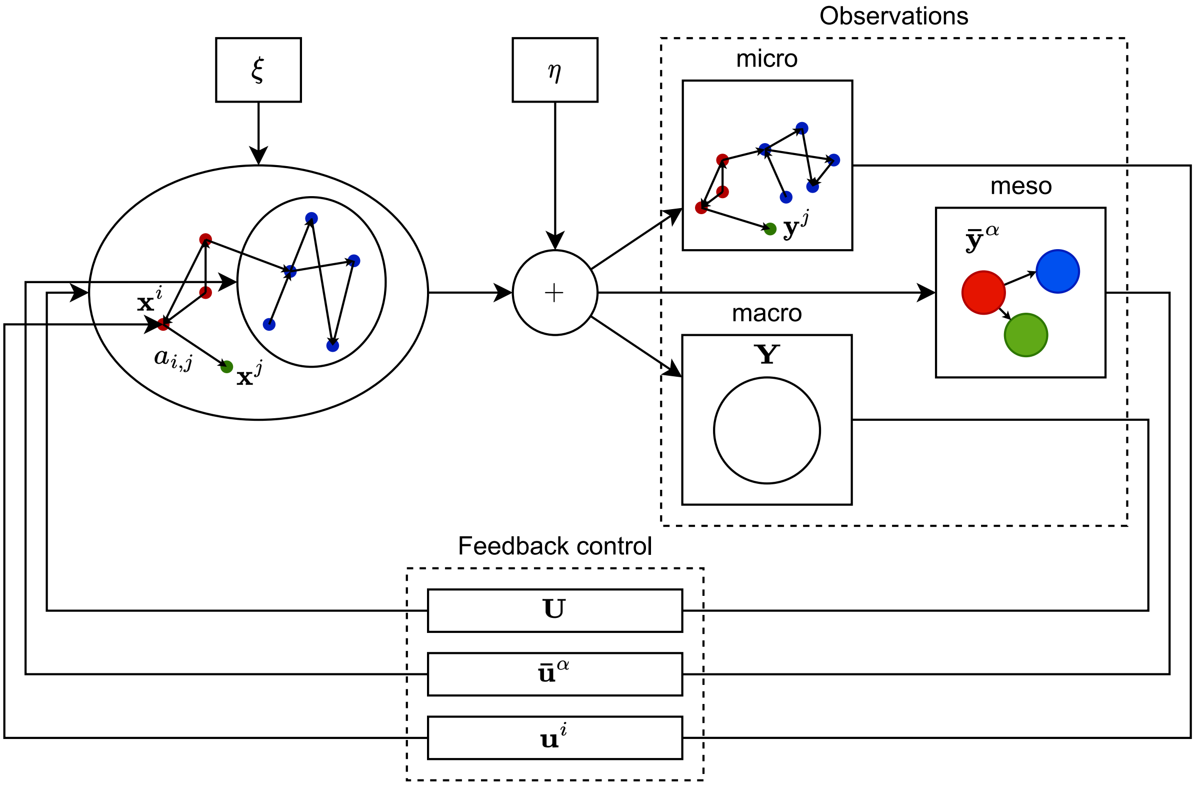

Figure 2.1: Observational feedback control for a complex multi-agent system with clustering. The system is represented as a composition of interacting agents defined by states $\mathbf{x}^i$ and connected by an adjacency matrix A with elements $a^{i,j}$ equal to 1 in the case of a connection between agents i and j, and equal to 0 if there is no connection. The system is affected by external disturbances ξ, which can directly change the connectivity or states of the agents. Observations are subject to η noise and are divided into three levels: micro- (individual agents), meso- (clusters) and macroscopic (system as a whole). Based on the type of observations, control is divided into three types of inputs: $\mathbf{u}^i$—local control input of agent i, $\bar{\mathbf{u}}^\alpha$—cluster control for coalitions α and $\mathbf{u}$—global control action, the same for all agents.

over local and global control when a realistic condition on the number of clusters relative to the number of agents in the system $s \ll N$ is met (see equation (1.4)).

2.2 Adaptive meso-scale control of complex multi-agent networked dynamical systems

Let's consider a cyber-physical (in the sense of modeling physical characteristics—coordinates and speeds) complex multi-agent networked dynamical system of N agent-particles in $2d$-dimensional space, each of them having a state $\{\mathbf{x}_t^i, \mathbf{c}_t^i\}$ (coordinate and speed), represented as a vector from $\mathbb{R}^d \times \mathbb{R}^d$. We consider the system on a finite time interval $\mathcal{T} = [0, \ldots, T]$. Let's assume that clustering occurs in the system, which (under the influence of the external environment or internal communications) changes at certain moments in time $\{t_1, \ldots, t_K\}$, and these time counts correspond to coalition structures $\mathcal{CS}_k$, $k \in \{1, \ldots K-1\}$. The time costs of transient processes when changing one cluster pattern to another are proportional to some time δ. Further we assume that $\delta \ll T$: the change of cluster patterns occurs quickly enough to consider the dynamics of the system only on "long" time intervals, on which the coalition structure is stable. We denote such time intervals as $\mathcal{T}_k = [t_k, t_{k+1}]$. Accordingly, on $\mathcal{T}_k$ the coalition structure $\mathcal{CS}_k$ is constant, i.e. the cluster pattern remains unchanged. Note that, following the paradigm of multi-agent technologies, local communications between agents stimulating convergence within one coalition [37, 38] (O. N. Granichin, V. A. Erofeeva) can contribute to the stabilization of the cluster pattern. Next, we propose a cyber-physical model of a discrete dynamic system with discrete time steps between the "observation" and "control" stages:

$$\mathbf{x}_{t+1}^i = \mathbf{x}_t^i + \mathbf{c}_t^i + \frac{1}{\mu}\mathbf{f}_t^i, \tag{2.4}$$

where $\mathbf{c}_t^i$ is the "velocity" vector; $\mathbf{f}_t^i \in \mathbb{R}^d$ is the "force" vector (acting on the agent), and μ is a certain weighting coefficient that is the same for all particles and plays the role of "mass". Consider this system further on the time interval $\mathcal{T}_k$, the beginning of which, without loss of generality, but for the convenience of subsequent reasoning, we set equal to 0. Given the initial conditions for the equation (2.4) and for the designated time interval be such that $\|\mathbf{x}_0^i\| > 0$ and $\|\mathbf{c}_0^i\| > 0$ (hereinafter the norm $\|\cdot\|$ is any, equivalent to ℓ_2), we set the control problem — bring all particles to one point (let it be 0, up to a change of variables, that is, translational invariance) *in one step*, which also formally means the problem of minimizing the following quality functional:

$$J(\mathbf{f}_t^1, \dots, \mathbf{f}_t^N) = \sum_{i=1}^{N} |\mathbf{x}_1^i|, t = 0, \tag{2.5}$$

with as little effort as possible. Below, two control strategies are proposed: based on observations of global and cluster states, then they are compared. The definition of clusters is carried out according to the model of cluster flows: we denote $\mathbf{z}_t^i = \mathbf{x}_t^i + \mathbf{c}_t^i$ as the outputs of agents to ensure synchronization between agents, then for some $\delta, \Delta_t^{ij} \leq \delta$ is the distance between agents in the cluster, limited by some δ. We also define the output for monitoring the states of agents $\mathbf{y}_t^i = \mathbf{z}_t^i$, equal to the output $\mathbf{z}_t^i$.

2.2.1 Macro-scale control strategy

In global control, a general force is applied: $\mathbf{f}_t^i = \mathbf{F}_t, \forall i$. This force is applied to all particle agents, which is essentially equivalent to the action on the center of mass:

$$\mathbf{X}_t = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_t^i. \tag{2.6}$$

Additionally, we define the average particle speed as

$$\mathbf{C}_t = \frac{1}{N} \sum_{i=1}^{N} \mathbf{c}_t^i. \tag{2.7}$$

Thus, $\mathbf{Y}_t = \mathbf{X}_t + \mathbf{C}_t$ plays the role of global observation and is further used to obtain the optimal control strategy.

Let us denote $\mathbf{U}_t := \mathbf{F}_t/\mu$ the control input with amplitude $|U|$ and control direction coinciding with the direction of the force. After substituting the global control into (2.4) and (2.5) we get

$$J_{\text{glob}}(\mathbf{U}_0) = \sum_{i=1}^{N} |\mathbf{x}_0^i + \mathbf{c}_0^i + \mathbf{U}_0| = \sum_{i=1}^{N} |\mathbf{y}_0^i + \mathbf{U}_0|. \tag{2.8}$$

The optimal solution (2.8) is the regulator

$$\mathbf{U}_{\text{opt},0} = -\mathbf{Y}_0 = -(\mathbf{X}_0 + \mathbf{C}_0),$$

thus the minimum (2.8) is

$$\min J_{\text{glob}}(\mathbf{U}_0) = \sum_{i=1}^{N} |\mathbf{y}_0^i - \mathbf{Y}_0|. \tag{2.9}$$

2.2.2 Meso-scale control strategy

Now consider the total force applied to each individual cluster (more precisely, the agent in the cluster): $\mathbf{f}_t^i = \bar{\mathbf{f}}_t^\alpha$, $i \in \mathcal{C}_t^\alpha$, where $\mathcal{C}_t^\alpha$—coalition, a set of agent indices in the α cluster. In this case, such a force acts on the centers of mass of each cluster and the average velocity of the cluster particles, respectively:

$$\bar{\mathbf{x}}_t^\alpha = \frac{1}{|\mathcal{C}_t^\alpha|} \sum_{i \in \mathcal{C}_t^\alpha} \mathbf{x}_t^i, \;\; \bar{\mathbf{c}}_t^\alpha = \frac{1}{|\mathcal{C}_t^\alpha|} \sum_{i \in \mathcal{C}_t^\alpha} \mathbf{c}_t^i.$$

Thus, $\bar{\mathbf{z}}_t^\alpha = \bar{\mathbf{x}}_t^\alpha + \bar{\mathbf{c}}_t^\alpha$ plays the role of monitoring the state of the cluster and is used further in deriving the optimal cluster (meso-scale) control strategies.

Let us denote $\bar{\mathbf{u}}_t^\alpha := \bar{\mathbf{f}}_t^\alpha/\mu$ the control input with amplitude $|\bar{\mathbf{u}}_t^\alpha|$ and control direction, coinciding with the direction of the force applied to the α cluster. Reasoning similar to that outlined in the previous subsection allows us to obtain an optimal cluster control strategy:

$$\min J_{\text{clust}}(\bar{\mathbf{u}}_0^1, \ldots, \bar{\mathbf{u}}_0^s) = \sum_{\alpha=1}^{s} \sum_{i \in \mathcal{C}_0^\alpha} |\mathbf{y}_0^i - \bar{\mathbf{y}}_0^\alpha|, \tag{2.10}$$

where s is the number of clusters, and then

$$\bar{\mathbf{u}}^{\alpha}_{\text{opt},0} = -\bar{\mathbf{y}}^{\alpha}_0.$$

2.2.3 Micro-scale control strategy

However, the most accurate method is micro-scale control, which sets individual impact for each agent separately. If we consider a system (2.4) for which the optimal micro-scale control strategy $\mathbf{u}^i_{\text{opt},0} = -\mathbf{y}^i_0$ is applied, then the functional quality

$$\min J_{\text{mic}}(\mathbf{u}^1, \ldots, \mathbf{u}^N) = 0 \tag{2.11}$$

at the optimal point, which represents the best possible solution.

2.2.4 Comparison of meso- and macro-scale control strategies

Next, we can compare the two strategies in terms of the corresponding quality functionals (2.9) and (2.10) for the calculated optimal controls. Intuitively, the cluster control strategy is specifically “tuned” to change the states of individual groups of agents, while the global strategy is roughly adjusted to the global state, which can vary significantly in different parts of the system. In this case, the macro-scale approach is unable to achieve the accuracy that the meso-scale strategy can achieve, provided that the cluster structure is non-trivial: $s > 1$. In fact, the equations (2.9) and (2.10) differ only in the corresponding control functions: $-\mathbf{Y}_0$ can be considered as a perturbed (or coarsened) version of $-\bar{\mathbf{z}}^{\alpha}_0$.

To summarize the above, a lemma is proposed that consolidates the above reasoning for a dynamical system and the difference between the quality functionals and the quality functionals.

L e m m a 1. *Consider a model* (2.4) *with* N *agents,* $\|\mathbf{x}_0^i\| > 0$ *and* $\|\mathbf{c}_0^i\| > 0$ *for all* i*. Let clustering arise at the time* $t = 0$*, the number of clusters is equal to* $s > 1$ *and each cluster has* $|\mathcal{C}_0^\alpha|$ *agents, so that* $\sum_{\alpha=1}^{s} |\mathcal{C}_0^\alpha| = N$*. Provided that the quality functionals* (2.9) *and* (2.10) *are in the corresponding minima (that is, the corresponding optimal controllers are substituted into them), it is true that from*

$$\left| \sum_{\alpha=1}^{s} |\mathcal{C}_0^\alpha| (\bar{\mathbf{y}}_0^\alpha - \mathbf{Y}_0) \right| > 0, \tag{2.12}$$

should

$$\min J_{\text{clust}} < \min J_{\text{glob}}. \tag{2.13}$$

Proof.

The lemma states that

$$\sum_{\alpha=1}^{s} \sum_{i \in \mathcal{C}_0^\alpha} |\mathbf{y}_0^i - \bar{\mathbf{y}}_0^\alpha| < \sum_{i=1}^{N} |\mathbf{y}_0^i - \mathbf{Y}_0| \tag{2.14}$$

if the inequality (2.12) is true. Consider the equation (2.10) with the added non-zero perturbation γ^α:

$$\sum_{\alpha=1}^{s} \sum_{i \in \mathcal{C}_0^\alpha} |\mathbf{y}_0^i - \bar{\mathbf{y}}_0^\alpha + \gamma^\alpha|,$$

and let $\gamma^\alpha = \bar{\mathbf{y}}_0^\alpha - \mathbf{Y}_0$. Hence,

$$\sum_{\alpha=1}^{s} \sum_{i \in \mathcal{C}_0^\alpha} |\mathbf{y}_0^i - \bar{\mathbf{y}}_0^\alpha + \gamma^\alpha| - \sum_{\alpha=1}^{s} \sum_{i \in \mathcal{C}_0^\alpha} |\mathbf{y}_0^i - \bar{\mathbf{y}}_0^\alpha| \geq$$

$$\geq \left| N\mathbf{Y}_0 - \sum_{\alpha=1}^{s} |\mathcal{C}_0^\alpha| \bar{\mathbf{y}}_0^\alpha + \sum_{\alpha=1}^{s} |\mathcal{C}_0^\alpha| \gamma^\alpha - N\mathbf{Y}_0 + \sum_{\alpha=1}^{s} |\mathcal{C}_0^\alpha| \bar{\mathbf{y}}_0^\alpha \right| = \left| \sum_{\alpha=1}^{s} |\mathcal{C}_0^\alpha| \gamma^\alpha \right| > 0$$

Then

$$\sum_{\alpha=1}^{s} \sum_{i \in \mathcal{C}_0^\alpha} \Big(|\mathbf{y}_0^i - \bar{\mathbf{y}}_0^\alpha + \gamma^\alpha| - |\mathbf{y}_0^i - \bar{\mathbf{y}}_0^\alpha| \Big) > 0,$$

and thus the inequality from the lemma (2.14) is true for the chosen perturbation, which completes the proof.

□

2.2.5 Comparison of meso- and micro-scale control strategies

Despite the fact that the micro-scale control strategy provides the best minimization of the considered functional (2.11), in many applications to complex systems with a large number of agents, it is impossible to implement such control for technical reasons and due to limited resources. Indeed, to find $\mathbf{u}^i_{\mathrm{opt},0}$ it is necessary to measure all microstates $\mathbf{x}^i_0$ and $\mathbf{c}^i_0$, which is often impossible. In terms of complexity, such a task is comparable to determining all the microstates of gas molecules in a certain volume instead of using the methods of statistical physics for a comprehensive description of the processes occurring. Another example is the task of monitoring brain activity: when measuring neural oscillations, aggregated activity is usually observed in different brain regions at the meso-scale, that is, at the level of many synchronized neurons. At the same time, measuring the activity of an individual neuron is much more difficult, both due to complex measuring equipment and large amounts of data that arise during the measurement process and require processing, and such accuracy is often not required for successful diagnosis.

However, the above-mentioned difficulties associated with the synthesis of micro-scale control can be circumvented using meso-scale control, provided that clusters have formed in the system. For sufficiently dense clusters, the difference between $\min J_{\mathrm{mic}}(\mathbf{u}^1_t, \ldots, \mathbf{u}^N_t)$ and $\min J_{\mathrm{clust}}(\bar{\mathbf{u}}^1_t, \ldots, \bar{\mathbf{u}}^s_t)$ will be as small as the maximum distance between agents in the cluster δ, which allows for more efficient (in the computational sense) synthesis of control without significant errors.

T h e o r e m 1. *Consider a complex multi-agent networked dynamical system of N agents following discrete dynamics* (2.4) *on a finite time interval $\mathcal{T} = [0, \ldots, T]$. Introduce outputs $\mathbf{y}_t^i$ to monitor the states of agents. Let a clustering arise in the system that changes at some points in time $\{t_1, \ldots, t_K\}$, and let these time counts correspond to coalition structures $\mathcal{CS}_k$, $k \in \{1, \ldots K-1\}$ that are stable on intervals $\mathcal{T}_k = [t_k, t_{k+1}]$, and let the transient processes when changing cluster patterns be proportional in duration to time $\tau \ll T$. We assume that the maximum distance between agents in the cluster δ is small. Let's consider the functional* (2.3) *in discrete form for micro- and meso-scale control methods:*

$$J_{\text{mic}}(\mathbf{u}_t^1, \ldots, \mathbf{u}_t^N) = \sum_{k=1}^{K-1} \sum_{i \in \mathcal{N}} |\mathbf{y}_{t_k}^i + \mathbf{u}_{t_k}^i| + C_1 K \tau, \tag{2.15}$$

$$J_{\text{clust}}(\bar{\mathbf{u}}_t^1, \ldots, \bar{\mathbf{u}}_t^s) = \sum_{k=1}^{K-1} \sum_{\alpha=1}^{s_k} \sum_{i \in \mathcal{C}_0^\alpha} |\mathbf{y}_{t_k}^i + \bar{\mathbf{u}}_{t_k}^\alpha| + C_2 K \tau, \tag{2.16}$$

where C_1, C_2 are some constants, and the terms $C_1 K\tau$ and $C_2 K\tau$ correspond to transient processes. Also assume that $s_k < S \ll N$, where S limits the number of clusters on the interval $\mathcal{T}$. Then, under optimal control $\mathbf{u}_{t_k}^i = -\mathbf{y}_{t_k}^i$ and $\bar{\mathbf{u}}_{t_k}^\alpha = -\bar{\mathbf{y}}_{t_k}^\alpha$ the difference between the values of the functionals (2.15) *and* (2.17) *is small and proportional to δ:*

$$\left| \min J_{\text{clust}}(\bar{\mathbf{u}}_t^1, \ldots, \bar{\mathbf{u}}_t^s) - \min J_{\text{mic}}(\mathbf{u}_t^1, \ldots, \mathbf{u}_t^N) \right| < \varepsilon, \tag{2.17}$$

where $\varepsilon = CKS\delta$ and C is some constant that limits the magnitude of the difference in the functionals on intervals corresponding to the transient processes.

Proof. Initially, we substitute $\mathbf{u}_{t_k}^i = -\mathbf{y}_{t_k}^i$ into (2.15) and this way we get $\min J_{\text{mic}}(\mathbf{u}_t^1, \ldots, \mathbf{u}_t^N) = C_1 K \tau$. Let's now consider

$$\min J_{\text{clust}}(\bar{\mathbf{u}}_t^1, \ldots, \bar{\mathbf{u}}_t^s) = \sum_{k=1}^{K-1} \sum_{\alpha=1}^{s_k} \sum_{i \in \mathcal{C}_0^\alpha} |\mathbf{y}_{t_k}^i - \bar{\mathbf{y}}_{t_k}^\alpha| + C_2 K \tau$$

with optimal meso-scale management. The triple sum can be bounded as follows:

$$\sum_{k=1}^{K-1}\sum_{\alpha=1}^{s_k}\sum_{i\in\mathcal{C}_0^\alpha}|\mathbf{y}_{t_k}^i-\bar{\mathbf{y}}_{t_k}^\alpha|\le KS\frac{\delta}{2},$$

where dividing by $\frac{\delta}{2}$ bounds the maximum distance between the centroid of a cluster and an agent from that cluster. Let $C_1=\widetilde{C}_1S\frac{\delta}{2\tau}$, $C_2=\widetilde{C}_2S\frac{\delta}{2\tau}$, where $\widetilde{C}_1$ and $\widetilde{C}_2$ are constants, then

$$\left|\min J_{\text{clust}}(\bar{\mathbf{u}}_t^1,\ldots,\bar{\mathbf{u}}_t^s)-\min J_{\text{mic}}(\mathbf{u}_t^1,\ldots,\mathbf{u}_t^N)\right|\le$$

$$\le KS\frac{\delta}{2}+\widetilde{C}_1KS\frac{\delta}{2}+\widetilde{C}_2KS\frac{\delta}{2}=\frac{(1+\widetilde{C}_1+\widetilde{C}_2)}{2}KS\delta=CKS\delta,$$

where $C=\frac{(1+\widetilde{C}_1+\widetilde{C}_2)}{2}$. We denote the resulting value $CKS\delta$ by ε. □

Theorem 1 demonstrates a small difference between the functionals depending on micro- and meso-scale control actions for systems with dense clusters. The smallness of this difference implies, to some extent, the approximating property of the meso-scale strategy, which allows achieving the control goal in a way close to optimal. At the same time, since $s_k<S\ll N$, the use of meso-scale strategies allows saving on the amount of resources spent on calculating the control action for more complex systems: for example, nonlinear ones, which will be discussed in the next section. Before this, a method of adaptive meso-scale control of complex multi-agent networked dynamical systems is formulated below.

2.2.6 Method (algorithm) of adaptive meso-scale control of complex multi-agent networked dynamical systems

Let us consider a discrete complex multi-agent networked dynamical system with clustering, which corresponds to the coalition structure $\mathcal{CS}_t$:

$$\mathbf{x}_{t+1}^i=\mathbf{f}_t^i(\mathbf{x}_t^i,\bar{\mathbf{u}}_t^\alpha), i\in\mathcal{C}_t^\alpha, \mathcal{C}_t^\alpha\in\mathcal{CS}_t$$

on the time interval $[0, \dots, T]$. Meso-scale control for which is synthesized by minimizing the functional

$$\bar{\mathbf{u}}_t^\alpha \leftarrow \underset{\bar{\mathbf{u}}}{\operatorname{argmin}}\, J(\bar{\mathbf{u}}), \alpha \in \mathcal{S}_t,$$

where $\mathcal{S}_t$ are cluster indices at time t. It is assumed that at some time points $t_k, k \in \{1, \dots K\}$ the coalition structure changes, and therefore the system needs to be adapted to such conditions. For the described system, the corresponding adaptive control algorithm is formulated in listing 1.

Algorithm 1 Adaptive meso-scale control of a complex multi-agent networked dynamical system

$t \leftarrow 0$
$\mathbf{x}_0^i, i \in \mathcal{N}$
while $t < T$ **do**
  compute coalition structure $\mathcal{CS}_t$
  if $\mathcal{CS}_t = \mathcal{CS}_{t-1}$ **then**
    $\bar{\mathbf{u}}_t^\alpha \leftarrow \bar{\mathbf{u}}_{t-1}^\alpha$
  else
    $\bar{\mathbf{u}}_t^\alpha \leftarrow \operatorname{argmin}_{\bar{\mathbf{u}}} J(\bar{\mathbf{u}})$
  end if
  $\mathbf{x}_{t+1}^i \leftarrow \mathbf{f}_t^i(\mathbf{x}_t^i, \bar{\mathbf{u}}_t^\alpha), i \in \mathcal{C}_t^\alpha, \mathcal{C}_t^\alpha \in \mathcal{CS}_t$
end while

2.3 Adaptive meso-scale control of nonlinear systems

Despite the fact that most cyber-physical systems used in practice can be successfully represented by linear models — especially near stationary points — the role of nonlinear models that can cover an incomparably larger range of situations cannot be underestimated. As an example of generalizing linear models to the class of nonlinear ones with a sinusoidal state exchange protocol, the current section will consider a simple but at the same time flexible nonlinear Kuramoto model with many applications [1] (J. Acebron), describing the oscillatory dynamics of locally interacting oscillators. The model will demonstrate the conditions for achieving and maintaining a cluster structure for the introduction of meso-scale control.

2.3.1 Kuramoto model of coupled oscillators

The dynamics of the Kuramoto model for a certain network of N agents with one degree of freedom (which is often called the oscillator phase) is described by the following system of differential equations:

$$\dot{\mathbf{x}}^i(t) = \mathbf{w}^i + \sum_{j=1}^{N} \mathcal{B}^{ij} \sin\left(\mathbf{x}^j(t) - \mathbf{x}^i(t)\right), \tag{2.18}$$

where $\mathbf{x}^i(t)$ is the phase of agent i, $\mathcal{B}^{ij}$ is the weighted network connectivity matrix, and $\mathbf{w}^i$—natural frequency. According to [18] (D. Benedetto), [26] (N. Chopra) and [56] (A. Jadbabaie), agents come to the state of frequency ($\dot{\mathbf{x}}^i = \dot{\mathbf{x}}^j\ \forall i, j \in \{1, \ldots, N\}$) or phase ($\mathbf{x}^i = \mathbf{x}^j\ \forall i, j \in \{1, \ldots, N\}$) synchronization under certain conditions on $\mathbf{w}^i$ and $\mathcal{B}^{ij}$.

There are many extensions of the Kuramoto model, for example, a model with time-dependent connectivity matrix $\mathcal{B}^{ij}(t)$ and frequencies $\mathbf{w}^i(t)$ [63] (W. Lu); version of the model with time and phase delays [60] (T. Kotwal), [70] (E. Montbrió). The main application of the model is the description of the processes occurring in the brain [94] (M. Sadilek). Thus, three modes of brain operation were identified, corresponding to three states of the oscillator network: asynchronous, completely synchronous and “chaotic”, corresponding to cluster synchronization. This model is also suitable for describing a large number of biological swarm systems.

Let’s consider the model (2.18) from the point of view of cybernetics and control theory. You can explicitly define a communication protocol between agents, which looks like this:

$$\mathbf{u}^i(t) = \mathbf{w}^i + \sum_{j=1}^{N} \mathcal{B}^{ij} \sin\left(\mathbf{x}^j(t) - \mathbf{x}^i(t)\right), \tag{2.19}$$

that is, the corresponding communication outputs $\mathbf{y}^i(t) \equiv \mathbf{x}^i(t)$. The matrix $\mathcal{B}$ plays the role of a connectivity matrix for a certain configuration of a network of oscillators ($b^{ij} \neq 0 \iff j \in \mathcal{N}^i \iff \exists (j \to i) \in \mathcal{E}(t)$); and $\mathbf{x}^i(t) \in S^1\ \forall i\ \forall t \geq 0$. Synchronization in the Kuramoto model can manifest itself in the form of frequency or phase consensus. The difference between them lies in the choice of output $\mathbf{z}^i(t)$: $\mathbf{z}^i(t) = \dot{\mathbf{x}}^i(t)$ and $\mathbf{z}^i(t) = \mathbf{x}^i(t)$ respectively. In the current work, the main attention is paid to the first option, in which clustering is determined by the coincidence of oscillator frequencies.

In many works on the analysis of the Kuramoto model, a constraint on connectivity between agents is introduced, represented in the form $b^{ij} = \frac{C}{N}\ \forall i,j$, where C is a constant. This restriction is called mean field coupling and forces full connectivity in the communication graph of communication between agents. However, in practice there are many more examples of networks with incomplete (or even sparse) graph topology than of highly connected networks: for example, these are collections of neurons, swarm structures in herds of animals, social networks or groups of robots. Moreover, the topology of the graph $\mathcal{G}(t)$ corresponding to a particular multi-agent network is assumed to change over time. In this regard, the Kuramoto model with more realistic restrictions is considered below:

$$\dot{\mathbf{x}}^i(t) = \mathbf{w}^i + \rho \sum_{j \in \mathcal{N}^i(t)} \sin\left(\mathbf{x}^j(t) - \mathbf{x}^i(t)\right), \tag{2.20}$$

where ρ is a constant, and i is influenced only by neighboring agents from $\mathcal{N}^i(t)$. Let us further introduce the incidence matrix of the graph $\mathcal{G}(t)$ as $A^{ij}(t) \in \{0,1\}\ \forall i,j\ \forall t \geq 0$, such thus $\mathcal{B}^{ij}(t) = \rho A^{ij}(t)$ in the general case with time dependence. Intuitively, a value of 0 could mean “agent j has no connection with agent i ($j \notin \mathcal{N}^i$)", while 1 means "agent i is available to agent j ($i \in \mathcal{N}^j$)".

Now we define the control goal: agents strive to synchronize using the protocol in (2.20), from which it follows that $\rho > 0$. However, the agents at the same time experience free movement (the so-called "drift") with a speed of $\mathbf{w}^i \geq 0$, which can be thought of as movement along a circle in the complex plane.

2.3.2 Meso-scale control

Let the clustering $\mathcal{CS}(t_1)$ occur at time t_1 in the model (2.20); let this clustering remain constant over the time interval $T = [t_1, +\infty)$. Everywhere below $t \in T$. Then the topology $\mathcal{G}(t)$ corresponding to a given multi-agent network does not change to T. Thus, A^{ij} is also constant. In accordance with the cluster flow framework, a modification of the Kuramoto model is proposed, including cluster control (assuming that $i \in \mathcal{C}_0^\alpha$):

$$\dot{\mathbf{x}}^i(t) = \mu^i \bar{\mathbf{u}}^\alpha(t, \bar{\mathbf{x}}^\alpha(t)) + \mathbf{w}^i + \rho \sum_{j=1}^{N} A^{ij} \sin\left(\mathbf{x}^j(t) - \mathbf{x}^i(t)\right), \tag{2.21}$$

where $\bar{\mathbf{u}}^\alpha(\cdot)$ is a meso-scale function in the sense that it is the same for the entire cluster $\mathcal{C}_0^\alpha$, μ^i —*sensitivity* to the control function $\bar{\mathbf{u}}^\alpha(\cdot)$.

The proposed control functions, however, only bring agents into a synchronized state if certain conditions are met. First, synchronization depends on the values of μ^i: some agents in the $\mathcal{C}_0^\alpha$ cluster may respond to $\bar{\mathbf{u}}(\cdot)$ control strongly enough that disrupt the cluster structure. Secondly, a large difference in the values of $\mathbf{w}^i$ can also lead to undesirable consequences, including chaotic behavior of the system.

Before looking for sufficient conditions for the invariance of the cluster structure on the parameters in the model (2.21), we first formulate and prove a theorem for the model (2.20) about the relationship between the natural frequencies $\mathbf{w}^i$ and the values $\mathcal{B}^{ij} = \rho A^{ij}$ necessary to achieve clustering.

T h e o r e m 2. *Consider a multi-agent network corresponding to the model* (2.20) *and the graph* $\mathcal{G}$ *with the connectivity matrix* A. *Let* $t \in T$, *output* $\mathbf{z}^i(t) = \dot{\mathbf{x}}^i(t)$ *and* $\Delta^{ij}(t) = |\mathbf{z}^i(t) - \mathbf{z}^j(t)|$. *The following conditions are sufficient to achieve cluster synchronization with* $r_i(t) = 0$ $\forall i$ *(coincidence of states of agents from the same cluster) in this network:*

1. *For* $i, j \in \mathcal{C}_0^\alpha$ *such that* $\mathbf{w}^j - \mathbf{w}^i \geq 0$

$$\mathbf{w}^j - \mathbf{w}^i \leq \rho \sin\left(\frac{\Delta \mathbf{x}^{ji}}{2}\right) \sum_{l=1}^{N} [A^{il} + A^{jl}], \tag{2.22}$$

where $\sin\left(\frac{\Delta \mathbf{x}^{ji}}{2}\right) = 1$ *in the case of* $A^{ij} = A^{ji} = 0$; *otherwise*

$$\sin\left(\frac{\Delta \mathbf{x}^{ji}}{2}\right) = \max\left\{\sqrt{1 - (\Gamma^i(j))^2}, \sqrt{1 - (\Gamma^j(i))^2}, \frac{\sqrt{2}}{2}\right\}, \tag{2.23}$$

Where

$$\Gamma^i(j) = \frac{-d^i(j) + \sqrt{(d^i(j))^2 + 8(A^{ij} + A^{ji})^2}}{4(A^{ij} + A^{ji})}. \tag{2.24}$$

2. *For* $i \in \mathcal{C}_0^\alpha$, $j \in \mathcal{C}_\beta$, $\alpha \neq \beta$

$$|\mathbf{w}^i - \mathbf{w}^j| > 0. \tag{2.25}$$

3. *The graph* $\mathcal{G}$ *is strongly connected.*

R e m a r k 1. *The idea stated in the theorem is the following: if* $|\mathbf{w}^i - \mathbf{w}^j|$ *is large, the coupling strength may not be enough to form clusters, then the value of* ρ *should be sufficient big to overcome the drift. In the simplest case, where* $\mathbf{w}^i = \mathbf{w}$

$forall i$, *synchronization always occurs* $\forall \rho > 0$. *It turns out that in order to achieve cluster synchronization with the coincidence of the states of agents from the same cluster, the difference* $|\mathbf{w}^i - \mathbf{w}^j|$ *must necessarily be non-zero for agents* i *and* j *from different clusters.*

Proof. In order to simplify the notation, the time dependence is omitted throughout the proof: $\dot{\mathbf{x}}^i(t)$ as $\dot{\mathbf{x}}^i$ (the same applies to $\mathbf{x}^i$ and Δ^{ij}), and such a dependence is implied. Following Definition 5, $\Delta^{ij} = \left|\dot{\mathbf{x}}^i - \dot{\mathbf{x}}^j\right| = 0\ \forall i, j \in \mathcal{C}_0^\alpha$ and $\Delta^{ij} > 0\ \forall i \in \mathcal{C}_0^\alpha, j \in \mathcal{C}_\beta, \alpha \neq \beta$.

Since the situation when i and j are from different clusters is easier to prove, we will consider it first. Substituting the right side of the model (2.20) instead of $\dot{\mathbf{x}}^i$, we obtain the following expression Δ^{ij}:

$$\begin{aligned}\Delta^{ij} = \Bigg|\mathbf{w}^i - \mathbf{w}^j + \rho\Bigg(&\sum_{l=1}^{N} A^{il}\sin(\mathbf{x}^l - \mathbf{x}^i) - \\ &- \sum_{l=1}^{N} A^{jl}\sin(\mathbf{x}^l - \mathbf{x}^j)\Bigg)\Bigg| > 0.\end{aligned} \tag{2.26}$$

Setting the arguments of the sines in (2.26) equal to 0, we obtain a sufficient condition for the natural frequencies.

Next, consider the situation $i, j \in \mathcal{C}_0^\alpha$. Let the left side (2.26) be strictly equal to zero; assume that $\mathbf{w}^j - \mathbf{w}^i \geq 0$ without loss of generality. Let E be the functional

$$\begin{aligned}E &= \sum_{l=1}^{N} A^{il}\sin(\mathbf{x}^l - \mathbf{x}^i) - \sum_{l=1}^{N} A^{jl}\sin(\mathbf{x}^l - \mathbf{x}^j) = \\ &= (A^{ij} + A^{ji})\sin(\mathbf{x}^j - \mathbf{x}^i) + \sum_{l=1, l\neq j}^{N} A^{il}\sin(\mathbf{x}^l - \mathbf{x}^i) - \\ &\quad - \sum_{l=1, l\neq i}^{N} A^{jl}\sin(\mathbf{x}^l - \mathbf{x}^j),\end{aligned} \tag{2.27}$$

which follows. First-order necessary conditions for maximizing E

$$\frac{\partial E}{\partial \mathbf{x}^i} = -(A^{ij} + A^{ji})\cos(\mathbf{x}^j - \mathbf{x}^i) - \sum_{l=1, l\neq j}^{N} A^{il}\cos(\mathbf{x}^l - \mathbf{x}^i) = 0, \tag{2.28}$$

$$\frac{\partial E}{\partial \mathbf{x}^j} = (A^{ij} + A^{ji}) \cos(\mathbf{x}^j - \mathbf{x}^i) + \sum_{l=1, l\neq i}^{N} A^{jl} \cos(\mathbf{x}^l - \mathbf{x}^j) = 0, \tag{2.29}$$

$$\frac{\partial E}{\partial \mathbf{x}^l} = A^{il} \cos(\mathbf{x}^l - \mathbf{x}^i) - A^{jl} \cos(\mathbf{x}^l - \mathbf{x}^j) = 0. \tag{2.30}$$

For the following reasoning to be true, let us assume $\exists l_1 : A^{il_1} = 1$ and $\exists l_2 : A^{jl_2} = 1$. Since i and j are arbitrary, the graph $\mathcal{G}$ thus turns out to be strongly connected. Consider the case $A^{ij} = A^{ji} = 0$. Then the cosines in the second terms of the equations (2.29) and (2.29) are also equal to 0, from which it follows

$$E_{\max} = \sum_{l=1}^{N} [A^{il} + A^{jl}]. \tag{2.31}$$

Now let $A^{ij} + A^{ji} \geq 1$. Consider the equation (2.30). In the case $A^{il} = A^{jl} = 0$, it turns out that $E_{\max}$ has the form in (2.31). If $A^{il} = 1$ and $A^{jl} = 0$, then, following (2.28), $\cos(\mathbf{x}^j - \mathbf{x}^i) = 0$. The same is true if $A^{il} = 0$ and $A^{jl} = 1$, following (2.29).

The most non-trivial situation is $A^{il} = A^{jl} = 1$. Using (2.30), we obtain that either $\mathbf{x}^l = \frac{\mathbf{x}^i + \mathbf{x}^j}{2}$ or $\mathbf{x}^i = \mathbf{x}^j = 0$. However, from the second it follows that $E \equiv 0$, which does not correspond to the goal of maximizing E. Thus, we substitute $\mathbf{x}^l = \frac{\mathbf{x}^i + \mathbf{x}^j}{2}$ into (2.28) and use the expression $\cos(2x) = 2\cos^2(x) - 1$:

$$\begin{aligned} 2(A^{ij} + A^{ji}) \cos^2\left(\frac{\mathbf{x}^j - \mathbf{x}^i}{2}\right) - (A^{ij} + A^{ji}) + \\ + \cos\left(\frac{\mathbf{x}^j - \mathbf{x}^i}{2}\right) \sum_{l=1, l\neq j}^{N} A^{il} = 0. \end{aligned} \tag{2.32}$$

After solving the quadratic equation (2.32) we obtain two solutions, the only one of which (with a “plus” sign) satisfies the condition

$$\cos\left(\frac{\mathbf{x}^j - \mathbf{x}^i}{2}\right) \in [-1, 1].$$

The chosen solution is denoted by $\Gamma^j(i)$—it corresponds to (2.24). In a similar way we obtain solutions (2.29). And again, only the solution with the "plus" sign, denoted as $\Gamma^i(j)$, is suitable. Let the optimal value $(\mathbf{x}^j - \mathbf{x}^i)_{\text{opt}} = \Delta\mathbf{x}^{ji}$ as in (2.23) , where the first two options are obtained from the expression $\sin(\arccos(x)) = \sqrt{1-x^2}$, while the last one is derived from (2.30) in case one of coefficients is 1 and the other is 0. Finally, let's substitute $\Delta\mathbf{x}^{ji}$ and $\mathbf{x}^l = \frac{\mathbf{x}^i+\mathbf{x}^j}{2}$ into (2.27):

$$E_{\max} = \sin\left(\frac{\Delta\mathbf{x}^{ji}}{2}\right)\sum_{l=1}^{N}[A^{il} + A^{jl}], \tag{2.33}$$

which completes the proof. □

The result can be generalized to the model (2.21). In order to simplify the notation, further $\bar{\mathbf{u}}^\alpha = \bar{\mathbf{u}}^\alpha(t, \bar{\mathbf{x}}^\alpha)$.

T h e o r e m 3. *Let's consider a multi-agent network corresponding to the model* (2.21)*. Let $t \in T$, output $\mathbf{z}^i(t) = \dot{\mathbf{x}}^i(t)$ and $\Delta^{ij}(t) = |\mathbf{z}^i(t) - \mathbf{z}^j(t)|$. Let also $\bar{\mathbf{u}}^\alpha$ not depend on $\mathbf{x}^i$ $\forall i$. The following conditions are sufficient to achieve* $(0,0)$ *synchronization in this network.*

1. *In the case of $i, j \in \mathcal{C}_0^\alpha$,*

$$|(\mu^j - \mu^i)\bar{\mathbf{u}}^\alpha| \leq 2\rho\sin\left(\frac{\Delta\mathbf{x}^{ji}}{2}\right)\sum_{l=1}^{N}[A^{il} + A^{jl}], \tag{2.34}$$

   where $\Delta\mathbf{x}^{ji}$ is the same as in Theorem 1 (including the case $A^{ij} = A^{ji} = 0$).

2. *For $i \in \mathcal{C}_0^\alpha$, $j \in \mathcal{C}_\beta$, $\alpha \neq \beta$*

$$|\mathbf{w}^i - \mathbf{w}^j + \mu^i\bar{\mathbf{u}}^\alpha(t, \bar{\mathbf{x}}^\alpha) - \mu^j\bar{\mathbf{u}}_\beta(t, \bar{\mathbf{x}}_\beta)| > 0. \tag{2.35}$$

3. *The graph $\mathcal{G}$ is strongly connected.*

Proof. Sufficient conditions can be obtained from Theorem 1 by substituting the meso-scale control $\mathbf{u}^i$ into Δ^{ij}:

$$\begin{aligned}\Delta^{ij} = \Bigg| & \mathbf{w}^i - \mathbf{w}^j + \mu^i \bar{\mathbf{u}}^\alpha - \mu^j \bar{\mathbf{u}}_\beta + \\ + \rho \Bigg(& \sum_{l=1}^{N} A^{il} \sin(\mathbf{x}^l - \mathbf{x}^i) - \sum_{l=1}^{N} A^{jl} \sin(\mathbf{x}^l - \mathbf{x}^j) \Bigg) \Bigg| .\end{aligned} \tag{2.36}$$

To begin with, let's assume that $i, j \in \mathcal{C}_0^\alpha$. Following the same reasoning as in the Proof of Theorem 1, the equation (2.36) is equal to 0, which yields (2.34). Now let $i \in \mathcal{C}_0^\alpha, j \in \mathcal{C}_\beta, \alpha \neq \beta$, so $\Delta^{ij} > 0$ in Eq. (2.36). Setting the sines equal to zero, as in Theorem 2, the desired condition for meso-scale control is obtained, which completes the proof. □

2.4 Summary

In the second chapter, a classification of control strategies is carried out. For a discrete system, methods for synthesizing macro-, meso-, and micro-scale control by minimizing the functional over the distance between the current state of agents and the target state are considered. The advantages of the meso-scale approach are demonstrated (lemma 1, theorem 1): compared to the macro-scale approach, it provides better accuracy, while in dense clusters it is slightly inferior to the micro-scale approach, while providing a significantly more efficient calculation of control inputs. A method for adaptive meso-scale control of complex multi-agent networked dynamical systems is proposed based on the cluster flow model introduced in the first chapter. Further, an example of meso-scale control synthesis is demonstrated for a nonlinear system of Kuramoto oscillators. The next chapter discusses a general method for obtaining aggregated features based on the compression sensing approach.

Chapter 3

Universal method “compressed sensing” for encoding and recognizing clusters in complex systems

To form a mesoscale control strategy, there remains a need to specify ways of transitioning to a space of reduced dimensionality, in which coalitions are considered as independent objects — meso-scale agents. It was mentioned in the previous chapter that certain aggregate characteristics of coalitions can be measured for such a transition. Methods for obtaining such characteristics will be discussed in detail in the current chapter, and to fully understand the scale of control of a complex system, we will provide a brief classification of them.

1. Algorithms like “local voting” [8, 9, 11] (N. O. Amelina), [7] (K. S. Amelin).

2. “External” (i.e., non-multi-agent) cluster identification methods, such as hierarchical clustering [71] (D. sukherjee) or centroid-based clustering [100] (S. Singh).

3. Data compression methods, such as compressed sensing [21] (E. J.

Candès), [31] (D. L. Donoho)

Among the three listed categories, the compression recognition method is of particular interest because it is a universal approach to compressing sparse data represented as a set of continuous values. It is known that the trajectories of agents in the physical environment (for example, sensor readings or coordinates of self-driving cars) are encoded as sequences of floating-point numbers, which can be effectively compressed using standard dictionary approaches based on the Lempel-Ziv-Markov chain-Algorithm (LZMA) [55] (A. Y. Horita) used in the ZIP software is not possible. Moreover, compression recognition bypasses the need to accurately compute cluster characteristics by calculating a compressed representation of the entire system at once. This sacrifices the visual interpretability of observations of a complex system, but this comes at a much higher cost of extremely efficient compression and the absence of a strict dependence on the number of clusters. In this regard, we further demonstrate the advantage of the compression identification method in relation to the compression of data on clusters in complex multi-agent systems.

One of the main disadvantages of compressed sensing is the loss of interpretability of the model: in a low-dimensional space, the trajectories of agent states transformed by randomized linear operations are aggregated in such a way that they can only be recognized by solving an optimization problem. However, in the approach proposed in the dissertation, recognition is not required to synthesize control, which allows saving time spent on solving the problem. Further, the advantage of the compressed sensing method in relation to the compression of data on clusters in complex multi-agent systems is demonstrated.

3.1 Compressed sensing method

The current section outlines the theoretical basis of the compression identification method in the terminology of signal processing theory. In what follows, the concept of signal presented here will be related to the concept of state in the MAS.

The discrete signal $\mathbf{x} \in \mathbb{R}^N$ is s-sparse with some *sparse* matrix $\Psi \in \mathbb{R}^{N\times N}$, if we represent it in the form

$$\mathbf{x} = \Psi\mathbf{s}, \tag{3.1}$$

where $\mathbf{s}$ has at most s $(s \ll N)$ nonzero components. We will further call such vectors s-sparse. This means that complete information is stored in only s components out of their total number, equal to dimension N. Let us define the matrix $m \times N(m \ll N)$ C as a measurement operator that transforms the original sparse signal from $\mathbb{R}^N$ to $\mathbb{R}^m$. In this section we will give an estimate for m depending on N and s, but for now it is enough to assume that $m \sim s$. Compression of data represented by the s-sparse vector $\mathbf{x}$, according to the compression identification methodology, is carried out as follows:

$$\mathbf{y} = C\mathbf{x} = C\Psi\mathbf{s} = \Phi\mathbf{s}, \tag{3.2}$$

where $\Phi - m\times N(m \ll N)$ "sampling matrix". We will further call the vector $\mathbf{y} \in \mathbb{R}^m$ the vector of measurements or the vector of compressed observations.

Restoring the original $\mathbf{x}$ from compressed observation is possible, despite the significant difference in dimensions. In the general case, the problem of estimating $\mathbf{x}$ for a given y is ill-conditioned, but in the sparse case the reconstruction becomes feasible if the following conditions for Φ, called the restricted isometry property (RIP), are satisfied:

$$(1 - \delta_s)||\mathbf{s}||_2^2 \leq ||\Phi\mathbf{s}||_2^2 \leq (1 + \delta_s)||\mathbf{s}||_2^2, \tag{3.3}$$

for all s-sparse vectors z for some δ_s between 0 and 1. In other words, intuitively, the matrix Φ should preserve the lengths of sparse vectors after compression.

Along with (3.3) in [53] (O. N. Granichin, D. V. Pavlenko) another frequently used condition is proposed - the modified restricted isometry property (sRIP), which allows us to reconstruct $\mathbf{x}$:

$$\lambda^{-1}||\mathbf{s}||_2 \leq ||\Phi\mathbf{s}||_2 \leq \lambda||\mathbf{s}||_2$$

for some $0 < \lambda < \infty$ and any nonzero vector $\mathbf{s}$ with nonzero components s.

According to [53] (O. N. Granichin, D. V. Pavlenko), RIP can be satisfied with high probability if the elements of C are randomly selected according to one of the following three distributions:

1. Gaussian:
$$c^{ij} \sim \mathcal{N}\left(0, \frac{1}{m}\right).$$
2. Symmetric Bernoulli distribution:
$$P(c^{ij} = \pm 1/\sqrt{m}) = \frac{1}{2}.$$
3. Trinomial Bernoulli distribution:
$$c^{ij} = \begin{cases} +\sqrt{3/m} & \text{with probability} \\ frac16 & \\ 0 & \text{with probability } \frac{2}{3} \\ -\sqrt{3/m} & \text{with probability} \\ frac16. & \end{cases}$$

For the randomized elements C described above and according to RIP (3.3),

$$s \geq k_1 s \log(N/s) \tag{3.4}$$

for $0 < \delta < 1$. The relation (3.4) guarantees that the measurement matrix C will satisfy RIP with probability $\geq 1 - 2e^{-k_2 s}$, where k_1 and k_2 are small positive constants depending only from δ. In [17] (R. Baraniuk) particular conditions for choosing the constants k_1 and k_2 are proposed. Under these conditions, C is universal (for all three types of distributions used to generate its elements) in the sense that any s-sparse $\mathbf{x}$ can be reconstructed from a given y of suitable dimension.

In addition to performing RIP, to unambiguously reconstruct the signal, it is necessary to pose an optimization problem with a constraint on the sparsity of the estimate $\mathbf{s}$:

$$\min ||\hat{\mathbf{s}}||_{\ell_0}, \text{ such that } ||\Phi\hat{\mathbf{s}} - \mathbf{y}||_2 = 0, \tag{3.5}$$

where $||\cdot||_{\ell_0}$ — ℓ_0-norm equal to the number of non-zero elements in the estimate of the vector $\hat{\mathbf{s}}$. Despite the fact that the introduction of such a norm makes it possible to explicitly optimize the estimates of the original signal in terms of sparsity, such a problem is NP-hard. The original paper on Compressed Sensing [21] (E. J. Candès) also proposed linear relaxation through the equivalent ℓ_1 norm, which provides an acceptable compromise between sparsity and computational complexity . The solution to the optimization problem (3.5) can be obtained by minimizing the norm ℓ_1 in polynomial time, using, for example, the interior point method [75] (Yu. E. Nesterov), [74] (A. S. Nemirovsky) or more advanced specialized solutions: "spectral projected gradient for L1" (SPGL1) [110, 111] (E. van den Berg), CVX [54] (D. A. Guimares) or deep learning methods [66] (A. L. sachidon), [27] (J. Chorowsky) Chorowski)). Robust approaches aimed at solving, among other things, the problem in the presence of disturbances are described.

3.1.1 Application to wireless sensor networks

Multi-agent networks with low-cost, simple hardware or software agents often have to process intensive data streams from multiple sensors, also called sensors. For more efficient signal detection, various decentralized approaches offer load balancing mechanisms for distributed computing, ranging from simple ones aimed at the cheapest microcontrollers (local voting [9] (N. O. Amelina), Kuramoto oscillator model [1] (J. Acebron), to highly accurate machine learning models for more advanced computer agents, such as the

Round-robin (RR) algorithm [15] (T. Balharith))). Even distribution of the load not only ensures that tasks are completed as quickly as possible, but also extends the life of the entire system. For example, energy-efficient load balancing in wireless sensor networks (WSN) [116] (Q. Wang), [119] (Y. Yuan) is relevant problem in the area under consideration.

Among related studies, the following two papers on clustering in wireless sensor networks describe clustering using compression recognition methodology. In [116] (Q. Wang) a detailed comparison of the clustering method using compression recognition and standard approaches to identifying clusters in networks, such as Threshold sensitive Energy Efficient sensor Network protocol (TEEN) is presented [120] (Y. Zaied). Finally, Y. Yuan describes an energy-efficient loop routing system with compression of sensor data.

3.2 Relationship between sparsity and clustering

In many applications, it is often impossible to accurately find clusters by specifying whether each agent belongs to a specific cluster. Moreover, the cluster structure may change over time, which may make it impossible to recalculate aggregated states using complex classical clustering algorithms. The compression identification method allows us to bypass these limitations, thanks to the method described below for representing the clustered system state vector as sparse, which opens up the possibility of compression according to the scheme described above. Suppose that by time $t = 0$ the multi-agent network has achieved complete clustering (that is, all agents within one cluster are grouped into a centroid) with s-clusters. This leads to $\mathbf{z}_t^i = \mathbf{z}_t^j$ for all $i, j \in \mathcal{C}^\alpha$ and α that are the cluster index. In fact, clustering leads to a certain pattern in the system that, geometrically and intuitively, is sparse in some basis. The above can be formalized by the following Lemma.

L e m m a 2. *Let a multi-agent system with N agents have complete clustering, resulting in the formation of $s \leq N$ clusters, which is equivalent to:*

$\mathbf{z}_t^i = \mathbf{z}_t^j$ for all $i, j \in \mathcal{C}^\alpha$, where α is the index of the coalition corresponding to the cluster. Then there exists a rarefaction matrix $\Psi \in \mathbb{R}^{N\times N}$ and a vector $\mathbf{s}_t^i$ for each i with s nonzero components such that $\mathbf{z}_t^i = \Psi\mathbf{s}_t^i$.

Proof. Consider a cluster $\alpha = 1$ with an aggregated state $\bar{\mathbf{z}}_t^\alpha = \mathbf{z}_t^i$ for all $i \in \mathcal{C}_t^\alpha$. Let us construct $\mathbf{s}_t^i$ as a vector whose α-th element is equal to $\bar{\mathbf{z}}^\alpha$, the rest are equal to 0. Next, we select the α-th column of Ψ as a column vector and assign its elements $i \in \mathcal{C}_t^\alpha$ the value 1, and the rest - 0.

In the case $s > 1$, the same procedure is repeated for all remaining $\alpha \in \{2, \ldots, s\}$. □

We can further demonstrate an even deeper relationship between the nature of sparsity and clustering. To do this, consider the model:

$$\mathbf{x}_{t+1}^i = \theta\mathbf{x}_t^i + \mathbf{u}_t^i, \tag{3.6}$$

and local voting protocol for the system (3.6):

- $\theta = 1$;
- $\mathbf{u}_t^i = \gamma\sum_{j\in\mathcal{N}_t^i} b_t^{ij}(\mathbf{x}_t^j - \mathbf{x}_t^i)$, where $\mathcal{N}_t^i$ is the set of agents adjacent to i, b_t^{ij} is an element of the adjacent matrix B_t, $\mathbf{x}_t^i$ — system state.

According to [8] (N. O. Amelina), such a system, with a given control protocol and a weighted-balanced communication graph, demonstrates global synchronization at time t_*, that is, $\lim_{t\to t_*}|\mathbf{x}_t^i - \mathbf{x}_t^j| = 0$ for all agents, where t_* is the point in time when full synchronization occurs. For the local voting protocol with clustering, the following theorem is proposed.

Theorem 1. *Let's consider the local voting protocol for the system (3.6). Let $\mathbf{x}_t^i \in \mathbb{R}^d$ for all agents. Let us introduce the state vector of the entire system $\mathbf{x}_t = \mathrm{col}(\mathbf{x}_t^1, \ldots, \mathbf{x}_t^N) \in \mathbb{R}^{Nd}$. Let $\mathbf{x}_{t_*}$ have $s \le N$ nonzero elements and the rest zeros. Then for a sparse basis Ψ and full clustering arising in a dual system with state $\mathbf{z}_{t_*} = \Psi\mathbf{x}_{t_*}$, the difference*

$$\lim_{t\to t_*}\left||\mathcal{C}_t^\alpha|\mathbf{x}_t - Q^{-1}\mathbf{z}_{t_*}P\right| = 0$$

under a local voting protocol, where $B = \Psi$, up to a permutation of elements z, specified by the matrices Q and P of the corresponding dimension.

Proof. A graph with an adjacency matrix Ψ has s connected components, since $Q^{-1}\Psi P$ is a block matrix with s blocks (the matrices Q and P sort the rows of Ψ without changing the graph structure). Since different coalitions correspond to different connected components, the clusters are not connected to each other, and the local voting protocol is only able to balance the values within each individual connected component. The Φ structure guarantees that each of its blocks forms a spanning tree for each connected component with the agent at the root of this tree having the initial state $|\mathbf{x}^i_{t_*}| > 0$ for each cluster α. Thus, each connected component eventually reaches a balanced state with $|\mathbf{x}^i_{t_*} - \mathbf{x}^j_{t_*}|$ for all agents from the same cluster. At the same time, since $\sum_{i\in\mathcal{C}^\alpha} \mathbf{x}^i_t$ is constant for all t (due to load balancing of the local voting protocol), the resulting states agents

$$\mathbf{x}^j_{t_*} = \frac{1}{|\mathcal{C}^\alpha_t|}\mathbf{x}^i_{t_*}$$

for all $j \in \mathcal{C}^\alpha_t$. □

3.2.1 Compressed sensing meso-scale control algorithm

In the case where the exact characteristics of clusters cannot be obtained by direct observations, the internal structure of the system can still be effectively extracted using compression identification methodology. As shown above, the sparseness of a multi-agent system is associated with the formation of cluster patterns in it. Let us now assume that the system has s clusters distributed in advance in an unknown way. According to the compression identification methodology [21] (E. J. Candès), a sufficient number of measurements to restore the state of the entire system is

$$s = 4s\log(N/s). \tag{3.7}$$

Accordingly, observations with this dimension can ensure controllability of the system at the meso-scale: $C\Psi\mathbf{u}_t^i \equiv \bar{\mathbf{u}}_t^\alpha \in \mathbb{R}^m, u \in \mathbb{R}^N$. Alternatively, if more precise control is required, the full state of the system can be restored before calculating the control by solving the ℓ_1-optimization problem for $\hat{\mathbf{u}}_t^i \in \mathbb{R}^N$:

$$\min \left\|\hat{\mathbf{u}}_t^i\right\|_{\ell_1}, \text{ such that } \left\|A\Psi\hat{\mathbf{u}}_t^i - \bar{\mathbf{u}}_t^\alpha\right\|_2 = 0,$$

using, for example, traditional interior point methods.

Listing 2 presents an algorithm for iterative meso-scale control with recognition by compression in a multi-agent system, generalizing the reasoning described above. The sparsity $z = \Psi x$ ensures successful state recovery from randomized compressed measurements after mutual restoration of state and control, while the control action s-sparse in Ψ $\bar{\mathbf{u}}_t^\alpha$ is synthesized according to any preferred optimization functional J with sparsity-inducing ℓ_1-regularization.

Algorithm 2 Compressed sensing meso-scale control algorithm

while $t < T$ **do**
- $\bar{\mathbf{x}}_{t-1} \leftarrow C\mathbf{x}_{t-1}$
- $\bar{\xi}_{t-1} \leftarrow C\xi_t$
- $\bar{\mathbf{u}}_{t-1} \leftarrow \text{argmin}\, J(\mathbf{u}_{t-1})$
- $\bar{\mathbf{x}}_t \leftarrow \theta\bar{\mathbf{x}}_{t-1} + \bar{\mathbf{u}}_{t-1} + \bar{\xi}_{t-1}$
- $\mathbf{y}_t \leftarrow \bar{\mathbf{x}}_t$
- $\hat{\mathbf{x}}_t \leftarrow$ restoring state by $\mathbf{y}_t$
- $\mathbf{x}_t \leftarrow \hat{\mathbf{x}}_t$
- $t \leftarrow t + 1$

end while

The algorithm is also schematically presented in Fig. 3.1. It is worth noting that the presented algorithm allows not only to compress data when directly collecting it during observation, but also to form control through indirect observation. For example, if there is no direct communication with the controlled system from a swarm of robots, observations can be generated using a video camera that records the trajectories of agents. In particular, this will be demonstrated for the example of the nonlinear Kuramoto model.

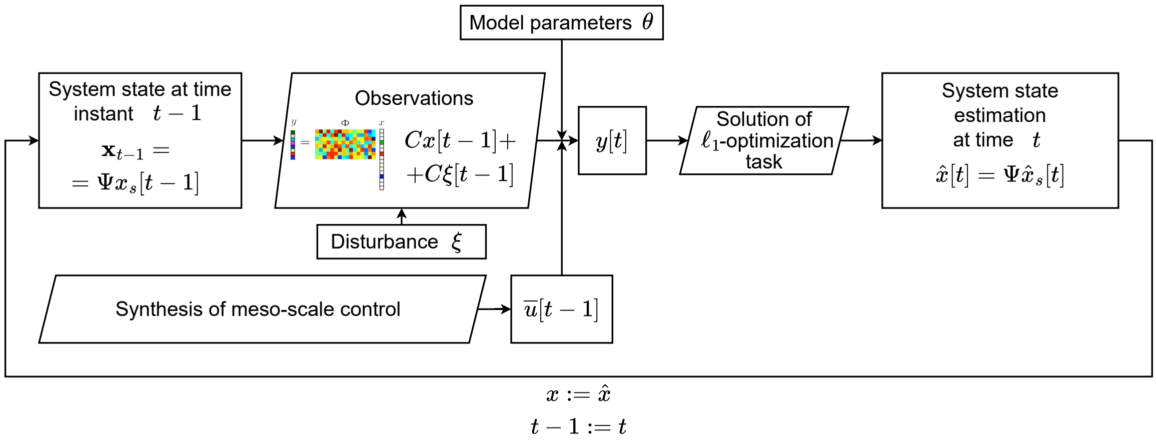

Figure 3.1: meso-scale control algorithm with identification by compression

3.3 Simulation environment for the compressed sensing meso-scale control algorithm

In order to model remote observations of a multi-agent system, dynamic trajectories are “quantized”: the continuous initial state space is transferred to a discrete counting space. The power of the new discrete space depends on the resolution and “field of view” of the observer sensor, which determines the sampling step. In the context of this work, to conduct simulations, the resolution and field of view parameters are selected heuristically based on the specific behavior of the system. To better understand the concept of an observer, we can consider a digital camera or video camera as an example: like filming a scene, the observer sensor records a certain area of space (states) with a certain degree of sampling. At the current moment, the stage of limiting and quantizing the state space is performed heuristically, which gives rise to the problem of optimal automatic selection of parameters for scaling the observation area and selection of the sampling step.

The quantization algorithm consists of three main steps:

1. Limit and quantize the state space (model a remote sensor).

2. Carry out observations with compression of the state vector.
3. Find clusters.
4. If necessary, change the cluster management strategy.

The indicated steps are repeated cyclically, which makes it possible to process streaming data.

3.3.1 Limitation and quantization of dynamic trajectories

Let's consider the simplest example of one-dimensional synchronization outputs $\{\mathbf{z}_t^i\}_{i\in\mathcal{N}}$. In this case, the trajectories can be represented by curves on the "coordinate plane", the horizontal axis of which is the time t, and the vertical axis is $\mathbf{z}$. The half-plane $t > 0$ is called *full state space* and we denote it $\mathfrak{S}$. From the point of view of a remote sensor, the half-plane $\mathfrak{S}$ is a set of time-varying scenes (one-dimensional realizations of agent states). However, since the outputs $\mathbf{z}_t^i$ in the general case can take on any values, it is practically impossible to cover the entire infinitely extending scene with a sensor, and therefore it is necessary to select the area where the trajectories in a specific task are located. This is equivalent to setting the "field of view" of the sensor — limiting $\mathbf{z}$ to the values of $z_{\min}$ and $z_{\max}$.

In addition, to implement the prototype in the current work, the beginning of observations and their end (constraints on t) should be specified — $t_{\min}$ and $t_{\max}$. Thus, for any moment of time $t \in [t_{\min}, t_{\max})$ the trajectory point $\mathbf{z}_t^i$ belongs to $\big[z_{\min}, z_{\max}\big)$. This time limit is not mandatory when implementing the streaming version of the algorithm.

Due to the actions described above in the half-plane $\mathfrak{S}$, it is possible to select a limited region: $\mathfrak{R} = \big[z_{\min}, z_{\max}\big) \times [t_{\min}, t_{\max}) \subset \mathfrak{S}$. The values $z_{\max}$ and $t_{\max}$ do not belong to $\mathfrak{R}$ in order to simplify further constructions.

Further sampling of $\mathfrak{R}$ involves dividing this area into "cells" (sampling):

$$\begin{aligned} \big[z_{\min}, z_{\max}\big) &= \big[z_{\min}, z_1\big) \cup \ldots \cup \big[z_{p-1}, z_{\max}\big), \\ [t_{\min}, t_{\max}) &= [t_{\min}, t_1) \cup \ldots \cup [t_{q-1}, t_{\max}), \end{aligned} \tag{3.8}$$

where the values p and q are determined by sampling steps: spatial resolution of the sensor $z_r = z_i - z_{i-1}$ and exposure length $t_e = t_j - t_{j-1}$. The values of z_r and t_e depend on the shape of the trajectories $\{\mathbf{z}_t^i\}_{i\in\mathcal{N}}$ and are chosen empirically. For example, the time interval t_e must be much shorter than the duration of the dynamic event of interest in order to capture rapid state changes. Such events may involve intra-cluster disturbances or inter-cluster exchange of agents.

Thus, to implement observations of trajectories, a p-pixel one-bit sensor is modeled, taking the value 1 at the location where the trajectory is recorded, otherwise 0. It is convenient to write a series of q observations in the form of a matrix $\mathfrak{B} \in \{0,1\}^{p\times q}$, the columns of which can be compressed in accordance with (3.2). It is clear that when choosing smaller values of z_r and t_e, the dimension of $\mathfrak{B}$ increases, which leads to an increase in the resolution and readability of the corresponding portraits of discrete trajectories, but at the same time the time of compression and reconstruction of observations increases.

For the subsequent search for clusters, the reconstructed matrix $\widehat{\mathfrak{B}}$ of threshold filtering with adaptive threshold adjustment: its elements $\hat{b}^{ij}$ are equated to one if they exceed a certain threshold value, otherwise they are equated to 0. This value can be chosen to be proportional to the average of all elements in the $5\hat{b}^{\cdot j}$ column — as will be shown later in the simulations, this threshold value is a good heuristic.

3.3.2 Compressed sensing

During the simulation process, the recording matrix $\mathfrak{B}$ is filled. Each of its columns, modeling a "snapshot" of the sensor, is compressed by multiplying on the right by a constant measurement matrix: $\mathbf{y} = C\mathbf{x}$, where $\mathbf{y}$ — vector of compressed observations. The matrix C is pre-generated, the elements of which are taken from the normal distribution

$$c^{ij} = \mathcal{N}\left(0, \frac{1}{m}\right).$$

Also, preliminary, based on the average sparsity of the columns $\mathfrak{B}$, the value $m = c \cdot s_0$ is determined, where c takes values from 2 to 4, and s_0 is the average (by all columns) column sparsity $\mathfrak{B}$. Based on the value of m and the dimension of the vector $\mathbf{x}$, equal to p (the number of sensor pixels), the degree of compression can be determined: $\gamma_c = N/s$, where c is the above-mentioned constant .

3.3.3 Cluster indentification

The task of searching for clusters currently comes down to the task of determining the number of clusters found and the location of their centroids. The initial data are binary sparse vectors, which makes it possible to introduce a Euclidean metric, which, based on two non-zero elements of such a vector, gives the modulus of the difference between their indices. The centroid of the cluster in this case is the average value of the indices included in the cluster due to their proximity according to the metric described above.

Hierarchical clustering allows you to select clusters without regard to their actual number, and therefore the predicted number of clusters may not coincide with the true value. Using this fact, you can build a metric to evaluate the accuracy of clustering. Let's introduce the following variable:

$$\chi[k] = \begin{cases} 1, & \text{if } \hat{s}[k] = s, \\ 0, & \text{if } \hat{s}[k] \neq s, \end{cases}$$

where k is the observation number from the series (takes values from 1 to q), $\hat{s}$ is the estimated number of clusters predicted by the algorithm, s is the true number of clusters. In this case, the percentage accuracy for q observations

$$\epsilon = \frac{\sum_{i=1}^{q} \chi[k]}{q} \cdot 100\%. \tag{3.9}$$

In some cases, ℓ_1-reconstruction of compressed state vectors of agents can lead to the appearance of unwanted bursts in places where there are no trajectories. These isolated outliers are usually recognized as small clusters consisting of 1–2 elements. As will be shown below, the accuracy of the cluster detection algorithm can be increased by eliminating such outliers from the general list of found clusters. Everywhere below, the size of the minimal cluster is denoted by $|\mathcal{C}|_{\min}$.

3.3.4 Estimation of complexity

The large dimension of the state space in multi-agent systems leads to the emergence of unpredictable emergent behavior in them. This means that the transfer of data from the micro-scale to the macro-scale can be a kind of bottleneck in the entire process of communication between the system and the observer (data center). Because of this phenomenon, the system turns out to be complex in terms of the intricacy of connections between its components, which leads to the problem of analyzing big data exchanged in the MAS. Thus, the complexity of the proposed algorithm is most correctly assessed in the amount of collected data. Consider column $b^{\cdot j}$ of matrix $\mathfrak{B}$, which is a vector in p-dimensional space. Identification with compression makes it possible to reduce the amount of transmitted data from a multiple of p to a multiple of $s \log(N/s)$, where s is equal to the number of non-zero components of the sparse signal, represented by the vector $b^{\cdot j}$. In the case of large state spaces, recognition with compression makes it possible to expand the mentioned "bottleneck" in proportion to the sparsity, in order to then quickly synthesize cluster control.

3.3.5 Functional requirements

Dynamics of a multi-agent system

To simulate the dynamics of the MAS, first of all, it is necessary to provide the ability to specify the number of agents in the system $0 \leq N \leq 100000$, as well as the graph of connections between them, for example, in the form of a connectivity matrix or an associative array. Since instead of a simulator in the future it will be possible to apply the algorithm to real cyber-physical systems, simulations should be carried out with time measured in seconds — for this it is necessary to provide the ability to set an elementary time step dt, used in the numerical solution of systems of differential equations describing the dynamics agent systems.

In order to simplify testing, the initial states of agents should be set not only by a pseudorandom number generator, but also manually — for example, to test special cases of unstable equilibrium in the system. It should also be possible to precisely set the sensitivity of agents to local and cluster control inputs. In addition, the operation of the system should not depend on the nature of the agent — in this regard, it should be possible to implement agents with the desired behavior.

Remote observations with compression

Remote observations of MAS are similar to the process of photographing with a classic digital camera. In this regard, it is necessary to provide the ability to set the following parameters: the number of "pixels" in the sensor, exposure time, as well as the generalized zoom ratio, expressed in the specified observation limits (in order to simplify the simulation of optics).

For compression in accordance with the "identification by compression" methodology, it is necessary to specify a matrix of observations outside the sensor (which should then also be provided to the data center for reconstruction) and, accordingly, the dimension of the vector of compressed observations. For caching and monitoring, you need to be able to record “snapshots” — original and compressed data from the sensor.

Reconstruction of system data

Since the reconstruction must be carried out in a remote data center, its implementation should be separated from the sensor implementation, allowing them to communicate with each other using some communication channel. At the moment, there is no goal to provide communication between them via the network, so it is enough to transfer the recorded data from the sensor to the data center locally.

For reconstruction, it should be possible to provide the data center with the same observation matrix that was provided to the sensor.

Search for clusters

Searching for clusters involves pre-setting the size of the minimum cluster (to filter out outliers) and a threshold value of the distance between clustered elements. If the distance between two elements exceeds this threshold, then they should be classified into different clusters. Therefore, it should be possible to choose the minimum cluster size and threshold value.

In addition, in addition to determining whether certain elements belong to a certain cluster, the algorithm must determine the position of the cluster centroid based on the coordinates of the elements included in it.

Formation of a cluster management strategy

Ensuring the possibility of forming cluster management based on the configuration of recognized clusters comes down to the requirement that the corresponding strategy can be changed using the software being developed. Thus, the software must be capable of solving a user-specified control task.

3.3.6 Non-functional requirements

The software is intended for installation and use on the user's personal computers. In this regard, the following requirements have been put forward:

- The software must be cross-platform (in particular, run on computers running an OS based on the Linux kernel or Windows 7 and higher);
- when running the software, it requires no more than 8 gigabytes of RAM;
- Software with all used libraries requires no more than 200 megabytes of permanent memory.

3.4 Numerical demonstration of the compressed sensing meso-scale control approach

To test the developed and implemented algorithm, a modified Kuramoto model with cluster control is used. At this stage, the main goal of testing is to test the algorithm for searching clusters using remote observations, as well as to determine the dependence of clustering accuracy on the degree of compression. In this regard, the cluster control action is determined by a predetermined true number of clusters that needs to be predicted using hierarchical clustering. In the future, cluster control action can be synthesized not according to predetermined clusters, but according to predicted ones — at the moment this is beyond the scope of the current work.

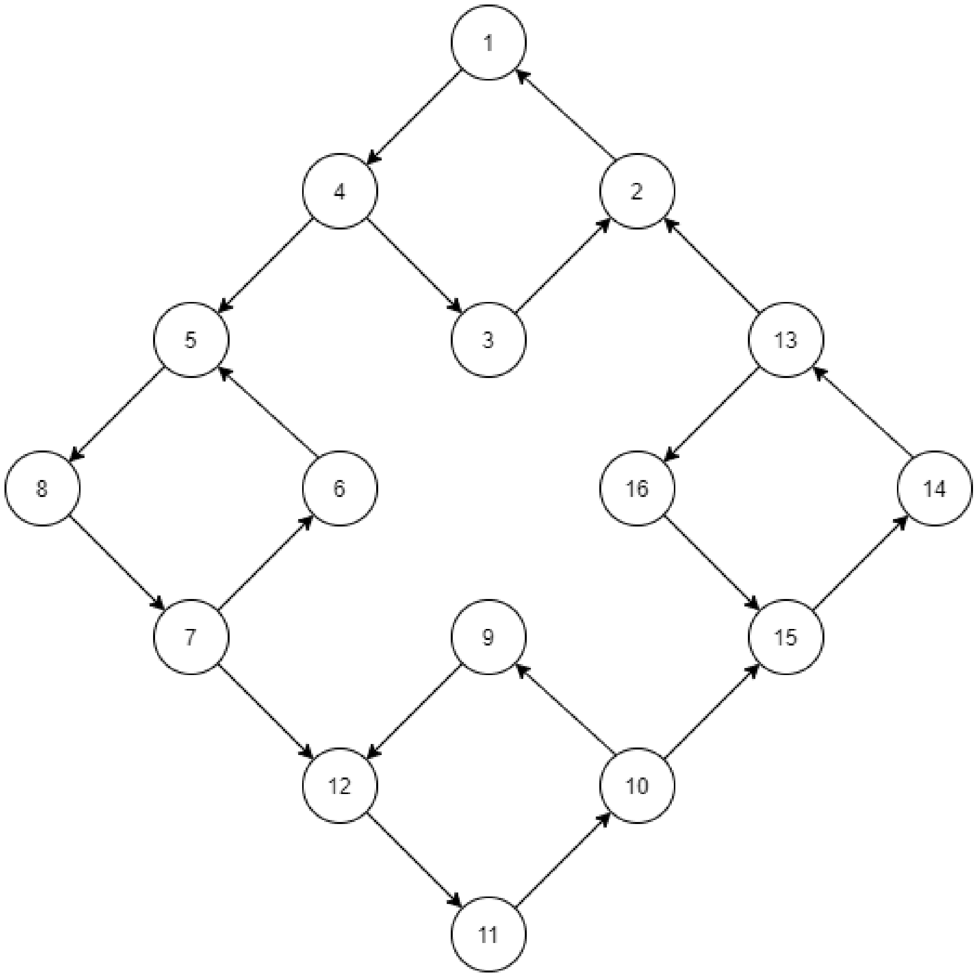

Figure 3.2: Topology of the graph used

3.4.1 Simulation of the Kuramoto oscillator model

Let's consider the model (2.21) and its simulation on the interval $T = [0, 60]$:

$$\mathbf{x}_{t+1}^i = \mu^i \bar{\mathbf{u}}_t^\alpha(\bar{\mathbf{x}}_t^\alpha) + \mathbf{w}^i + \rho \sum_{j=1}^{N} A^{ij} \sin\left(\mathbf{x}_t^j - \mathbf{x}_t^i\right).$$

As an example, let's take the sinusoidal cluster control algorithm:

$$\bar{\mathbf{u}}_t^\alpha(\bar{\mathbf{x}}_t^\alpha) = \sin(2\pi f^\alpha (t - 20)) \tag{3.10}$$

as an example, where $\bar{\mathbf{u}}_t^\alpha$ is activated from time $t = 20$.

Let's set the following model parameters:

- $N = 16$;
- graph topology $\mathcal{G}$ as in Figure 3.2;
- initial phases $\mathbf{x}^i(0)$ from a uniform distribution on the circle S^1;
- values of f^α are taken from a uniform distribution on $[0, 1]$;
- $\rho = 0.5$;
- natural frequencies $\mathbf{w}^i$ are presented in Table 3.1;
- values of μ^i are presented in Table 3.2.

Table 3.1: w^i values used in the simulation

$\mathbf{w}_1$	$\mathbf{w}_2$	$\mathbf{w}_3$	$\mathbf{w}_4$	$\mathbf{w}_5$	$\mathbf{w}_6$	$\mathbf{w}_7$	$\mathbf{w}_8$
4.1	4.2	4.3	4.4	8.1	8.2	8.3	8.4

$\mathbf{w}_9$	$\mathbf{w}_{10}$	$\mathbf{w}_{11}$	$\mathbf{w}_{12}$	$\mathbf{w}_{13}$	$\mathbf{w}_{14}$	$\mathbf{w}_{15}$	$\mathbf{w}_{16}$
12.1	12.2	12.3	12.4	16.1	16.2	16.3	16.4

Table 3.2: μ^i values used in the simulation

μ_1	μ_5	μ_9	μ_{13}	μ_2	μ_6	μ_{10}	μ_{14}
0.375				0.75			

μ_3	μ_7	μ_{11}	μ_{15}	μ_4	μ_8	μ_{12}	μ_{16}
1.125				1.5			

Next, we will determine the cluster synchronization of agent-oscillators in the Kuramoto model by the proximity of the values of their angular frequencies $\dot{\mathbf{x}}^i$, which play the role of determining the synchronization of outputs $\mathbf{z}_t^i$. Simulation of the dynamics of 16 agents allows us to obtain trajectories of changes in their angular frequencies (see Figure 3.3). From the equations (2.34) and (2.35) it follows that the presented parameter values allow the cluster pattern to remain unchanged over time. In Figure 3.3, the trajectories of agents from the same cluster are marked with the same color for clarity. Despite the rather small number of agents, the discrete state space of such a system nevertheless turns out to be quite large in dimension. If

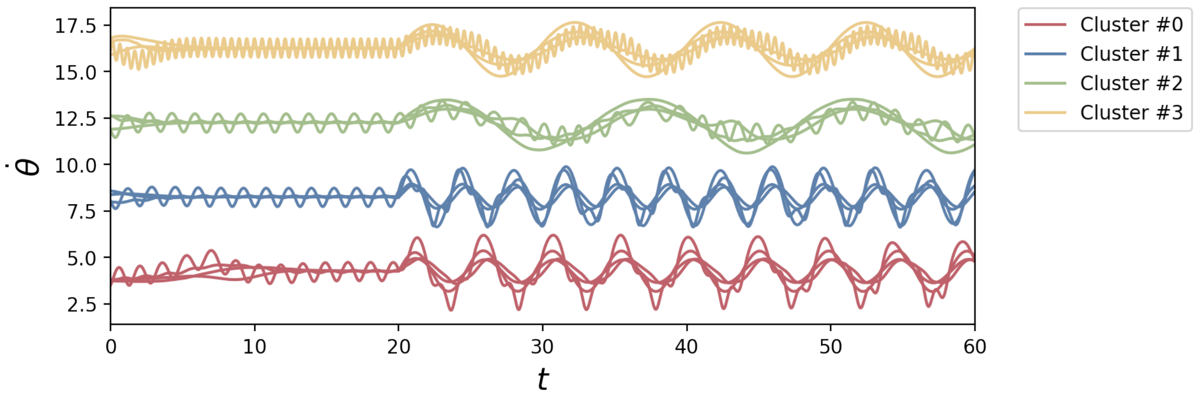

Figure 3.3: Trajectories of angular frequencies of 16 oscillator agents forming 4 clusters under sinusoidal cluster control described in the equation (3.10)

we consider the visualization of this state space further, it becomes clear that such a space can include a much larger number of agents. However, after cluster synchronization, agents will in any case occupy a relatively small area of space (thus reducing the state space to a sparse form). Thus, at the moment, a simple example with 16 agents is used solely to demonstrate the algorithm, while the system remains highly scalable.

3.4.2 Determination of clusters from discrete observations

Let us denote the empirically selected parameters of the observation system from the sensor and recording device (state space sampling parameters):

- $z_{\min} = 0$, $z_{\max} = 20$, $z_r = 0.1$;
- $t_{\min} = 20$, $t_{\max} = 60$, $t_e = 0.1$.

With these parameters, a simulation of a 200-pixel sensor is performed, with the help of which 400 of one-dimensional "snapshots" of dynamic trajectories (angular frequencies) are made over a period of time from 20 to 60 seconds from the beginning of the simulation of the oscillator system. These images can be represented as a matrix $\mathfrak{B}$ of dimension 200×400. Figure 3.4 shows the matrix $\mathfrak{B}$ obtained in the simulation, where white pixels are responsible for the presence of one or more agents in the corresponding region of state space.

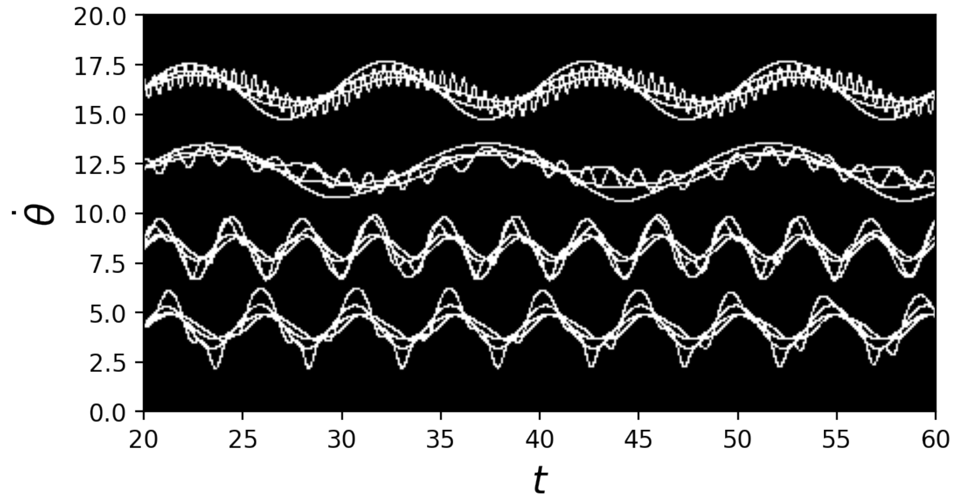

Figure 3.4: 400 one-dimensional sequential 200-pixel snapshots of the trajectories of 16 oscillators, represented as columns of the matrix $\mathfrak{B}$; white pixels are responsible for the presence of one or more agents with the corresponding state

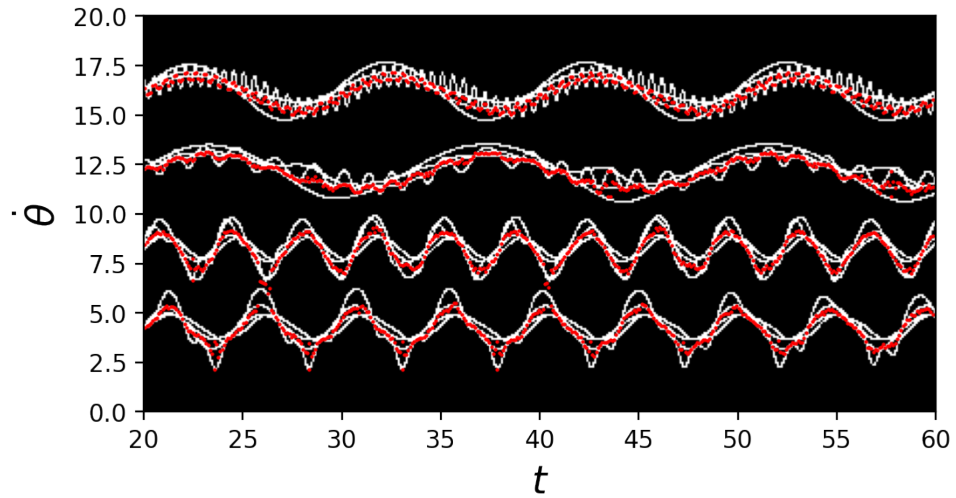

Figure 3.5: Centroids of clusters (highlighted in red) recognized in each snapshot of the state space from a series of 400 snapshots, represented as columns of the matrix $\mathfrak{B}$, with the value of the minimum cluster $|\mathcal{C}|_{\min} = 2$

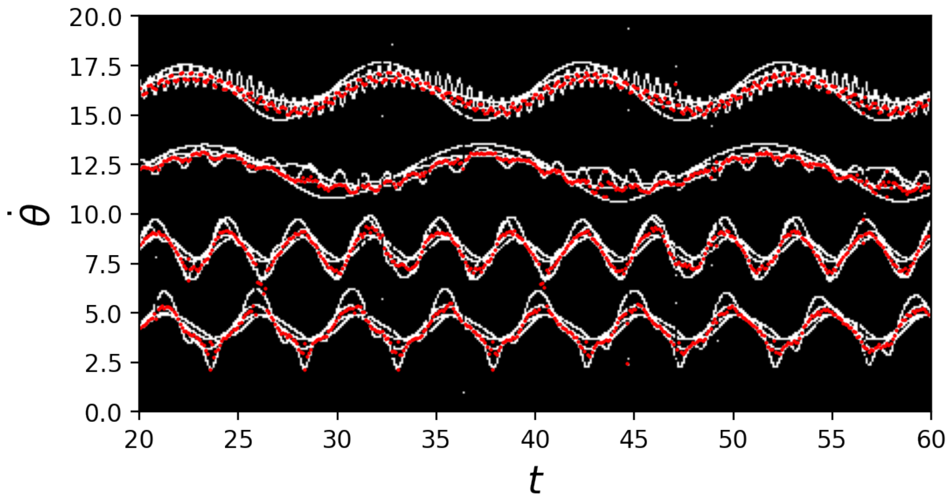

Figure 3.6: Centroids of clusters (highlighted in red) recognized in reconstructed snapshots of the state space from a series of 400 snapshots, presented as columns of the matrix $\widehat{\mathfrak{B}}$, with a minimum cluster value of $|\mathcal{C}|_{\min} = 2$

Next, we should determine the value of m: let it be equal to the integer part of $c \cdot s_0 = 3 \cdot 24.78$, that is, $m = 74$ (where s_0 is the average sparsity value of the columns $\mathfrak{B}$, and c is some constant). The corresponding compression ratio is $\gamma_3 = N/s \approx 2.7$. After compression and reconstruction using the metric (3.9), one can compare the accuracy of finding clusters in the original set of observations (matrix $\mathfrak{B}$) and in observations after compression-reconstruction (matrix $\widehat{\mathfrak{B}}$). Let all clusters in which the number of elements is less than 2 be considered as outliers ($|\mathcal{C}|_{\min} = 2$). Let also everywhere below the value $\delta = 0.8$ be the threshold value for defining clusters (if two non-zero elements of the observation vector are located closer than δ from each other, then they belong to the same cluster). The accuracy comparison results are presented in Table 3.3. The found centroids are indicated in Figures 3.5 and 3.6.

Table 3.3: Clustering accuracy with $c = 3$ and minimum cluster size $|\mathcal{C}|_{\min} =$ 2.

ϵ, %	
$\mathfrak{B}$	$\widehat{\mathfrak{B}}$
95.74	94.24

3.4.3 Study of the dependence of accuracy on the compression ratio

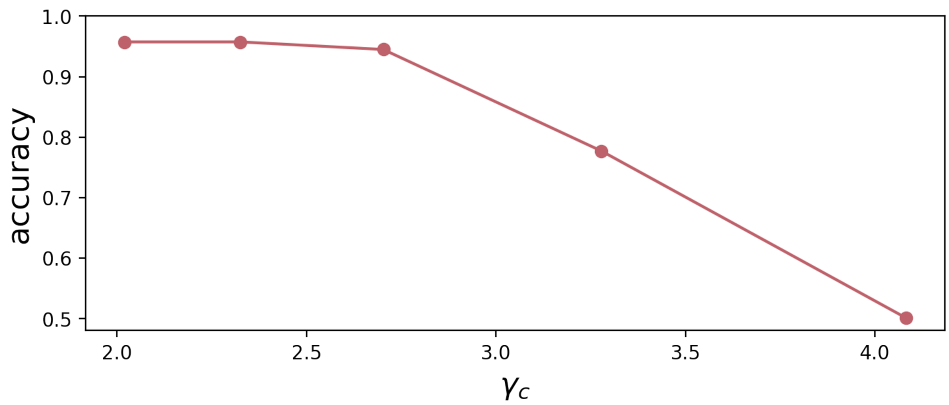

Figure 3.7: Accuracy of cluster identification in $\widehat{\mathfrak{B}}$ for $|\mathcal{C}|_{\min} = 2$ and various values of compression degree γ_c (values of c from 2 to 4 in increments of 0.5)

The accuracy of finding clusters depends on the degree of compression: the higher the degree of compression, the more losses, which means lower accuracy of cluster recognition. Figure 3.7 shows the dependence of the accuracy of determining clusters in $\widehat{\mathfrak{B}}$ for various values of γ_c and $|\mathcal{C}|_{\min} = 2$. It can be seen that the accuracy decreases monotonically with increasing compression ratio.

3.4.4 Study of the dependence of accuracy on the size of the minimum cluster

When the minimum cluster size is $|\mathcal{C}|_{\min} = 1$, any outliers are accepted as a separate cluster. At the same time, for large values of $|\mathcal{C}|_{\min}$ the set of centroids may turn out to be empty. Both situations lead to low clustering accuracy, so it is important to find the optimal minimum cluster size to maximize accuracy. Figure 3.8 shows the dependence of the accuracy of determining clusters in $\widehat{\mathfrak{B}}$ for various values of $|\mathcal{C}|_{\min}$ and $\gamma_{2.5}$ =3.28. From the resulting graph it follows that the optimal value of the minimum cluster is 2.

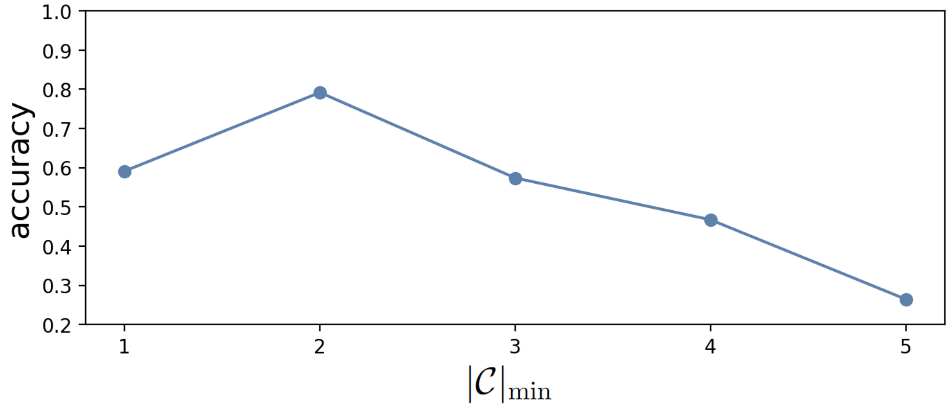

Figure 3.8: Accuracy of cluster identification in $\widehat{\mathfrak{B}}$ at $\gamma_{2.5} = 3.28$ and various values of the minimum cluster $|\mathcal{C}|_{\min}$ (from 1 to 5)

3.5 Numerical demonstration of the robustness of the algorithm in a large-scale system

Simulations were carried out for three systems with different numbers of agents: $N = 1000, 10000, 100000$, while all three systems were clustered into ten coalitions. State values range from 10 to 100 in 10 increments. The number of clusters s and the total number of agents in the system N determine the compression degree m, determined through the logarithm of the number of agents (see equation (3.7)).

The control strategy is synthesized to reduce the aggregated states of the cluster by half at each iteration towards the zero value with minimal effort, which can be modeled by the leading function and the functional

$$\bar{\mathbf{p}}_t^\alpha = \frac{1}{2}\bar{\mathbf{x}}_t^\alpha, \;\; J(\bar{\mathbf{u}}_t^\alpha) = \sum_{\alpha=1}^{|\mathcal{C}^\alpha|} |\bar{\mathbf{x}}_t^\alpha + \bar{\mathbf{u}}_t^\alpha - \bar{\mathbf{p}}_t^\alpha|.$$

For simplicity, the case of $\theta \equiv 1$ is considered below, since the main interest is the accuracy of compression-recovery instead of the features of the linear system. Various types of perturbations ξ were tested: Gaussian zero mean and uniform (an example of an unknown, but limited with non-zero mathematical expectation) distribution. For the described number of agents in the system and the corresponding dimensions of the coding space m, the distances ℓ_2 between the agent states under control without compression and the restored states after meso-scale control in the low-dimensional space are calculated. The results are demonstrated in tables 3.4 (Gaussian perturbation) and 3.5 (uniform perturbation).

Table 3.4: ℓ_2 distances between the reference (in the original space) and restored states of agents after identification by compression with Gaussian perturbations

		Standard deviation of Gaussian perturbations			
N	m	0	0.001	0.01	0.1
1000	184	$2.83 \cdot 10^{-9}$	$2.92 \cdot 10^{-2}$	$3.26 \cdot 10^{-1}$	2.92
10000	276	$2.09 \cdot 10^{-11}$	$8.47 \cdot 10^{-2}$	$8.19 \cdot 10^{-1}$	7.59
100000	368	$3.25 \cdot 10^{-11}$	$2.91 \cdot 10^{-1}$	3.08	27.6

Table 3.5: ℓ_2 distances between the reference (in the original space) and reconstructed states of agents after identification by compression with unknown but limited disturbances

		sup of the support of the uniform distribution function			
N	m	0	0.001	0.01	0.1
1000	184	$4.10 \cdot 10^{-9}$	$1.68 \cdot 10^{-2}$	$1.46 \cdot 10^{-1}$	1.43
10000	276	$4.09 \cdot 10^{-8}$	$4.55 \cdot 10^{-2}$	$4.63 \cdot 10^{-1}$	4.43
100000	368	$1.72 \cdot 10^{-10}$	$1.57 \cdot 10^{-1}$	1.30	13.6

It can be seen that the proposed method provides acceptable accuracy for values of standard deviation (for a Gaussian distribution) or sup (for a uniform distribution) up to 0.01. At higher disturbance intensities, the distance ℓ_2 between the standard and the reconstructed state estimate exceeds the minimum value of the agent's state, equal to 10. In other words, the signal-to-noise ratio becomes too low for a qualitative interpretation of such an assessment. It is worth noting that the highest signal peaks in amplitude (from 30 to 100 arbitrary units) remain distinguishable.

3.6 Summary

The third chapter demonstrates a universal method for compressing sparse real-valued data (or data written as floating point numbers) — "identification by compression". The process of data compression and recovery using randomized transformations and ℓ_1-optimization is described, which forms the basis of the compression identification method. Finally, as the main result of the chapter, the connection between the sparsity of the full state vector of a multi-agent system and its form during clustering is demonstrated.

As a result of three chapters, software was developed and tested that synthesizes meso-scale control for a low-dimensional representation of the state space of the system, transformed under the influence of identification by compression. Testing was carried out on the nonlinear Kuramoto model, as well as on a large-scale multi-agent network with up to 100,000 agents.

Conclusion

In complex multi-agent networked dynamical systems with a large number of atomic structural units—agents—ordered patterns have a possibility to emerge due to interactions with each other and with the environment surrounding the system. First of all, this effect is valid for natural multi-agent systems: physical, consisting of interacting elementary particles or molecules, or biological, such as ensembles of connected neurons or communicating groups of animals. Aside from that, man-made systems are subject to pattern emergence too: robotic or, for example, computing systems. Against the background of active mathematical control theory development in the 20th century, which to this day attracts increasing interest amid specialists in the natural, technical sciences, as well as the humanities, many scientifically and practically significant models of complex systems have been derived. These models describe the above-mentioned emergent structures, as well as prescribe effective methods of influencing the state of these systems. In the dissertation, the diversity of multi-agent models is supplemented by describing clusters of agents that are combined into groups based on the similarity of their states during local interactions or under external disturbances. The description of clustering is supplemented by derived prescriptive rules for synthesizing optimal control focused on controlling groups of agents as a more effective way to influence a complex system. These results are shown to be applicable for a wide class of systems, where they are possible to be described and controlled on so-called "meso-scale" with negligible losses in accuracy. Here, a meso-scale is an intermediate scale of system modeling (in the sense of generalization in its state) compared to the micro- and macro-scales. In addition, an algorithm for meso-scale control of a complex system exploiting the "compressed sensing" approach is proposed in the dissertation. This algorithm is based on the observed fact, stating that clustering and the formation of sparse data in models of the states of complex systems are similar phenomena.

Main scientific results of the dissertation research achieved within the framework of the set tasks:

1) a new approach to modeling information and control processes in complex multi-agent networked dynamical systems is proposed and justified, describing time-varying clustering in dynamic networks of elementary control objects (chapter 1);

2) developed a method for controlling complex multi-agent networked dynamical systems with clustering, in which the synthesis of control action occurs in a space of reduced dimensionality, the effectiveness of the developed method in comparison with classical approaches was demonstrated (chapter 2);

3) developed an approach to encoding sparse information in complex multi-agent networked dynamical systems with clusters based on the "compressed sensing" method, and demonstrated the connection between agent clustering and the sparseness of the system representation (chapter 3).

The results described in the dissertation can form the basis for more in-depth studies of mathematical models of complex systems: for example, when describing a system as a topological space with clusters generalized further as homologies of higher groups (for example, for hole-type patterns in higher-dimensional spaces). The practical contribution of the dissertation could presumably affect development of more effective and feasible methods for controlling swarm robotic systems, as well as to some extent biological and physical systems: networks and ensembles of neurons or physical particles.

Bibliography

[1] Acebron J., Bonilla L., Pérez-Vicente C. et al. The Kuramoto model: a simple paradigm for synchronization phenomena // Reviews of Modern Physics. — 2005. — 04. — Vol. 77.

[2] Acharya A., Adhikary H., Aduszkiewicz A. et al. Measurements of $\pi^{\pm}$, $K^{\pm}$, p and $\bar{p}$ spectra in ^{7}Be+^{9}Be collisions at beam momenta from 19A to 150A GeV/c with the NA61/SHINE spectrometer at the CERN SPS // The European Physical Journal C. — 2021. — 01. — Vol. 81.

[3] Aduszkiewicz A., Andronov E., Anticic T. et al. Measurements of Ξ^{-} and $\overline{\Xi}^{+}$ production in proton–proton interactions at $\sqrt{s_{NN}}$=17.3 GeV in the NA61/SHINE experiment // The European Physical Journal C. — 2020. — 09. — Vol. 80. — P. 833.

[4] Al-Rifaie M. M., Bishop J. Stochastic diffusion search review. — De Gruyter Open Access, 2013. — 01. — Vol. 4. — P. 155–173.

[5] Amelin K., Granichin O., Sergeenko A. et al. Emergent intelligence via self-organization in a group of robotic devices // Mathematics. — 2021. — Vol. 9, no. 12.

[6] Amelin K., Granichin O., Uzhva D. et al. A new method of adaptive mesoscale control in complex multiagent networked dynamical systems // Cybernetics and Physics. — 2022. — 12. — P. 175–189.

[7] Amelin K. S., Amelina N. O., Granichin O. N. et al. Decentralized group control of autonomous robots swarm without data routing // Robotics and Technical Cybernetics. — 2021. — Vol. 9, no. 1. — P. 32–41.

[8] Amelin K. S., Amelina N. O., Granichina O. A. et al. Local voting protocol for packet routing in networks // Proceedings of the XIV-

th Multiconference on Control Problems, Russia. — Vol. 2. — 2021. — P. 31–33.

[9] Amelina N., Fradkov A., Jiang Y. et al. Approximate consensus in stochastic networks with application to load balancing // IEEE Transactions on Information Theory. — 2015. — Vol. 61, no. 4. — P. 1739–1752.

[10] Amelina N. O. Application of local voting protocol for decentralized load balancing of a network with variable topology and measurement noise // Vestnik S.-Petersburg Univ. — 2013. — no. 3.

[11] Amelina N. O., Fradkov A. L. Approximate consensus in the dynamic stochastic network with incomplete information and measurement delays // Automation and Remote Control. — 2012. — Vol. 73, no. 11. — P. 6–29.

[12] Arnold V. I. Ordinary Differential Equations. — Springer, 1997.

[13] Austin C., Kusumoto F. The application of Big Data in medicine: current implications and future directions // Journal of interventional cardiac electrophysiology : an international journal of arrhythmias and pacing. — 2016. — 10. — Vol. 47.

[14] Babanezhad M., Behroyan I., Taghvaie Nakhjiri A. et al. Investigation on performance of particle swarm optimization (PSO) algorithm based fuzzy inference system (PSOFIS) in a combination of CFD modeling for prediction of fluid flow // Scientific Reports. — 2021. — 01. — Vol. 11.

[15] Balharith T., Alhaidari F. Round robin scheduling algorithm in CPU and cloud computing: a review // 2019 2nd International Conference on Computer Applications Information Security (ICCAIS). — 2019. — P. 1–7.

[16] Bandyopadhyay M., Mishra M., Mishra B. et al. A comprehensive study on load balancing algorithms in cloud. — Springer, 2021. — 01. — P. 423–436.

[17] Baraniuk R., Davenport M., DeVore R. et al. A simple proof of the restricted isometry property for random matrices // Constructive Approximation. — 2008. — 12. — Vol. 28. — P. 253–263.

[18] Benedetto D., Caglioti E., Montemagno U. On the complete phase synchronization for the Kuramoto model in the mean-field limit // Communications in Mathematical Sciences. — 2014. — 07. — Vol. 13.

[19] Bullo F. Motion coordination for multi-agent networks // Hybrid Systems: Computation and Control. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2006. — P. 1–1.

[20] Bullo F., Cortés J., Martínez Jorge. Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms. — Princeton University Press, 2009.

[21] Candès E. J., Romberg J., Tao T. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information // IEEE Transactions on Information Theory. — 2006. — Vol. 52, no. 2. — P. 489–509.

[22] Cassandras C. G., Lin X., Ding X. An optimal control approach to the multi-agent persistent monitoring problem // IEEE Transactions on Automatic Control. — 2013. — Vol. 58, no. 4. — P. 947–961.

[23] Chanfreut P., Keijzer T., Ferrari R. M.G. et al. A topology-switching coalitional control and observation scheme with stability guarantees // IFAC-PapersOnLine. — 2020. — Vol. 53, no. 2. — P. 6477–6482. — 21st IFAC World Congress.

[24] Chanfreut P., Maestre J. M., Camacho E. F. A survey on clustering methods for distributed and networked control systems // Annual Reviews in Control. — 2021. — Vol. 52. — P. 75–90.

[25] Chanfreut P., Maestre J. M., Gallego A. J. et al. Clustering-based model predictive control of solar parabolic trough plants // Renewable Energy. — 2023. — Vol. 216. — P. 118978.

[26] Chopra N., Spong M .W. On synchronization of Kuramoto oscillators. — Vol. 2005. — 2006. — 01. — P. 3916–3922.

[27] Chorowski J., Weiss R. J., Bengio S. et al. Unsupervised speech representation learning using wavenet autoencoders // IEEE/ACM Transactions on Audio, Speech, and Language Processing. — 2019. — Vol. 27, no. 12. — P. 2041–2053.

[28] Comparing adaptive and non-adaptive models of cargo transportation in multi-agent system for real time truck scheduling / O. Granichin, P. Skobelev, A. Lada et al. // International Joint Conference on Computational Intelligence. — 2016. — 01. — P. 282–285.

[29] Cooperative control of multi-agent systems: optimal and adaptive design approaches / F. L. Lewis, H. Zhang, K. Hengster-Movric, A. Das. — Springer London, 2014. — ISBN: 9781447155744.

[30] DeGroot M. H. Reaching a consensus // Journal of the American Statistical Association. — 1974. — Vol. 69, no. 345. — P. 118–121.

[31] Donho D. L. Compressed sensing // IEEE Transactions on Information Theory. — 2006. — Vol. 52, no. 4. — P. 1289–1306.

[32] Doyle John C. Robust and optimal control // Proceedings of 35th IEEE Conference on Decision and Control. — Vol. 2. — 1996. — P. 1595–1598.

[33] Egger G., Chaltsev D., Giusti A. et al. A deployment-friendly decentralized scheduling approach for cooperative multi-agent systems in production systems // Procedia Manufacturing. — 2020. — 01. — Vol. 52. — P. 127–132.

[34] Enikeeva L., Potemkin D., Uskov S. et al. Gravitational search algorithm for determining the optimal kinetic parameters of propane prereforming reaction // Reaction Kinetics, Mechanisms and Catalysis. — 2021. — 01. — Vol. 132.

[35] Erofeeva V., Granichin O. Distributed state estimation of moving targets using cyclic simultaneous perturbation stochastic approximation // IFAC-PapersOnLine. — 2018. — Vol. 51, no. 23. — P. 218–223. — 7th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2018.

[36] Erofeeva V., Granichin O., Uzhva D. et al. Adaptive distributed cluster flow control for a group of autonomous robots // IFAC-PapersOnLine. — Vol. 56. — 2023. — P. 9384–9389.

[37] Erofeeva V., Granichin O., Volkovich Z. et al. Multilevel modeling and control of dynamic systems // Scientific Reports. — 2024. — Nov. — Vol. 14, no. 1. — P. 27903.

[38] Erofeeva V., Granichin O., Volodina E. Accelerated decentralized load balancing in multi-agent networks // IEEE Access. — 2024. — Vol. 12. — P. 161954–161967.

[39] Ester M., Kriegel Ha.-P., Sander J. et al. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise // Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. — KDD'96. — AAAI Press, 1996. — P. 226–231.

[40] Falco M., Robiolo G. A systematic literature review in multi-agent systems: patterns and trends // 2019 XLV Latin American Computing Conference (CLEI). — 2019. — 09. — P. 1–10.

[41] Fischer M. J., Lynch N. A., Paterson M. S. Impossibility of distributed consensus with one faulty process // J. ACM. — 1985. — apr. — Vol. 32, no. 2. — P. 374—-382.

[42] Flasiński M. Automata-based multi-agent model as a tool for constructing real-time intelligent control systems // From Theory to Practice in Multi-Agent Systems / Ed. by Barbara Dunin-Keplicz, Edward Nawarecki. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2002. — P. 103–110.

[43] Fradkov A. L. Cybernetical Physics. — Springer, 2007.

[44] Galbraith G. N., Vinter R. B. Lipschitz continuity of optimal controls for state constrained problems // SIAM J. Control and Optimization. — 2003. — 01. — Vol. 42. — P. 1727–1744.

[45] Goldstein J. Emergence as a construct: history and issues // Emergence. — 1999. — 03. — Vol. 1. — P. 49–72.

[46] Gorodetski V. I., Granichin O. N., Skobelev P. O. Cluster flow control in distributed network systems // Proceedings of the XIV-th Russian Multiconference on Management Problems. — 2021. — P. 130–132.

[47] Gorodetski V. I., Karsayev O. V., Samoilov V. V. Multi-agent data fusion technology // Tr. SPIIRAN. — 2002. — Vol. 1, no. 2. — P. 12–37.

[48] Granichin O., Erofeeva V. Cyclic stochastic approximation with disturbance on input in the parameter tracking problem based on a multiagent algorithm // Automation and Remote Control. — 2018. — Jun. — Vol. 79, no. 6. — P. 1013–1028.

[49] Granichin O., Uzhva D. Invariance preserving control of clusters recognized in networks of Kuramoto oscillators // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). — Vol. 12412. — LNAI, 2020. — P. 472–486.

[50] Granichin O., Uzhva D. Cluster control of complex cyber-physical systems // Cybernetics and Physics. — 2021. — Vol. 10. — P. 191–200.

[51] Granichin O., Uzhva D., Volkovich Z. Cluster flows and multiagent technology // Mathematics. — 2021. — 12. — Vol. 9. — P. 22.

[52] Granichin O., Volkovich Z., Toledano-Kitai D. Randomized Algorithms in Automatic Control and Data Mining. — Springer Berlin Heidelberg, 2015.

[53] Granichin O. N., Pavlenko D. V. Randomizationofdataacquisitionandℓ_1-optimization, (recognition with compression) // Autom. Remote Control. — 2010. — Vol. 71, no. 11. — P. 2259–2282.

[54] Guimaraes Dayan A., Floriano Giovanni H. F., Chaves Lucas S. A tutorial on the CVX system for modeling and solving convex optimization problems // IEEE Latin America Transactions. — 2015. — Vol. 13, no. 5. — P. 1228–1257.

[55] Horita A. Y., Loubach D. S., Bonna R. Analysis and identification of possible automation approaches for embedded systems design flows // Information. — 2020. — Vol. 11, no. 2.

[56] Jadbabaie A., Motee N., Barahona M. On the stability of the Kuramoto model of coupled nonlinear oscillators. — 2005. — 05. — P. 4296–4301.

[57] Kálmán R. E. A new approach to linear filtering and prediction problems // Transactions of the ASME–Journal of Basic Engineering. — 1960. — Vol. 82, no. Series D. — P. 35–45.

[58] Karp R. Reducibility among combinatorial problems // Complexity of Computer Computations. — Vol. 40. — 1972. — 01. — P. 85–103.

[59] Khantuleva T. A. Mathematical Modeling of Shock-Wave Processes in Condensed Matter. — Springer, 2022.

[60] Kotwal T., Jiang X., Abrams D. Connecting the Kuramoto model and the chimera state // Physical Review Letters. — 2017. — 12. — Vol. 119.

[61] Leonard N. E. Multi-agent system dynamics: bifurcation and behavior of animal groups // IFAC Proceedings Volumes. — 2013. — Vol. 46, no. 23. — P. 307–317. — 9th IFAC Symposium on Nonlinear Control Systems.

[62] Lodge P. Leibniz's mill argument against mechanical materialism revisited // Ergo, an Open Access Journal of Philosophy. — 2014. — Vol. 1, no. 3.

[63] Lu W., Atay F. Stability of phase difference trajectories of networks of Kuramoto oscillators with time-varying couplings and intrinsic frequencies // SIAM Journal on Applied Dynamical Systems. — 2018. — 01. — Vol. 17. — P. 457–483.

[64] Lyapunov A. The general problem of the stability of motion // International Journal of Control. — 1992. — Vol. 55, no. 3. — P. 531–534.

[65] Lynch N. A. Distributed Algorithms. — San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1996.

[66] Machidon A. L., Pejović V. Deep learning for compressive sensing: a ubiquitous systems perspective // Artificial Intelligence Review. — 2022. — Vol. 56, no. 4. — P. 3619–3658.

[67] Mattioni M., Monaco S. Cluster partitioning of heterogeneous multi-agent systems // Automatica. — 2022. — Vol. 138. — P. 110136.

[68] Medvedeva T., Sysoev I., Sysoeva M. et al. Dynamical mesoscale model of absence seizures in genetic models // PLOS ONE. — 2020. — 09. — Vol. 15, no. 9. — P. 1–23.

[69] Mohammed A., Furbach U. Multi-agent systems: modeling and verification using hybrid automata // Programming Multi-Agent Systems / Ed. by Lars Braubach, Jean-Pierre Briot, John Thangarajah. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2010. — P. 49–66.

[70] Montbrió E., Pazó D., Schmidt J. Time delay in the Kuramoto model with bimodal frequency distribution // Physical review. E, Statistical, nonlinear, and soft matter physics. — 2006. — 12. — Vol. 74. — P. 056201.

[71] Mukherjee D., Ghose D. Generalized hierarchical cyclic pursuit // Automatica. — 2016. — Vol. 71. — P. 318–323.

[72] Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. — 2009.

[73] Nedic A., Ozdaglar A. Distributed subgradient methods for multi-agent optimization // IEEE Transactions on Automatic Control. — 2009. — Vol. 54, no. 1. — P. 48–61.

[74] Nemirovski A., Todd M. Interior-point methods for optimization // Acta Numerica. — 2008. — 05. — Vol. 17. — P. 191–234.

[75] Nesterov Y., Todd M. Primal-dual interior-point methods for self-scaled cones // SIAM Journal on Optimization. — 1998. — 02. — Vol. 8. — P. 324–364.

[76] Notarstefano G., Egerstedt M., Haque M. Containment in leader–follower networks with switching communication topologies // Automatica. — 2011. — Vol. 47, no. 5. — P. 1035–1040.

[77] Olfati-Saber R. Flocking for multi-agent dynamic systems: algorithms and theory // IEEE Transactions on Automatic Control. — 2006. — Vol. 51, no. 3. — P. 401–420.

[78] Olfati-Saber R., Fax J. A., Murray R. M. Consensus and cooperation in networked multi-agent systems // Proceedings of the IEEE. — 2007. — Vol. 95, no. 1. — P. 215–233.

[79] Ongaro D., Ousterhout J. In search of an understandable consensus algorithm // Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference. — USENIX ATC'14. — USA : USENIX Association, 2014. — P. 305—-320.

[80] Oyelade J., Isewon I., Oladipupo F. et al. Data clustering: algorithms and its applications // 2019 19th International Conference on Computational Science and Its Applications (ICCSA). — 2019. — 07. — P. 71–81.

[81] Pasqualetti F., Zampieri S., Bullo F. Controllability metrics, limitations and algorithms for complex networks // IEEE Transactions on Control of Network Systems. — 2014. — Vol. 1, no. 1. — P. 40–52.

[82] Pease III M. C., Shostak R. E., Lamport L. Reaching agreement in the presence of faults // J. ACM. — 1980. — apr. — Vol. 27, no. 2. — P. 228—-234.

[83] Pinto S. C., Andersson S. B., Hendrickx J. M. et al. Multiagent persistent monitoring of targets with uncertain states // IEEE Transactions on Automatic Control. — 2022. — Vol. 67, no. 8. — P. 3997–4012.

[84] Proskurnikov A., Granichin O. Evolution of clusters in large-scale dynamical networks // Cybernetics and Physics. — 2018. — 11. — Vol. 7, no. 3. — P. 102–129.

[85] Rashid S., Razak S. Big Data challenges in 5G networks // 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN). — 2019. — 07. — P. 152–157.

[86] Ravazzi C., Dabbene F., Proskurnikov A. et al. Learning hidden influences in large-scale dynamical social networks: a data-driven sparsity-based approach, in memory of roberto tempo // IEEE Control Systems Magazine. — 2021. — Vol. 41, no. 5. — P. 61–103.

[87] Ravve E. V., Volkovich Z., Weber G.-W. A uniform approach to incremental automated reasoning on strongly distributed structures // Global Conference on Artificial Intelligence, GCAI 2015, Tbilisi, Georgia, October 16-19, 2015. — Vol. 36 of EPiC Series in Computing. — EasyChair, 2015. — P. 229–251.

[88] Ren W., Beard R. W. Consensus seeking in multiagent systems under dynamically changing interaction topologies // IEEE Transactions on Automatic Control. — 2005. — Vol. 50, no. 5. — P. 655–661.

[89] Ren W., Beard R. W. Distributed consensus in multi-vehicle cooperative control. — Springer, 2008. — Vol. 27.

[90] Requicha A. Nanorobots, NEMS, and nanoassembly // Proceedings of the IEEE. — 2003. — 12. — Vol. 91. — P. 1922–1933.

[91] Ribarics P. Big Data and its impact on agriculture // Ecocycles. — 2016. — 07. — Vol. 2. — P. 33–34.

[92] Rossi F., Bandyopadhyay S., Wolf M. et al. Review of multi-agent algorithms for collective behavior: a structural taxonomy // IFAC-PapersOnLine. — 2018. — Vol. 51, no. 12. — P. 112–117. — IFAC Workshop on Networked Autonomous Air Space Systems NAASS 2018.

[93] Rossi F., Rodrigues G., Calheiros R. et al. Dynamic network bandwidth resizing for big data applications // 2017 IEEE 13th International Conference on e-Science (e-Science). — 2017. — 10. — P. 423–431.

[94] Sadilek M., Thurner S. Physiologically motivated multiplex Kuramoto model describes phase diagram of cortical activity // Scientific Reports. — 2014. — 09. — Vol. 5.

[95] Seryakov A., Uzhva D. Convolutional neural network for centrality determination in fixed target experiments // Physics of Particles and Nuclei. — 2020. — Vol. 51, no. 3. — P. 331–336.

[96] Shehadeh H. A hybrid sperm swarm optimization and gravitational search algorithm (HSSOGSA) for global optimization // Neural Computing and Applications. — 2021. — 03.

[97] Shimizu Y., Haga H. Solving partial differential equations using multi-agent modelling // AIP Conference Proceedings. — 2019. — 12. — Vol. 2184, no. 1. — P. 060018.

[98] Shtessel Y., Edwards C., Fridman L. et al. Sliding Mode Control and Observation. — Springer New York, 2014.

[99] Silva E. DG-ABC: an integrated multi-agent and cellular automata urban growth model. — Information Science Reference; 1st edition, 2014. — 01. — P. 57–92.

[100] Singh S. Study of K-means and enhanced K-means clustering algorithm // International Journal of Research in Computer Science. — 2021. — 03. — Vol. 4. — P. 103–107.

[101] Strogatz S. H. Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering. — Westview Press, 2000.

[102] Tahmassebi A., Ehtemami A., Mohebali B. et al. Big data analytics in medical imaging using deep learning // Big Data: Learning, Analytics, and Applications. — 2019. — 05. — P. 13.

[103] Tharwat A., Schenck W. A conceptual and practical comparison of PSO-style optimization algorithms // Expert Systems with Applications. — 2021. — Vol. 167. — P. 114430.

[104] Tsitsiklis J. N., Bertsekas D. P., Athans M. Distributed asynchronous deterministic and stochastic gradient optimization algorithms // IEEE Transactions on Automatic Control. — 1986. — Vol. 31, no. 9. — P. 803–812.

[105] Utkin V. I. Sliding modes and their applications in systems with variable structure. — Nauka, 1974.

[106] Uzhva D. Compressed sparse magnetometric data sensing // Proceedings of the XXIV Russian Conference of Young Scientists “Navigation and Traffic Control”'. — 2022. — P. 140–142.

[107] Uzhva D., Granichin O., Granichina O. Compressed cluster sensing in multiagent IoT control // Proceedings of 61st IEEE Conference on Decision and Control. — Cancun, Mexico, 2022. — P. 3580–3585.

[108] Uzhva D. R., Granichin O. N., Granichina O. A. Cluster flow control in distributed network systems // Proceedings of the XIV-th Russian Multiconference on Management Problems. — 2021. — P. 130–132.

[109] Vamvoudakis K. G., Jagannathan S. Control of Complex Systems: Theory and Applications. — Butterworth-Heinemann; 1st edition, 2016. — 07.

[110] Van den Berg E., Friedlander M. P. Probing the Pareto frontier for basis pursuit solutions // SIAM Journal on Scientific Computing. — 2008. — Vol. 31, no. 2. — P. 890–912.

[111] Van den Berg E., Friedlander M. P. SPGL1: A solver for large-scale sparse reconstruction. — 2019. — December.

[112] Vicsek T. nd Zafeiris A. Collective motion // Physics Reports. — 2012. — Vol. 517, no. 3. — P. 71–140.

[113] Vilasini V., Nurgalieva N., del Rio L. Multi-agent paradoxes beyond quantum theory // New Journal of Physics. — 2019. — Vol. 21, no. 11. — P. 113028.

[114] Vinyals O., Babuschkin I., Czarnecki W. et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning // Nature. — 2019. — 11. — Vol. 575.

[115] Vittikh V. A., Skobelev P. O. Multiagent Interaction Models for Constructing the Needs-and-Means Networks in Open Systems // Automation and Remote Control. — 2003. — Vol. 64, no. 1. — P. 162–169.

[116] Wang Q., Lin D., Yang P. et al. An energy-efficient compressive sensing-based clustering routing protocol for wsns // IEEE Sensors Journal. — 2019. — Vol. 19, no. 10. — P. 3950–3960.

[117] Welikala S., Cassandras C. G. Distributed nonconvex optimization of multiagent systems using boosting functions to escape local optima // IEEE Transactions on Automatic Control. — 2021. — Vol. 66, no. 11. — P. 5175–5190.

[118] Wooldridge M. An Introduction to Multiagent Systems. — 2 edition. — Chichester, UK : Wiley, 2009.

[119] Yuan Y., Liu W., Wang T. et al. Compressive sensing-based clustering joint annular routing data gathering scheme for wireless sensor networks // IEEE Access. — 2019. — Vol. 7. — P. 114639–114658.

[120] Zaied Y., Saad W., Shokair M. Energy efficient of grid clustering based TEEN protocol for cognitive radio wireless sensor networks // 2021 International Conference on Electronic Engineering (ICEEM). — 2021. — P. 1–6.