A NEW METHOD OF ADAPTIVE MESOSCALE CONTROL IN COMPLEX MULTIAGENT NETWORKED DYNAMICAL SYSTEMS

Konstantin Amelin, Oleg Granichin, Anna Leonova, Vikentiy Pankov, Denis Uzhva
Laboratory "Control of Complex Systems" IPME RAS
Faculty of Mathematics and Mechanics, Science Educational Center,
"Mathematical Robotics and Artificial Intelligence"
St. Petersburg State University, St. Petersburg, Russia
konstantinamelin@gmail.com, o.granichin@spbu.ru, annia.leonova@gmail.com
vkpankov@gmail.com, d.uzhva@spbu.ru

Victoria Erofeeva
Laboratory "Control of Complex Systems"
IPME RAS
St. Petersburg, Russia
eva@ipme.ru

Vladislav Ershov
Faculty of Mathematics and Mechanics
St. Petersburg State University
St. Petersburg, Russia
st055883@student.spbu.ru

Article history:
Received 28.11.2022, Accepted 14.12.2022

Abstract

Centralized strategies applied to large-scale systems require a vast amount of computational and communication resources. In contrast to them, distributed strategies offer higher scalability and reliability. However, communication and coordination among agents tremendously impact performance of systems controlled in the distributed manner. The existing methods lead to clustering, where the coordination between agents is limited to groups of entities to be controlled. The size of these groups are usually known in advance. In turn, many systems exhibit self-organization and dynamically form clustering structure. In that sense, control methods should adapt to such dynamic structures offering the same balance between performance and communication/computational demands. In this paper, we propose a new approach to complex system control based on efficient cluster (mesoscopic) control paradigm. We demonstrate its efficacy in scenarios, where a group of agents should reach a certain goal.

Key words

control of complex systems, multiagent technologies, dynamical systems, adaptive control, mesoscale control

1 Introduction

During the recent decades, we have witnessed a tendency to gathering a deluge of data generated by heterogeneous sources spatially distributed across wide geographical areas. The examples include vehicle and traffic systems, autonomous robotic networks, and others. Centralized strategies applied to such large-scale systems require a vast amount of computational and communication resources. In contrast to centralized approaches, where a single controlling unit has full information about the system, distributed strategies rely on system decentralization. These strategies are usually applied to local controllers also referred to as agents that control different subsystems.

Distributed strategies offer higher scalability and reliability. However, communication and coordination among agents tremendously impact performance of systems controlled in the distributed manner. Therefore, researches have started exploring strategies that seek a balance between optimality and coordination efforts influenced by communication and computational demands [Dörfler et al., 2014; Yang et al., 2020]. The new line of works has led to system partitioning problem, in which the overall system is decomposed into several subsystems and control inputs are assigned to some agents within the subsystems. By other words, the agents are arranged into clusters that determine their actions using intra-area communication, i.e., they do not communicate with agents outside their cluster [Fele et al., 2017]. Other works study sparsity-promoting control strategies that reduce a set of communication links by generating a sparse structure of controller matrices [Furieri et al., 2020]. Furthermore, hierarchical architecture may reduce the communication demand and provide good system performance. In hierarchical control, the first level consists of nodes in a group pursuing its leader within the group. In the second level, the weighted centroid of a group chases the weighted centroid of its leading group. A similar extension holds for more levels. It was shown that hierarchical control leads to an increased rate of convergence [Mukherjee and Ghose, 2016]. Besides the trade-off between system performance and resource demands, some papers focus on handling the system failures occurring due to switching topologies, communication delays, and unpredicted malfunctions [Schiffer et al., 2017].

The described methods lead to clustering, where the coordination between agents is limited to groups of entities to be controlled. The size of these groups are usually known in advance or/and adjusted to system performance. In turn, many systems exhibit self-organization and dynamically form clustering structure [Granichin and Uzhva, 2021]. In that sense, control methods should adapt to such dynamic structures offering the same balance between performance and communication/computational demands.

Estimating cluster structure in a system still remains a relevant task nowadays. During the last two decades, advances in randomized signal acquisition strategies allowed to treat data compression in a novel efficient way. For example, the compressed sensing methodology [Candes et al., 2006; Donoho, 2006] opened opportunities for data acquisition with compression "on the fly" by utilizing consecutive randomized sums of the incoming signal. This approach utilizes signal sparsity for recovery from compressed codes, which is innate for most of the natural or artificial meaningful signals. In our related work [Granichin and Uzhva, 2022], it was shown how compressed sensing can be embedded in cluster control of complex multiagent systems. Due to the clusterization, state vector of the system exhibits sparse features, which are then used in a lower-dimensional system representation and corresponding cluster control. We further develop this idea in the current work

Our contribution is threefold. First, we develop a new approach to complex system control, based on efficient cluster (mesoscopic) control paradigm. We propose a novel framework for complex multiagent system analysis, which aims to utilize system sparsity for cluster control synthesis. Second, with the help of the compressed sensing methodology, a new distributed cluster control algorithm is developed. By receiving sparse system features in a latent space, it is shown how these features can be exploited to synthesize efficient control action. And third, a universal hardware module for evaluation of aggregated cluster characteristics is proposed. A possible approach to implementing a hardware device for efficient cluster control computation according to the proposed theory is described.

The rest of the paper is organized as follows. In Section 2, a general approach to complex systems modeling and cluster synchronization regime is described. Next, Section 3 provides a description of the proposed cluster flows control framework. Section 4 consequently develops the theoretical background revealed in the previous ones by illustrating a novel approach to cluster control using the compressed sensing methodology. Then, in Section 5, a distributed protocol for application of the compressed-sensing-based control among agents in a system is described. Next, Section 6 unifies the proposed approach in the form of an algorithm. The following Section 7 describes a hardware implementation of the proposed algorithm. And finally, Section 8 provides numeric simulations, which is then followed by a conclusion.

2 Processes of Formation of Emergent (appearing) Artificial Intelligence

First, we need to provide a definition of a complex system, for further motivation of complex systems study to become clear. Next, two ways of system modeling are described, with corresponding pros and cons of both mentioned. Finally, we discuss the possible peculiarities of complex systems dynamics (stability and clusterization), which emerge due to internal and external processes, relatively to the system.

2.1 Understanding of Complex Systems

As we study patterns and their evolution in time, we may find that some generalization and corresponding dynamic analysis problems may appear exceptionally challenging. For example, consider the task of forecasting, which requires us to predict the weather in future. As we dive into this problem, we may find that near future weather W_F^1 (about a few hours) is quite similar to the present one W_{PR}, so that future can be thought as of slightly transformed present:

$$W_F^1 = T(W_{PR}) \approx W_{PR},$$

where T is the corresponding transformation. However, as we progress through time by stacking these transformations ($T^n(W_{PR})$, where T repeatedly applied to W_{PR} n times, corresponding to a forecast for multiple weeks ahead), future

$$W_F^n = T^n(W_{PR})$$

becomes less and less similar to present; moreover, in case present weather would be slightly different $\widetilde{W}_{PR}$ comparing to W_{PR} (i.e. $\widetilde{W}_{PR} \approx W_{PR}$), then stacking the same transformations for $\widetilde{W}_{PR}$ would result in some future weather

$$\widetilde{W}_F^n = T^n(\widetilde{W}_{PR}),$$

which may differ significantly from W_F^n. In other words, $\widetilde{W}_{PR} \approx W_{PR}$ does not necessarily lead to $\widetilde{W}_F^n \approx W_F^n$. This is due to the nature of T, which is governed by the way various internal components and external disturbances (e.g. air molecules, the Sun) of the climate system *interact* between each other, forming a *networked* system. The amount of the components and interconnections between them leads to inability of deriving and computing precise model of T using *limited resources* and *obvious reasoning*. Given the example above, we can now define a complex system as a composition of a large number of simple interacting elements, placed in some environment, with which they also perpetually interact. This leads to the resulting overall system behavior to be unpredictable (by classical approaches) on distant time horizon, given simple behavior of its components—a special property, which we call *emergence*.

With the above being said, we can outline two main branches to dealing with complexity:

1. Quantitative complexity reduction: apply unjustified amount of resources.
2. Qualitative complexity reduction: apply unconventional study approaches.

Nowadays, quantitative approaches tend to become infeasible, due to the resource limitations mentioned above. Thus, we aim to develop advanced mathematical approaches to complex system modeling.

2.2 Complex Systems Modeling

As it was noted in Introduction, dynamic way of modeling allows for intuitively clear representation of systems, that evolve in time. Three approaches, which we will discuss further, are of the highest interest:

1. Discrete (automata) modeling [Ravazzi et al., 2021; Li et al., 2020; Silva, 2014].
2. Continuous dynamical systems modeling [Strogatz, 2000; Arnold and Silverman, 1987; Gazi and Fidan, 2007; Proskurnikov and Granichin, 2018; **?**; Granichin et al., 2020b; Fradkov, 2007].
3. Field theory modeling [Hu et al., 2019; Luo et al., 2021].

According to the first method, a complex system is composed of finite number of elementary autonomous units (further called *agents*), each of them having their individual state, which changes iteratively in time. A corresponding system model can be expressed in a difference equation; as an example, consider an autoregressive model:

$$x_i[t] = c + \sum_{k=1}^{K} \theta_k x_i[t-k] + w_i[t], \qquad (1)$$

where $x_i[t]$ is the state of an agent i at time t, c is a constant, $\theta_1, \ldots, \theta_p$ are parameters, and $w_i[t]$ is a stochastic disturbance. At $p = 1$, we obtain a so-called Markov process, which in fact is subject to lack of memory due to $x_i[t]$ being dependent only on its previous iteration $x_i[t-1]$. Model (1) can be augmented by adding non-linearity and inter-connections between multiple neighbor agents. Moreover, probabilistic variations of the automata models based on the Markov chain formalism can be considered [Li et al., 2020]. We also include Poincaré and Lorenz maps [Strogatz, 2000], notwithstanding such models are conventionally associated with dynamical systems. Summing up, automata approach is convenient for describing stochastic discrete processes (therefore any discrete dynamical system can be viewed as an automata system), which is relevant in the context of large-scale systems with a numerous number of digital agents.

Despite the automata modeling utilizes iterative approach to describe system evolution and, accordingly, such models may have straightforward implementation, cyber-physical systems may often continuously depend on time. A corresponding dynamical systems approach is thus favorable, as it allows to express a complex system (composed of N agents) dynamics using ordinary differential equations:

$$\dot{x}_i(t) = f_i(x_i(t), u_i(t), w_i(t)), \qquad (2)$$

where $x_i(t) \in \mathbb{R}^{n_i}$ is a so-called state vector of an agent $i \in \mathcal{N} = \{1, \ldots, N\}$ (n_i is the number of variables required to describe the agent state), $u_i(t)$ is a local control action, and $w_i(t) \in \mathbb{R}^{m_i}$ is a stochastic disturbance. Moreover, this method of systems modeling is not prone to lack of memory phenomenon mentioned above, due to more freedom available over choosing time intervals $x_i(t)$ defined on. Continuous dynamical systems are also easy for precise analytical analysis,

Finally, field theory approach may be regarded as the most generalized one (perhaps derived from Equation (2) in the limit $N \to \infty$), since it is able to model a continuum of agents:

$$\dot{x}(\lambda, t) = f(\lambda, x(\lambda, t), u(\lambda, t), w(\lambda, t)), \qquad (3)$$

where agents are now “enumerated” (or rather localized) by a point λ in $\mathbb{R}$ or its equinumerous subset (i.e. also with the cardinality of the continuum).

Modern cybernetics is mainly focused on discrete and continuous dynamical system modeling, while field theory modeling appears too early to discuss about. Consequently, we further focus on dynamical systems with finite number agents, without losing a hope that the developed theory can be generalized into model (3).

2.3 Clusterization

In Equation (2), control input u regulates the behavior of the system by setting the rules of the agent state change, depending on the states of neighbor agents and other environmental factors. Artificial complex systems are often required to change its global state in a controllable manner, so that the rules u should lead to the specified global goal. Aiming to study control inputs u, we first should provide a clear description of possible goals, with corresponding models describing such goals rigorously. In [Fradkov, 2007], five types of goals are defined:

1. Stabilization (bringing the all agent states x to their corresponding constant state vectors x_*):

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

2. Tracking (bringing agent states to some function $x_*(t)$, perhaps different for each agent):

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

3. Excitation of oscillations:

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

for some scalar function $G(x)$.

4. Synchronization (matching all agent states):

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

5. Limit set modifications (qualitative changes to the system, e.g. modifications of bifurcation types).

However, this classification is primarily applicable to quite simple systems, primarily single-element ones. As for the multiagent systems, synchronization-type goals are usually of the most interest, since they relate to pattern emergence and complexity reduction possibilities. Indeed, in case all agent states converge to a single synchronous stable manifold, the whole system can then be controlled as one bunch of equally behaving components, thus requiring a single control input.

Recently, in the related works [Proskurnikov and Granichin, 2018; Granichin and Uzhva, 2020; Granichin et al., 2020b] it was noticed that many artificial (and natural) complex systems exhibit so-called *cluster synchronization* (also referred to as clusterization), according to which agents synchronize in groups: system components from one group synchronize, while the ones belonging to different groups do not. For example, cluster synchronization occurs in human brain activity, assuming a brain can be accurately represented by a non-linear coupled oscillators model [Sadilek and Thurner, 2014]. According to the research provided in these articles, cluster synchronization mainly emerges in systems with incomplete connectivity between agents and due to external disturbances, which may affect connectivity and agent states.

In the phenomenon of clusterization, multiple synchronous stable manifolds form (or exist in our models), corresponding to separate clusters. We denote the number of such cluster manifolds m, and the following relation between the number of agents N and number of clusters m is often true:

$$N \gg m \gg 1. \tag{4}$$

Equation (4) motivates the need to study clusterization phenomenon in complex systems for simple cluster control strategies development. In the current paper, we study the ways of cluster synchronization application to efficient system control strategy synthesis.

3 Macro-, Micro- and Meso-scale Control Strategies

In the previous Section, different types of goals for artificial complex system control were described. It was stated that both the specific type of a goal and the desired terminal system state values are regulated by the nature of the control input u. Current Section reveals possible approaches to the control input synthesis, regardless of the goal.

3.1 Open-loop vs Feedback Control

Amid the most simple yet straightforward ways to model and implement control action u would be to construct a corresponding function $u_i(t)$ for each agent i, which only depends on time. For example, consider an ordinary linear system [Kalman, 1960]

$$\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t), \tag{5}$$

where A_i and B_i are some matrices of an appropriate dimensionality, and $u_i \not\equiv 0$. By introducing such control function u, we obtain a non-homogeneous system, the state x_i of which changes independently on any functions of its current state, except the linear one. We further call such control approach a program control or *open-loop* strategy, to emphasize its independence on the system state.

Frequently, in synchronization-type goals it is not enough for the control input u to only depend on time, like in Equation (5). As an example, local voting [Amelina, 2013] and Kuramoto oscillator models [Acebron et al., 2005; Sadilek and Thurner, 2014; Benedetto et al., 2014; Chopra and Spong, 2006; Jadbabaie et al., 2005] demonstrate interesting complex behavior, provided corresponding control actions appear in relatively simple form due to the permission to use system state for control synthesis. Therefore, in contrast to the class of control inputs described above, we define a *feedback* (thus emphasizing its dependence on state) control strategy as a class of control functions $u_i(\boldsymbol{x}, t)$, which now also depend on a set of agent states $\boldsymbol{x}$. This set may contain the state vector x_i and, for instance, state vectors of its neighbors (agents, which affect the agent i).

Real world systems are often subject to various disturbances, which may render agent states inaccurate for further feedback control. In [Proskurnikov and Granichin, 2018; ?], a concept of an observation was thoroughly discussed: basically, we assume that the precise agent state values may be unavailable, while we can only rely on a measurement procedure

$$y_j(\boldsymbol{x}(t), v_i(t)), \tag{6}$$

called an observation, with $\boldsymbol{x}(t)$ being a set of agent states (over which the observation is performed), and v_i being the measurement error due to noise. Using this procedure, one can synthesize a control input $u_i(\boldsymbol{y}, t)$, where $\boldsymbol{y}$ is a set of all necessary observations of the corresponding agents.

Accordingly, we distinguish three classes of control strategies:

1. Open-loop control.
2. State feedback control.
3. Observation feedback control.

3.2 Optimal Control

Another classification scheme of control strategies can be described in terms of the control action *feasibility*. Indeed, one is unable to apply infinite force to instantly achieve the desired system state; such force would always be limited by the abilities of the force actor: $|u(x, t)| < \infty$. The same reasoning is true for the system states $|x(t)| < \infty$ and observed outputs $|y(x, t)| < \infty$. Often, even stronger conditions on such functions may be regarded: for example, in [Galbraith and Vinter, 2003; Hernandez and Garcia, 2014] Lipschitz continuous control inputs are studied, where a function $x(t)$ is called Lipschitz continuous in case there exists a constant $K > 0$ (so-called Lipschitz constant) such that

$$|x(t_1) - x(t_2)| \leq K|t_1 - t_2|$$

is true for all real t_1 and t_2 (or for all t_1 and t_2 on the time interval under consideration). In other cases (e.g. in optimal control as optimization in Hilbert space), it might be required that x, y or u are bounded in $L_p(0, T)$ sense:

$$\int_0^T |u(t)|^p dt < \infty$$

for $u(t)$ as an example, where $T > 0$ is terminal time (finite or infinite), up to which the system operates.

Recently, stability analysis and optimization techniques for control synthesis are of the most interest. The former seeks for such classes of control u, which lead to stable system states (or for ways to check if a given input u leads to stable system states) [Lyapunov, 1892; Jadbabaie et al., 2005]. At the same time, the objective of the latter study is to find such u, which would lead the system to a stable manifold at the fastest rate [Kalman, 1960; Doyle, 1996; Le and Mendes, 2008]. Despite these two branches of complex system control study were initially clearly separated, nowadays they are closely related to each other. More than that, they became significantly more popular recently due to enough computational power available for numerical optimization methods, which are the only feasible solution to optimization for systems of significant complexity, intractable by analytical approaches.

In the situation of limited resources, we can generally minimize one of the following values $J(u)$ for optimal control strategy synthesis.

1. Time consumption for convergence to the stable manifold, provided the control input is bounded:

$$\tilde{u}_1 = \arg\min_u J_1(u) = \arg\min_u T(x, u), \tag{7}$$

   where T is total time consumption as a function of systems state x and bounded control input u from a set of strategies $\mathfrak{U}$.
2. Conversely, control input, expressing "expended efforts", given the time is fixed and limited:

$$\begin{gathered}\tilde{u}_2 = \arg\min_u J_2(u) = \\ = \arg\min_u \int_0^T \phi(x(t), u(x, t), t) dt,\end{gathered} \tag{8}$$

   where ϕ is some function of systems state x, control input u and time t; minimization goes for u amid a set of strategies $\mathfrak{U}$, a proper strategy should therefore be chosen for the integral of ϕ over t and with respect to $x(t)$ to reach its minimum.

We thus obtain two classes of control strategies, classified by optimality, further referred to as Equations (7) and (8). As an example, recall the system (5). In [Kalman, 1960] (for the number of agents $N = 1$), control action $u(x, t) = -Kx(t)$ called static-state feedback was shown to be optimal according to Equation (8).

In presence of noise, modifications to the optimization functionals become needed. In [Granichin and Fomin, 1986; Jerray et al., 2021], "minimax" control strategies are discussed, which first maximize the functional value with respect to the unknown or noise parameters, prior to minimization over the control inputs. In other words, only worst-case scenarios are considered.

It is worth noting that in many complex system control problems it is not possible to obtain an optimal control strategy analytically. In these cases, iterative gradient methods [Kelley, 1960; Polyak, 1964; Liang et al., 2020] allow to find an optimal solution in the following form:

$$u_{k+1} = u_k - \gamma_k \nabla_k J(u_k),$$

where γ_k regulates the gradient descent speed.

3.3 Control on Different Scales

We further consider only synchronization-inducing and tracking control actions, as we deal with large-scale multiagent complex systems. Due to the numerous amount of possible computation steps required to synthesize the desired optimal control strategies, we again return to the basic concept of information and signal compression. In [Proskurnikov and Granichin, 2018; **?**; Granichin et al., 2020b], first attempts to generalize the theory of complex multiagent system control, which utilizes clusterization, were proposed. It was shown how clusterization could reduce the number of required control inputs by the relation between the number of agents N and clusters $m \ll N$. According to the results proposed in these papers, we distinguish three scale classes:

1. Local (microscopic) control, different for each agent.
2. *Cluster (mesoscopic)* control, different for all separate clusters.
3. Global (macroscopic) control, equal for all agents.

We summarize possible scale classes by a scheme of the corresponding observation feedback control model with agent clusterization, see Figure 1.

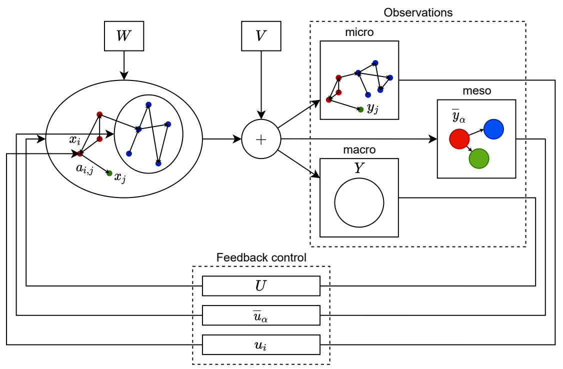

Figure 1. Observation feedback control for a multiagent complex system with clusterization. A system is represented as a composition of interacting agents, defined by their states x_i and connected according to an adjacency matrix A with elements $a_{i,j}$ equal to 1 in case a connection between agents i and j exists, otherwise equal to 0. The system is affected by external disturbances W, which may change connectivity or agent states directly. Observations are exposed to noise V and are divided into three levels: micro- (individual agents), macro- (clusters) and mesoscopic (whole system). Provided the observations, control is separated into three types of inputs: u_i Is local control input of an agent i, $\overline{u}_\alpha$ is a cluster control of a cluster α, and U is global control action equal for all agents.

Cluster control is possible for some integral (aggregated) cluster characteristics as separate cluster states, where clusters are assumed as separate subsystems. These characteristics can be obtained using, for example,

1. Local voting [Amelina, 2013].
2. "External" (i.e. non-multiagent) cluster identification methods, such as hierarchical clustering [Giordani et al., 2020; Karna and Gibert, 2022] or centroid-based clustering [Mughnyanti et al., 2020; Singh, 2021].
3. Data compression techniques, e.g. compressed sensing [Candès et al., 2006; Granichin et al., 2020b].

3.4 Stability of Cluster Control

While cluster control may lead to efficient and straightforward results, it still deals with complex systems with possibly unpredictable emergent behavior. This feature requires to study stability of such systems under control inputs. We further treat system stability in terms of cluster structure invariance, as in [Granichin and Uzhva, 2020]. An example of system analysis directed to invariance preserving cluster control synthesis is shown on the example of a non-linear model below.

We utilize a simple yet versatile nonlinear model proposed by Yoshiki Kuramoto [Acebron et al., 2005]. It describes oscillatory dynamics of coupled oscillators. Given a network of N agents each having one degree of freedom (often called a phase of an oscillator), its dynamics is described by the following system of differential equations:

$$\dot{\theta}_i(t) = w_i + \sum_{j=1}^{N} K_{ij} \sin\left(\theta_j(t) - \theta_i(t)\right), \qquad (9)$$

where $\theta_i(t)$ is a phase of an agent i, K_{ij} is a weighted adjacency matrix of the network and w_i is a natural frequency. According to [Benedetto et al., 2014], [Chopra and Spong, 2006] and [Jadbabaie et al., 2005], agents approach the state of frequency ($\dot{\theta}_i = \dot{\theta}_j \; \forall i, j \in \overline{1, N}$) or phase ($\theta_i = \theta_j \; \forall i, j \in \overline{1, N}$) synchronization under certain conditions on w_i and K_{ij}.

We propose an approach to treating the Kuramoto model of coupled oscillators, which is quite peculiar for cybernetics and control theory. Consider the model (9). The coupling protocol is as follows:

$$u_i(t) = w_i + \sum_{j=1}^{N} K_{ij} \sin\left(\theta_j(t) - \theta_i(t)\right), \qquad (10)$$

so that the corresponding communication outputs of the agents are $y_i(t) = \theta_i(t)$. It is assumed that $K = \{K_{ij}\}$ is an adjacency matrix for a specific configuration of a network of oscillators, i.e. $K_{ij} \neq 0 \iff j \in \mathcal{N}_i \iff \exists (j \to i) \in \mathcal{E}(t)$; and $\theta_i(t) \in S^1 \; \forall i \; \forall t \geq 0$. Synchronization in the Kuramoto model can appear in the forms of frequency or phase lock. The difference between them is the choice of the output $z_i(t)$: $z_i(t) = \dot{\theta}_i(t)$ and $z_i(t) = \theta_i(t)$ correspondingly. We only consider the first case, since it is more general and has more practical applications.

Unlike many other works on the Kuramoto model, we do not restrict ourselves to mean-field coupling $K_{ij} = \frac{C}{N}\ \forall i,j$, where C is some constant. Indeed, the real physical world has more examples of networks with incomplete (or even sparse) graph topologies, i.e. networks of neurons, flocks of birds or even sometimes swarms of robots. More than that, the topology of the graph $\mathcal{G}(t)$ corresponding to a certain multi-agent network may change with time. With that in mind, we propose the following modification to the model (9):

$$\dot{\theta}_i(t) = w_i + \rho \sum_{j\in\mathcal{N}_i(t)} \sin\left(\theta_j(t) - \theta_i(t)\right), \qquad (11)$$

where ρ is a constant and i is only affected by the agents from $\mathcal{N}_i(t)$. we denote an adjacency of $\mathcal{G}(t)$ by $\Upsilon_{ij}(t) \in \{0,1\}\ \forall i,j\ \forall t \geq 0$, so that $K_{ij}(t) = \rho\Upsilon_{ij}(t)$ now becomes dependent on time. The value 0 can be interpreted as "no connection from j to i ($j \notin \mathcal{N}_i$)" and 1 stands for "i is accessible to j ($i \in \mathcal{N}_j$)".

Since agents tend to synchronize using the sum in the protocol of (11), we assume that $\rho > 0$. However, the agents also drift with speeds $w_i \geq 0$, which are basically their natural frequencies.

3.4.1 Mesoscopic Control We assume that there exists some algorithm $\mathfrak{A}$ that, given a multi-agent network $\mathcal{N}$, returns the corresponding clustering at any given moment of time. Such algorithm, for example, can be the new proposed compressed-sensing-based clusterization method described in the next Section. Let clustering $\mathcal{M}(t_1)$ emerge at time t_1 for the model (11) and remain constant on interval $T = [t_1, +\infty)$. Henceforth, $t \in T$. Assume that topology of $\mathcal{G}(t)$, corresponding to a given multi-agent network, also does not change on T. Thus, Υ_{ij} is also constant. We propose the following modification to the original Kuramoto model, assuming $i \in \mathcal{M}_\alpha$:

$$\begin{gathered}\dot{\theta}_i(t) = \mu_i\mathcal{F}_\alpha(t, \overline{x}_\alpha(t)) + \\ +w_i + \rho\sum_{j=1}^{N} \Upsilon_{ij} \sin\left(\theta_j(t) - \theta_i(t)\right),\end{gathered} \qquad (12)$$

where $\mathcal{F}_\alpha(\cdot)$ is a mesoscopic function in a sense it is equal for the whole cluster $\mathcal{M}_\alpha$, μ_i is (a constant) agent's *sensibility* to the control function $\mathcal{F}_\alpha(\cdot)$. If we compare the model (12) with equation (2), it becomes clear how the coupling protocol $u_i(t)$ and the mesoscopic control input $U_i(t)$ are separated:

$$\begin{gathered}u_i(t) = w_i + \rho\sum_{j=1}^{N} \Upsilon_{ij} \sin\left(\theta_j(t) - \theta_i(t)\right), \\ U_i(t) = \mu_i\mathcal{F}_\alpha(t, \overline{x}_\alpha).\end{gathered} \qquad (13)$$

Besides time t, an additional argument in $\mathcal{F}_\alpha(\cdot)$ is $\overline{x}_\alpha(t)$. It stands for characteristics of the cluster α with *physical* nature, i.e. a position of the cluster in space.

The control inputs (13) allows agents to synchronize only if certain conditions are satisfied. As it was discussed for drift, cluster synchronization depends on the values of μ_i: some agents in a cluster $\mathcal{M}_\alpha$ may react to $\mathcal{F}(\cdot)$ with much greater intensity, which may affect the overall structure of the cluster.

In order to find conditions for the parameters in (12) sufficient for cluster structure to remain invariant, we firstly propose a theorem for the model (11) concerning relations between the natural frequencies w_i and values $K_{ij} = \rho\Upsilon_{ij}$ necessary for cluster synchronization.

Theorem 1. *Consider a multi-agent network corresponding to* (11) *and to some graph $\mathcal{G}$ with an adjacency matrix Υ. Let $t \in T$, output $z_i(t) = \dot{\theta}_i(t)$ and $\Delta_{ij}(t) = |z_i(t) - z_j(t)|$. The following conditions are sufficient for this network to be output* $(0,0)$*-synchronized:*

1. *For $i, j \in \mathcal{M}_\alpha$ such that $w_j - w_i \geq 0$*

$$w_j - w_i \leq \rho \sin\left(\frac{\Delta\theta_{ji}}{2}\right) \sum_{l=1}^{N} [\Upsilon_{il} + \Upsilon_{jl}], \qquad (14)$$

where $\sin\left(\frac{\Delta\theta_{ji}}{2}\right) = 1$ in case $\Upsilon_{ij} = \Upsilon_{ji} = 0$; otherwise,

$$\begin{gathered}\sin\left(\frac{\Delta\theta_{ji}}{2}\right) = \max\Bigg\{\sqrt{1 - (\Gamma_i(j))^2}, \\ \sqrt{1 - (\Gamma_j(i))^2}, \frac{\sqrt{2}}{2}\Bigg\},\end{gathered} \qquad (15)$$

where

$$\Gamma_i(j) = \frac{-d_i(j) + \sqrt{(d_i(j))^2 + 8(\Upsilon_{ij} + \Upsilon_{ji})^2}}{4(\Upsilon_{ij} + \Upsilon_{ji})}. \qquad (16)$$

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

$$|w_i - w_j| > 0. \qquad (17)$$

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

The idea behind Theorem 1 is that if $|w_i - w_j|$ is very high, there may appear to be not enough strength of coupling, so that ρ should be appropriately large to "overcome" drift. In the simplest case, where $w_i = w\ \forall i$, the synchronization always appear $\forall \rho > 0$. We obtain that $|w_i - w_j|$ should be non-zero for agents i and j from different clusters for $(0,0)$-synchronization to remain. The proof for the Theorem 1 is proposed in [Granichin and Uzhva, 2020].

Accordingly, the result can be generalized for the model (12). We denote $\mathcal{F}_\alpha = \mathcal{F}_\alpha(t, \overline{x}_\alpha)$ for the sake of notation simplicity.

Theorem 2. *(Stable cluster control for the Kuramoto system). Consider a multi-agent network corresponding to* (12). *Let* $t \in T$, *output* $z_i(t) = \dot{\theta}_i(t)$ *and* $\Delta_{ij}(t) = |z_i(t) - z_j(t)|$. *Let also* $\mathcal{F}_\alpha$ *does not depend on* θ_i $\forall i$. *The following conditions are sufficient for this network to be output* $(0, 0)$*-synchronized.*

1. *In case* $i, j \in \mathcal{M}_\alpha$,
$$|(\mu_j - \mu_i)\mathcal{F}_\alpha| \leq 2\rho \sin\left(\frac{\Delta\theta_{ji}}{2}\right) \sum_{l=1}^{N} [\Upsilon_{il} + \Upsilon_{jl}], \tag{18}$$
where $\Delta\theta_{ji}$ *is as in Theorem 1 (including the case* $\Upsilon_{ij} = \Upsilon_{ji} = 0$*).*
2. *For* $i \in \mathcal{M}_\alpha$, $j \in \mathcal{M}_\beta$, $\alpha \neq \beta$
$$|w_i - w_j + \mu_i \mathcal{F}_\alpha(t, \overline{x}_\alpha) - \mu_j \mathcal{F}_\beta(t, \overline{x}_\beta)| > 0. \tag{19}$$
3. *Graph* $\mathcal{G}$ *is strongly connected.*

Proof. The sufficient conditions can be derived from Theorem 1 by substitution of the mesoscopic control U_i in the Δ_{ij}:

$$\Delta_{ij} = \Bigg| w_i - w_j + \mu_i \mathcal{F}_\alpha - \mu_j \mathcal{F}_\beta + \rho \cdot \\ \cdot \left(\sum_{l=1}^{N} \Upsilon_{il} \sin(\theta_l - \theta_i) - \sum_{l=1}^{N} \Upsilon_{jl} \sin(\theta_l - \theta_j) \right) \Bigg|. \tag{20}$$

First, we assume that $i, j \in \mathcal{M}_\alpha$. Following the same reasoning as in the Proof for Theorem 1, equation (20) is equal to 0, thus (18) can be easily derived. Now let $i \in \mathcal{M}_\alpha, j \in \mathcal{M}_\beta$ $\alpha \neq \beta$, thus $\Delta_{ij} > 0$ in equation (20). Setting the sines to zero as in Theorem 1, the desired condition on the mesoscopic control can be easily derived, which concludes the proof.

4 Finding the Clustering Structure

This section will describe the approach of obtaining the cluster structure of the entire system. This approach is based on the compressed sensing methodology for compact representation of the agent state and transmission over the network. Compressed data is used in the cluster evaluation process, that is, the data recovery stage is not applied.

The goal of this approach is to reduce the computational costs of reconstructing the cluster structure and at the same time eliminate centralization in data collection and calculations. This will allow incomplete measurements of the agent's state to be processed, complementing them with a consensus protocol.

4.1 Compressed Sensing

Assume that a sampled signal $x \in \mathbb{R}^N$ is s-sparse in some *sparsifying* basis (domain) $\Psi \in \mathbb{R}^{N \times N}$, namely

$$x = \Psi x_s, \tag{21}$$

where x_s has, at most, s ($s \ll N$) non-zero components. We further call such vectors s-sparse. It means that the comprehensive information is stored in only s units of data out of N; in other words, x has only s principal components. We define an $m \times N (m \ll N)$ matrix A as a measurement operator, transforming the initial sparse signal from $\mathbb{R}^N$ into $\mathbb{R}^m$. Subsequently, the compressed sensing can be described as:

$$y = Ax = A\Psi x_s = \Phi x_s, \tag{22}$$

where Φ is an $m \times N (m \ll N)$ sampling matrix. The vector $y \in \mathbb{R}^m$ is called a measurement vector or a vector of compressed observations.

Since $m \ll N$, the problem of x estimation by given y is ill-conditioned. However, according to [Candes and Romberg, 2005], reconstruction is feasible if the following conditions for Φ called the Restricted Isometry Property (RIP) are satisfied:

$$(1 - \delta_s)||x_s||_2^2 \leq ||\Phi x_s||_2^2 \leq (1 + \delta_s)||x_s||_2^2, \tag{23}$$

holding for all s-sparse vectors z for some δ_s between 0 and 1. Roughly speaking, matrix Φ should retain lengths of sparse vectors.

Along with (23), another frequently used condition is offered in [Candes et al., 2006], the Modified Restricted Isometry Property (MRIP), which allows for x reconstruction:

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

for some $0 < \lambda < \infty$ and any non-zero vector x_s with s non-zero components.

According to [Candes et al., 2006], RIP can be satisfied with high probability in case the elements of A are randomly sampled according to one of the three following distributions:

1. Gaussian distribution:
$$a[i, j] \sim \mathcal{N}\left(0, \frac{1}{m}\right).$$
2. Symmetrical Bernoulli distribution:
$$P(a[i, j] = \pm 1/\sqrt{m}) = \frac{1}{2}.$$
3. Three-element Bernoulli distribution:
$$a[i, j] = \begin{cases} +\sqrt{3/m} & \text{with the probability } \frac{1}{6} \\ 0 & \text{with the probability } \frac{2}{3} \\ -\sqrt{3/m} & \text{with the probability } \frac{1}{6}. \end{cases}$$

By designing A as outlined above and according to RIP (23),

$$m \geq c_1 s \log(N/s) \tag{24}$$

at $0 < \delta < 1$. The relation (24) ensures the measurement operator A would satisfy RIP with probability $\geq 1 - 2e^{-c_2 m}$, where c_1 and c_2 are small positive constants depending only on δ. In [Baraniuk et al., 2008], particular conditions for the selection of the constants c_1 and c_2 are suggested. Under these conditions, A is universal (for all three types of distributions used to generate its elements) in the sense that any s-sparse x can be reconstructed for given y of an appropriate dimensionality.

In general, an inverse problem concerning the direct one (22) would have an infinite number of solutions. With the sparsity-inspired constraints proposed in [Candes and Romberg, 2005; Donoho, 2006], unambiguous signal reconstruction is feasible in a constrained setup:

$$\min ||x_s||_{\ell_0} \quad \text{s.t.} \quad ||\Phi x_s - y||_2 = 0,$$

where $|| \cdot ||_{\ell_0}$ is the ℓ_0 norm, which counts the number of non-zero elements and thus corresponds to the sparsest solution. However, this problem is NP-hard, and a linear relaxation via the ℓ_1 norm provides a good compromise between the sparsity and the computational complexity [Candes and Romberg, 2005]. In [Granichin and Uzhva, 2022], a comprehensive review of methods to compute an optimized solution is provided. Briefly, these methods vary from the classical iterative interior-point algorithms (see [Nesterov and Todd, 1998]) to more advanced deep learning techniques (e.g. [Zhang et al., 2019]).

4.2 Randomized Compressed Measurements

Each agent has state $x^i \in \mathbb{R}^d$. Let $\mathbf{x} = \text{col}(x^1, \dots, x^N) \in \mathbb{R}^{Nd}$ be the overall system state. The compressed sensing adjusted to the corresponding dimensions of $\mathbf{x}$ takes the form:

$$\bar{\mathbf{y}} = A\mathbf{x}, \quad A = A \otimes \mathbf{I}_d \in \mathbb{R}^{md \times Nd}, \tag{25}$$

where $\mathbf{I}_d$ is the identity matrix, A is $m \times N (m \ll N)$ matrix A as a measurement operator, transforming the initial sparse signal from $\mathbb{R}^N$ into $\mathbb{R}^m$.

We assume that each agent $i \in \mathcal{N}$ independently collects private measurements as follows:

$$y^i = A^i x^i, \tag{26}$$

where $y^i \in \mathbb{R}^{md}$ is the compressed observation of agent i, $A^i = A_{(\cdot,i)} \in \mathbb{R}^{md \times d}$ is a measurement operator of agent i, $A_{(\cdot,i)}$ represents the columns of matrix A corresponding to i-th agent. Thus, we can obtain the overall vector of measurements in the following way:

$$\bar{\mathbf{y}} = \sum_{i \in \mathcal{N}} y^i = \sum_{i \in \mathcal{N}} A^i x^i, \tag{27}$$

where $\bar{\mathbf{y}}$ is a set of centroids, computed as weighted sums of agent states with randomized weights.

4.3 Distributed Compressed Cluster Recovery

The following algorithm is proposed for recovering of clusters from compressed measurements:

1. Train generative neural network $\mathcal{F}$ on simulated dataset with the loss function:
$$\sum_{l=1}^{L} \left\| \mathcal{F}(A\bar{\mathbf{y}}^{(l)}) - \mathbf{x}^{(l)} \right\|^2,$$
where L is the batch size, $\mathbf{x}^{(l)}$ is an example from training dataset, A is the sampling matrix.
2. Reconstruct $\widehat{\mathbf{x}}$ by $\bar{\mathbf{y}}$ using pretrained model: $\widehat{\mathbf{x}} = \mathcal{F}(\bar{\mathbf{y}})$, where $\widehat{\mathbf{x}}$ is an estimate of $\mathbf{x}$.
3. Fit Gaussian mixture model Γ with s components to $\widehat{\mathbf{x}}$.
4. compute the mean and covariance matrix for each cluster using model Γ.
5. Identify agent's cluster membership with the closest center using nearest neighbour algorithm.

Obtaining a vector $\bar{\mathbf{y}}$ in a distributed way is a place for variation in the application of methods. This article discusses one of them, this is the method of local voting protocol (LVP) [Amelina et al., 2015].

The simple experiment is conducted to to verify the performance of the algorithm for simulated states of agents. The architecture of the generative network is depicted on Fig. 3. It is fitted to simulated data with five clusters.

We show the result of an experiment on the Fig. 2 to demonstrate accuracy of the algorithm for clusters recovery.

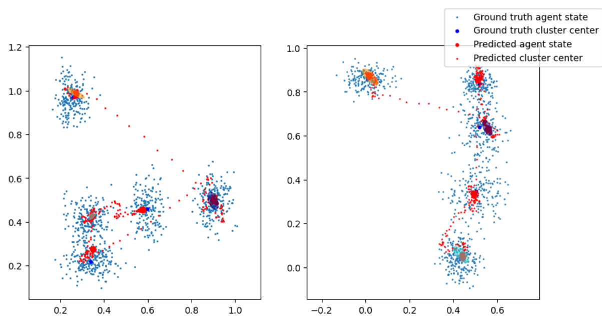

Figure 2. Example of clusters recovery

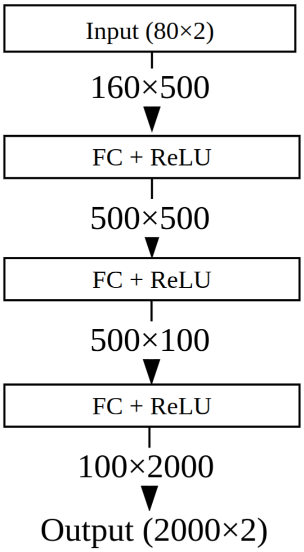

Figure 3. Architecture of the generative model for reconstruction of the state vector of agents. FC is fully connected (linear) layer, ReLU is nonlinear activation function.

5 Distributed Data Aggregation Protocol

In this section, we present a communication protocol suitable for data aggregation in networks. Our goal is to design a protocol tolerant to network unreliability during the aggregation. These unreliability factors include:

- communication failures due to time-varying topology;
- presence of communication noise and delays;
- limited bandwidth of communication channels.

First, we provide a network model that we use and state the required assumptions. Next, we describe the proposed protocol and its properties.

5.1 Network Model

Given a network consisting of n agents. Let communication between agents be described by the directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, \dots, n\}$ is a set of vertices and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is a set of edges. A subgraph of $\mathcal{G}$ is a graph $\bar{\mathcal{G}} = (\mathcal{N}_{\bar{\mathcal{G}}}, \mathcal{E}_{\bar{\mathcal{G}}})$, where $\mathcal{N}_{\bar{\mathcal{G}}} \subseteq \mathcal{N}$ and $\mathcal{E}_{\bar{\mathcal{G}}} \subseteq \mathcal{E}$. Denote by $i \in \mathcal{N}$ an identifier of i-th agent and $(j, i) \in \mathcal{E}$ if there is a directed edge from agent j to agent i. The latter means that agent j is able to transmit data to agent i. For an agent $i \in \mathcal{N}$, the set of *neighbors* is defined as $\mathcal{N}^i = \{j \in \mathcal{N} : (j, i) \in \mathcal{E}\}$. Here, the identifier of i-th agent is used as a superscript and not as an exponent.

Let $c^{i,j} > 0$ be the weight associated with the edge $(j, i) \in \mathcal{E}$ and $c^{i,j} = 0$ whenever $(j, i) \notin \mathcal{E}$. Let $C = [c^{i,j}]$ be the *weighted adjacency matrix*, or simply *connectivity matrix*. Denote by $\mathcal{G}_C = (\mathcal{N}_C, \mathcal{E}_C)$ the weighted directed graph, where $\mathcal{N}_C \equiv \mathcal{N}$ and $\mathcal{E}_C \equiv \mathcal{E}$. We assume that weight $c^{i,j}$ is the cost of data transmission through the edge $(j, i) \in \mathcal{E}_C$. The *weighted in-degree* of $i \in \mathcal{N}_C$ is defined as $\deg^+_i(C) = \sum_{j=1}^n c^{i,j}$, the maximum in-degree among all nodes contained in the graph $\mathcal{G}_C$ as $\deg^+_{\max}(C)$.

The above-mentioned unreliability factor related to limited bandwidth can be associated with the cost of data transmission in the network and characterized by matrix C. As in [Granichin et al., 2020a], we represent cost constraints of agent $i \in \mathcal{N}$ as a predefined upper bound Q: $\deg^+_i(C) \le Q$. Thus, the total bandwidth of the network is adaptively adjusted in terms of the total cost of communication with neighbors of agent i. To satisfy these constraints, we may generate at each time instant t subgraph $\mathcal{G}_{B_t} \subset \mathcal{G}_C$ associated with the weighted connectivity matrix B_t such that $\deg^+_i(B_t) \le Q$. One way of doing this is to use a randomized topology similar to the scheme used in *gossip* algorithms [Boyd et al., 2006].

Next, we consider a data aggregation protocol satisfying the predefined averaged cost constraints: *Definition. [Granichin et al., 2020a]* Random subgraph $\mathcal{G}_{B_t}$ satisfies the averaged cost constraints with level Q if

$$\mathbb{E} \deg^+_{\max}(B_t) \le Q. \tag{28}$$

5.2 Data Aggregation Protocol Based on LVP

During each time interval $[t; t+1]$, the agents perform K communication rounds. We assume that at communication round k agents are able to communicate with their neighbors through the network defined by graph $\mathcal{G}_{B_k} = (\mathcal{N}_{B_k}, \mathcal{E}_{B_k})$. Also, the corresponding connectivity matrix B_k satisfies some averaged cost constraints (28) with level Q.

If set $\mathcal{N}^i_k = \{j \in \mathcal{N}_{B_k} : (j, i) \in \mathcal{E}_{B_k}\}$ is not empty, the agent receives measurements transmitted by its neighbors through noisy communication channels

$$\bar{\mathbf{y}}^{i,j}_k = \bar{\mathbf{y}}^j_{k-d^{i,j}_k} + \mathbf{w}^{i,j}_k,\ j \in \mathcal{N}^i_k, \tag{29}$$

where $\mathbf{w}^{i,j}_k$ is communication noise, $0 \le d^{i,j}_k \le \bar{d}$ are integer-valued delays, and $\bar{d}$ is a maximum possible delay. If $j \notin \mathcal{N}^i_k$ we set $\bar{\mathbf{y}}^{i,j}_k = 0$.

After all, we apply local voting protocol and derive data aggregation protocol:

$$\bar{\mathbf{y}}^i_0 = y^i,$$
$$\bar{\mathbf{y}}^i_{k+1} = \bar{\mathbf{y}}^i_k + \gamma \sum_{j \in \mathcal{N}^i_k} b^{i,j}_k (\bar{\mathbf{y}}^i_k - \bar{\mathbf{y}}^{i,j}_k),\ k = 1, \dots, K, \tag{30}$$

where $\gamma > 0$ is a consensus step-size. Here we assume that the protocol requires the amount of time that is much smaller then each time interval $[t; t+1]$. In [Amelina et al., 2015], it is showed that $\bar{\mathbf{y}}^i_k$ converges to $\bar{\mathbf{y}}$ under noised and delayed measurements if graph $\mathcal{G}_{B_{av}}$ is strongly connected, where $B_{av} = [b^{i,j}_{av}]$, $\mathbb{E} b^{i,j}_t = b^{i,j}_{av}$. Therefore $\bar{\mathbf{y}}^i_K \approx \bar{\mathbf{y}}$.

6 A New Method of Adaptive Mesoscale Control

We summarize the proposed cluster control theory in this Section by manifesting a complete control algorithm for multiagent group steering toward a pre-defined goal. Each agent is assumed to perform its own compressed measurements using the compressed sensing methodology regarding its neighboring agents, forming a local sub-system. The whole network can therefore efficiently estimate its high-dimensional state in a distributed manner with the local voting protocol and perform intelligent decisions due to emergent complex behavior led by numerous local communications and corresponding dynamics. Correspondingly, we combine these steps in the Algorithm 1, which describes the control pipeline.

Algorithm 1 Adaptive mesoscale control method

Require: measurement matrix $A \in \mathbb{R}^{md \times Nd}$, consensus step-size $\gamma > 0$, cost constraint Q

for each $i \in \mathcal{N}$ during time interval $[t; t+1]$ **do**

obtain private measurement

$$y^i = A^i x^i$$

$k \leftarrow 1$

set initial value $\bar{\mathbf{y}}_0^i = y^i$

while $k \neq K$ **do**

randomly define set of neighbors $\mathcal{N}_k^i$ to satisfy cost constraint Q

collect measurements of neighbors $\bar{\mathbf{y}}_k^{i,j}$, $j \in \mathcal{N}_k^i$, taking form (29)

apply local voting protocol

$$\bar{\mathbf{y}}_{k+1}^i = \bar{\mathbf{y}}_k^i + \gamma \sum_{j \in \mathcal{N}_k^i} b_k^{i,j} (\bar{\mathbf{y}}_k^i - \bar{\mathbf{y}}_k^{i,j})$$

$k \leftarrow k + 1$

end while

set $\bar{\mathbf{y}} = \bar{\mathbf{y}}_K^i$

reconstruct $\widehat{\mathbf{x}}_t$ by $\bar{\mathbf{y}}$ using pretrained model: $\widehat{\mathbf{x}} = \mathcal{F}(\bar{\mathbf{y}})$, where $\widehat{\mathbf{x}}_t$ is an estimate of $\mathbf{x}_t$

fit Gaussian mixture model Γ with s components to $\widehat{\mathbf{x}}_t$

compute the mean and covariance matrix for each cluster using model Γ

identify agent's cluster membership with the closest center using nearest neighbour algorithm

apply control input $\bar{u}$ to the identified cluster

end for

We refer to the system control scheme illustrated in Figure 1 and modify it according to the proposed algorithm. The new scheme is shown in Figure 4.

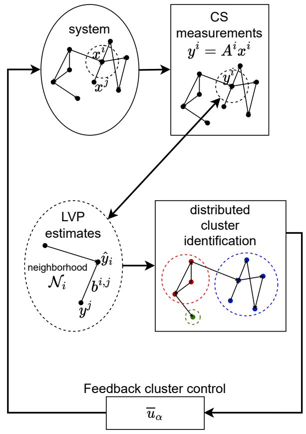

Figure 4. Observation feedback cluster control for a multiagent complex system with clusterization according to the new proposed Algorithm 1.

7 A Universal Plug-in Module for Evaluating the Aggregated Characteristics of the System

In order to perform distributed data aggregation and subsequent adaptive meso-scale control, we develop a universal embedded hardware and software module. This module extends the functionality of mobile robots of various types, sizes and types of portable equipment. Such a complex is a high-performance microcomputer based on a Russian-made TRIK board with support for various sensors and various autopilots. On such a microcomputer, the real-time operating system Embox OS is installed, developed at the Department of Software Engineering of St. Petersburg State University. The complex is designed to quickly increase the functionality of a mobile robot without changing the "native" software for the autopilot and maintaining the reliability of its operation. The main functions of the complex are organization of decentralized group interaction, work with additional devices (camera, thermal imager, rangefinder, etc.), autonomous adjustment of robot control parameters in conditions of uncertainty. It is also possible to expand the functionality of the complex, for example, by introducing optimization algorithms, pattern recognition algorithms, etc. When solving a large class of tasks to optimize and manage the aggregated characteristics of the entire group of robots (load balancing, goal allocation, territory coverage, etc.) to ensure decentralized interaction, we propose to abandon the traditional routing of data in the network and use a locally generated aggregated assessment of the state of the entire system as a whole on each robot, calculated from its own data and data from the nearest neighbors. To synchronize the evaluation between all robots, we use the local voting protocol and multi-agent system technologies.

The final goal is to develop a universal module embedded in the robot that will be easily integrated into the control system of a mobile robot, interact with autopilots of various manufacturers, increase functionality (work in a decentralized group, support for various sensors, work with additional devices) and robot performance (implementation of algorithms for optimizing drone flight, pattern recognition, support for autonomy, etc.). Due to the fact that the module will naturally integrate into the robot's architecture and work with the autopilot in its usual mode, namely, sending new flight coordinates and the type of task to the autopilot, the autopilot will not be loaded with new software and functionality, and therefore, the level of reliability of the autopilot will remain. Such a system will allow you to freely embed various equipment and new algorithms for working with actuators and sensors without changing the autopilot software. The main technical parameters of the proposed solution are described below. The proposed hardware and software complex should support various autopilots and control microcomputers existing on the market.

Thus, the developed complex should have the following technical parameters for hardware and software:

- work with autopilots via FTDI, USB or COM directly;
- work with protocols of various autopilots. One of the most common channels for communication with autopilot at the moment is MAVLink;
- support for USB 3.0+ devices (at least 2 pcs.);
- work with various means of wireless communication: Bluetooth, Xbee (Zigbee), Wi-Fi, GSM (GPRS);
- work with various types of additional equipment – camera, thermal imager, additional telemetry sensors, etc.;
- work with various types of batteries (components of the complex, which include means of protection, preservation of the battery, as well as a built-in integrated stabilizer);
- work with actuators by PWM signal;
- work in group interaction mode with support for new decentralized data exchange protocols (without a single decision-making center).

8 Simulation

To evaluate the effectiveness of control algorithms, a simulation was implemented and a comparison of the algorithms was carried out.

8.1 Simulation details

To make a comparison, a world was developed in which agents exist, the laws of this world, and a task was set for agents that they must complete.

- The world consists of identical regular hexagons that create an endless plane.
- In the world, time passes discretely.
- On one cell of hexagons there can be only one agent at a time.
- The task of the agents is to reach a certain cell on the plane – target.
- Each agent can communicate with other agents by sending messages.
- Agents can move to adjacent cells to their location.
- The distance between two cells is entered as the minimum number of steps an agent must take to move from one cell to another.
- For each time step, an agent can exchange messages with other agents and make moves.
- Penalty steps are applied to agents for not fulfilling the rule of having only one agent on one cell.
- Agents cannot move during penalty steps.

8.2 Controls for comparison

For an objective assessment of the proposed control algorithm, a comparison is made with micro- and macro-control algorithms. The results of comparisons of such algorithms make it possible to carry out analogies for a wide range of algorithms for control a group of agents.

- With micro-control, each agent independently calculates the path to the target, not taking into account other agents. This can lead to collisions of agents, when several agents want to be on the same cell, or raids of agents on another agent - one agent is already on the cell that the neighbors want to get into. In the event of a collision of agents, each of them receives ten penalty steps and remain in place, in the event of a collision, the agents who wanted to hit them receive penalty steps. To avoid repetition of collisions, agents make a random movement after the end of penalty steps.
- With macro-control, agents calculate the center of mass of the entire group, for this the local voting protocol is used. Then calculate the trajectory from the center to the target. The movement of each agent is defined as follows: a trajectory for the movement of the center is taken and applied to the location of the agent.
- With meso-control, agents determine clusters and their membership in them, while one agent can be in only one cluster at a time. Then, the optimal positions of the end points of the clusters and the optimal paths to them from the initial position are determined. After that, for each cluster, the macro-control algorithm is executed, while each cluster may have its own goal, which differs from the target of the entire group.

The use of macro- and meso-control algorithms makes it possible to avoid collisions of agents.

8.3 Research Questions

- **RQ1** Which control will give the best dynamics of task execution?
- **RQ2** What accuracy will each of the controls give?
- **RQ3** The end result of which control will be better?

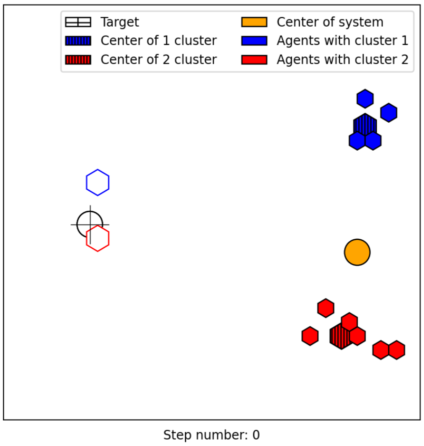

Figure 7. Meso-control

Figure 8. Macro-control

8.4 Metrics

Let's introduce metrics that will allow us to evaluate the efficiency and success of the task execution by a group of agents.

- The group diameter is entered as the maximum distance between a pair of agents.
- The accuracy of the group is introduced as the sum of the accuracy values of all agents. The agent's accuracy, in turn, is the distance from the agent's location to the target.

8.5 Results

In the experiments, the number of agents was 10 on a 24 by 24 plane for each control. According to their results, graphs are illustrated and animated pictures are built.

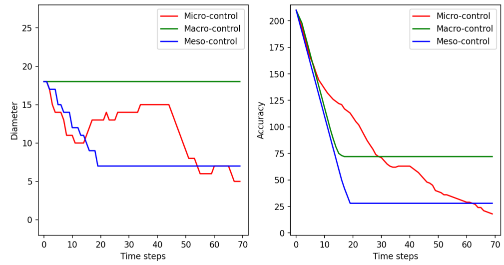

Figure 5. Diameter and accuracy measurements for 10 agents

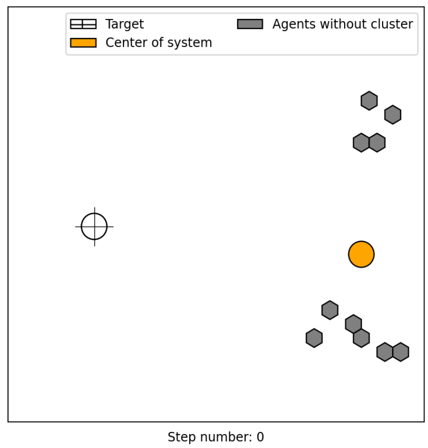

Figure 6. Micro-control

Figures 5-8 demonstrate that meso-control gives the best dynamics of achieving results in terms of the combination of two metrics.

8.5.1 RQ1 As can be seen from the graphs, the best dynamics in the execution of the task is provided by macro-control and meso-control algorithms. In this case, the end result of macro-control is worse.

8.5.2 RQ2 According to the results, micro-control and meso-control algorithms have the best accuracy. However, meso-control algorithms achieve the best accuracy much faster.

8.5.3 RQ3 The micro-control algorithm slightly outperforms the meso-control algorithm. Also, the macro-control algorithm significantly loses to both. However, the micro-control strategy reaches consensus in an unstable way and requires more time.

8.6 The discussion of the results

For a better understanding of the experimental results, let us scale them.

First, imagine that there are not 10 agents, but several thousand. In this case, the micro-control algorithm will further stretch the entire group, and also exponentially increase the number of collisions. All this will lead to a serious deterioration in the achievement of the target.

In the case of macro- and meso-control, the result will remain the same, except that the number of casters will increase. But at the same time, the meso-control algorithm will give.

Therefore, we can conclude that the meso-control algorithm will give the best result with an increase in the number of agents.

9 Conclusions

In this paper, we have described a new approach to complex system control based on efficient cluster (mesoscopic) control paradigm. A novel framework for complex multiagent system analysis aims to utilize system sparsity for cluster control synthesis. Also, with the help of the compressed sensing methodology, we have developed a new distributed cluster control algorithm. By receiving sparse system features in a latent space, we have shown how these features can be exploited to synthesize efficient control action. Finally, we have proposed a universal hardware module for evaluation of aggregated cluster characteristics. Our simulations show that the new adaptive meso-scale control method outperforms other strategies chosen for the comparison.

10 Acknowledgments

Theoretical results in Sections 1–7 were supported by the IPME RAS by Russian Science Foundation (project no. 21-19-00516). Experimental results in Section 8 was supported by the St. Petersburg State University (project ID 93024916).

References

Acebron, J., Bonilla, L., Pérez-Vicente, C., Farran, F., and Spigler, R. (2005). The Kuramoto model: A simple paradigm for synchronization phenomena. *Reviews of Modern Physics*, **77**.

Amelina, N. (2013). Local voting protocol for decentralized load balancing of network with switched topology and noise in measurements. *Vestnik St. Petersburg University: Mathematics*, **46**.

Amelina, N., Fradkov, A., Jiang, Y., and Vergados, D. J. (2015). Approximate consensus in stochastic networks with application to load balancing. *IEEE Transactions on Information Theory*, **61** (4), pp. 1739–1752.

Arnold, V. and Silverman, R. (1987). *Ordinary Differential Equations*. Mit Press.

Baraniuk, R., Davenport, M., DeVore, R., and Wakin, M. (2008). A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, **28**, pp. 253–263.

Benedetto, D., Caglioti, E., and Montemagno, U. (2014). On the complete phase synchronization for the Kuramoto model in the mean-field limit. *Communications in Mathematical Sciences*, **13**.

Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006). Randomized gossip algorithms. *IEEE/ACM Transactions on Networking (TON)*, **14** (SI), pp. 2508–2530.

Candes, E. and Romberg, J. (2005). l1-magic: Recovery of sparse signals via convex programming. pp. 1–19.

Candès, E., Romberg, J., and Tao, T. (2006). Robust uncertainty principles : Exact signal frequency information. *Information Theory, IEEE Transactions on*, **52**, pp. 489–509.

Candes, E. J., Romberg, J., and Tao, T. (2006). Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, **52** (2), pp. 489–509.

Chopra, N. and Spong, M. (2006). On synchronization of Kuramoto oscillators. vol. 2005, 01, pp. 3916–3922.

Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, **52** (4), pp. 1289–1306.

Dörfler, F., Jovanović, M. R., Chertkov, M., and Bullo, F. (2014). Sparsity-promoting optimal wide-area control of power networks. *IEEE Transactions on Power Systems*, **29** (5), pp. 2281–2291.

Doyle, J. (1996). Robust and optimal control. In *Proceedings of 35th IEEE Conference on Decision and Control*, vol. 2, pp. 1595–1598.

Fele, F., Maestre, J. M., and Camacho, E. F. (2017). Coalitional control: Cooperative game theory and control. *IEEE Control Systems Magazine*, **37** (1), pp. 53–69.

Fradkov, A. (2007). *Cybernetical Physics*. Springer.

Furieri, L., Zheng, Y., Papachristodoulou, A., and Kamgarpour, M. (2020). Sparsity invariance for convex design of distributed controllers. *IEEE Transactions on Control of Network Systems*, **7** (4), pp. 1836–1847.

Galbraith, G. and Vinter, R. (2003). Lipschitz continuity of optimal controls for state constrained problems. *SIAM J. Control and Optimization*, **42**, pp. 1727–1744.

Gazi, V. and Fidan, B. (2007). Coordination and control of multi-agent dynamic systems: Models and approaches. vol. 4433, 07, pp. 71–102.

Giordani, P., Ferraro, M., and Martella, F. (2020). *Hierarchical Clustering*, pp. 9–73.

Granichin, O., Erofeeva, V., Ivanskiy, Y., and Jiang, Y. (2020a). Simultaneous perturbation stochastic approximation-based consensus for tracking under unknown-but-bounded disturbances. *IEEE Transactions on Automatic Control*, **66** (8), pp. 3710–3717.

Granichin, O. and Fomin, V. (1986). Dynamic programming method in minimax control problem. *Vestnik Leningradskogo Universiteta, Matematika*, **19** (1), pp. 26–30.

Granichin, O. and Uzhva, D. (2020). Invariance preserving control of clusters recognized in networks of kuramoto oscillators. In *Artificial Intelligence*, vol. 11, Springer International Publishing, pp. 472–486.

Granichin, O. and Uzhva, D. (2021). Cluster control of complex cyber-physical systems. *Cybernetics and Physics*, **10**, pp. 191–200.

Granichin, O. and Uzhva, D. (2022). Compressed cluster sensing in multiagent iot control. In *Proc. of the 2022 IEEE 61st Conference on Decision and Control (CDC), Mexico*, pp. 3580–3585.

Granichin, O., Uzhva, D., and Volkovich, Z. (2020b). Cluster flows and multiagent technology. *Mathematics*, **9**, pp. 22.

Hernandez, S. and Garcia, R. (2014). An observer for switched lipschitz continuous systems. *International Journal of Control*, **87**.

Hu, S., Leung, C., and Leung, H. (2019). Modelling the dynamics of multiagent q-learning in repeated symmetric games: a mean field theoretic approach. In *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc.

Jadbabaie, A., Motee, N., and Barahona, M. (2005). On the stability of the Kuramoto model of coupled nonlinear oscillators. vol. 5, pp. 4296–4301.

Jerray, J., Fribourg, L., and André, E. (2021). An approximation of minimax control using random sampling and symbolic computation. *IFAC-PapersOnLine*, **54**, pp. 265–270.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, **82** (Series D), pp. 35–45.

Karna, A. and Gibert, K. (2022). Automatic identification of the number of clusters in hierarchical clustering. *Neural Computing and Applications*, **34** (1), pp. 119–134.

Kelley, H. (1960). Gradient theory of optimal flight paths. *ARS Journal*, **30**, pp. 947–954.

Le, H. and Mendes, E. (2008). Optimal control of first order linear systems with fixed proportional-integral structure controller. In *2008 10th International Conference on Control, Automation, Robotics and Vision*, pp. 1799–1804.

Li, H., Li, Y., and Zhao, H. V. (2020). Modeling decision process in multi-agent systems: A graphical markov game based approach. In *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 197–204.

Liang, D., Ma, F., and Li, W. (2020). New gradient-weighted adaptive gradient methods with dynamic constraints. *IEEE Access*, **8**, pp. 110929–110942.

Luo, G., Zhang, H., He, H., Li, J., and Wang, F. (2021). Multiagent adversarial collaborative learning via mean-field theory. *IEEE Transactions on Cybernetics*, **51** (10), pp. 4994–5007.

Lyapunov, A. (1892). *The General Problem of the Stability of Motion*. PhD thesis.

Mughnyanti, M., Efendi, S., and Zarlis, M. (2020). Analysis of determining centroid clustering x-means algorithm with davies-bouldin index evaluation. *IOP Conference Series: Materials Science and Engineering*, **725**.

Mukherjee, D. and Ghose, D. (2016). Generalized hierarchical cyclic pursuit. *Automatica*, **71**, pp. 318–323.

Nesterov, Y. and Todd, M. (1998). Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on Optimization*, **8**, pp. 324–364.

Polyak, B. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, **4** (5), pp. 1–17.

Proskurnikov, A. and Granichin, O. (2018). Evolution of clusters in large-scale dynamical networks. *Cybernetics and Physics*, **7** (3), pp. 102–129.

Ravazzi, C., Dabbene, F., Lagoa, C., and Proskurnikov, A. (2021). Learning hidden influences in large-scale dynamical social networks: A data-driven sparsity-based approach, in memory of roberto tempo. *IEEE Control Systems Magazine*, **41** (5), pp. 61–103.

Sadilek, M. and Thurner, S. (2014). Physiologically motivated multiplex Kuramoto model describes phase diagram of cortical activity. *Scientific Reports*, **5**.

Schiffer, J., Dörfler, F., and Fridman, E. (2017). Robustness of distributed averaging control in power systems: Time delays & dynamic communication topology. *Automatica*, **80**, pp. 261–271.

Silva, E. (2014). *DG-ABC: An Integrated Multi-Agent and Cellular Automata Urban Growth Model*, pp. 57–92.

Singh, S. (2021). Study of k-means and enhanced k-means clustering algorithm. *International Journal of Research in Computer Science*, **4**, pp. 103–107.

Strogatz, S. H. (2000). *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Westview Press.

Yang, H., Zhang, K., Zheng, K., and Qian, Y. (2020). Leveraging linear quadratic regulator cost and energy consumption for ultrareliable and low-latency iot control systems. *IEEE Internet of Things Journal*, **7** (9), pp. 8356–8371.

Zhang, X., Zhang, S., Lin, J., Sun, F., Zhu, X., Yang, Y., Tong, X., and Yang, H. (2019). An efficient seismic data acquisition based on compressed sensing architecture with generative adversarial networks. *IEEE Access*, **7**, pp. 105948–105961.