ADAPTIVE EXCHANGE PROTOCOL FOR MULTI-AGENT COMMUNICATION IN AUGMENTED REALITY SYSTEM

Ekaterina Moseiko
Saint Petersburg
State University
Russia
ekaterina.mosseyko@compvisionsys.com

Victoria Eroveeva
Institute of Problems of
Mechanical Engineering RAS
Russia
eva@ipme.ru

Oleg Granichin
Saint Petersburg
State University
Institute of Problems of of
Mechanical Engineering RAS
Russia
oleg_granichin@mail.ru

Article history:
Received 28.11.2023, Accepted 11.12.2023

Abstract

Augmented Reality (AR) is one of the modern ways of providing users with different types of information. In applications based on this technology, the main influence on the subjective assessment of product quality is the speed of interaction between the system and the user and between users of the system. The main purpose of the work series are creation and modification for algorithms to assessing and improving the quality of AR-services. This paper describes and justifies an adaptive communication protocol for multi-agent interaction. We consider a general nonlinear dynamic system with introducing feedback control which is based on measurements under almost arbitrary noise. In practice, this control is realized as a superposition of neural networks with the estimation of the result of mutual additional learning based on the system identification method (M.K.Campi and E.Weyer's LSCR is used). The prototype is built for a system with a distributed server architecture and multi-agent behavior of both clients and servers in the system.

Key words

Adaptive control, Multi-agent communication, Augmented Reality, Simulation

1 Introduction

Human-machine interaction is one of the popular areas for research. Over the decades, numerous console applications have grown into windowed GUIs and then web and mobile applications.

One of the modern ways of providing information is the use of Augmented Reality services (AR). In applications based on this technology, objects created by programmers coexist with objects in real space. In contrast to Virtual Reality services (VR), the main influence on the quality of AR is the accuracy of positioning in space and the adequate speed of user interaction. The positioning of clients agents in space with high accuracy should be 10-100 times higher than the capabilities of modern navigation trackers (GPS or other sensor data accuracy). The "speed of interaction" should provide an acceptable response time when interacting with elements of the expanded world and in the group interactions within the selected AR-space.

A mobile phone or tablet allows the user to perform more complex actions than a conventional windowed GUI.

The number of agents and objects can be placed to the space of augmented reality is important. The difficult of the task that to be decided depends on this number.

Currently particularly promising areas of application of AR are interactive tours, marketing, urban information systems, interactive assistants in huge shopping centers, airports. Separately, it is worth noting such a scope as digital twins of production.

The combination of these factors makes solutions consumers of large computing resources and required high data exchange rates.

But science is not far behind. There are more and more works devoted to research adaptive flow control methods [Proskurnikov and Granichin, 2018] and distributed optimization-based approaches [Wang et al., 2018].

A lot of new research in designing and analyzing cooperative control algorithms for large-scale multiagent systems [Dalai et al., 2007; Xia et al., 2021] may be applied in the AR-technology.

The purpose of the work is a description and justification of nn algorithm for assessing and improving the quality of AR-services in the scope distributed server and multi-agent client environment. The time limit for any task in the system should be predictable.

The benefit of designed solution is achieved by introducing feedback control of a nonlinear dynamic system considering under measurements with almost arbitrary noise. For decentralized load balancing for multi-servers and multi-agent network with variable topology and noise in the measurements Local Voting Protocol modification used [Amelina et al., 2015]. The part about balancing requests is not this the aim of this paper. Main discuss here is about place of neural networks and usage LSCR [Dalai et al., 2007] as a service-level agreement (SLA) metric for our system.

In practice, this control is implemented in the form of a composition of neural networks with an assessment of the result of mutual additional training based on the method for determining the system parameters.

The prototype is built for a model system with a distributed server architecture and multi-agent behavior of both clients and servers in the system. The integration of the provided algorithm into current system of augmented reality services is in a stage of agreement with one of customers.

2 Background

One of the first devices that can be called an augmented reality device is mentioned in the children's novel "The Wizard of Oz" - glasses through which you can see not only the surrounding space, but also objects that are not there.

The term "augmented reality" was officially used by Tom Cadell, a Boeing Researcher, in 1990.

In 2009 ARToolKit has made AR technology available to all Internet browsers.

Since 2016 and up to now, AR-theme is at the forefront of science and is actively developing. Also it becomes not only gaming, but also socially significant. For example, in 2020, smart glasses for almost invisible people were presented. In 2022, such glasses began to be sold on behalf of Envision.

The second side of the work presented is densely related to math in control theory.

It has been known since the 1960s that many practical problems can be reformulated as a robust convex optimization problem [Rastrigin, 1963] where a convex function has to be optimized under restrictions that are also given by convex functions.

An identification approach is often used for adaptive control [Vakhitov et al., 2010]. This approach constructs estimates of the possible values of unknown parameters x based on sequences of observations, and these estimates are then used in a parameterized feedback loop that, if properly chosen, usually provides a closed-loop system quality that satisfies the user.

This work extends our experience in the practical implementation of the ideas of adaptive control and randomized algorithms under uncertainty, which is the focus of a sequence of papers by O. Granichin and K. Amelin [Amelin and Granichin, 2016] and O. Granichin and D. Uzhva [Uzhva and Granichin, 2021].

3 Augmented reality context

Let us introduce the following concepts:

1. *Client* — a physical device with a camera and sensors that allow you to position yourself in space and time (in the augmented reality). Must have access to the interaction network.
2. *Server* — a physical device with sufficient capacity to perform complex calculations, receive and process requests. Must have access to the interaction network.
3. *Object* — an element of augmented reality.
4. *Reconstruction or cloud of points* — a 3D space model formed on the basis of frames in the process of shooting with a mobile device. It's may be *sparsed* or *full*.
5. *Position* is a place of *Client* or *Object* in the real space coordinated within model space.
6. *Key frame (or key image)* — randomly selected frame with specified characteristics. In relation to the existing set of key frames it is searched for the next ones to stabilize the system operation. The number of key frames is important input parameter.
7. *Localization* — high accuracy positioning of the client or object in a specific place of a given space (both real and virtual). Let call the *localization rate* a metric (real number in the $[0; 1]$) calculated as relation of successful matching real frames within base frames to all matches. This is a metric of observable result.
8. *Request and response* — a message transmitted via http(s) according to the protocol for interaction and data transmission in the network.
9. Let *Service* is a set of queries (requests and replies) and feedback about it (user's subjective opinion and *localization rate*)
10. *Quality of service* is a customer satisfaction, real number from 0 to 1. This is a key metric for assessing customer satisfaction and platform performance.

Consider the process of using an augmented reality service. At some moment of time t user C^i contacts the server with a request to get a response to a localization request or to send a reconstruction or object management task for processing. A separate microservice is responsible for the execution of each type of task, which is not connected to the others in any way except by a message queue.

Next, consider the four most important calculated *inputs* as a common vector x and the two *outputs* that describe the quality of our software solution (the first on the client side, the second on the server side) as a vector! y.

We want to analyze the statistics of requests, so in terms of setting parameters we have exactly one request at each moment (i.e. the process is discrete), based on which we will "guess" the view of the system within its use by an average user.

The inputs for us are: number of key frames, area of view and area of localization, the mean responsible for the trajectory change in time (position refinement). All these parameters are either monotonically non-decreasing (linear with some error) or constant with some error.

Two parameters are outputs: the probability that the client will be satisfied with the quality (the quality value is individual, but is known in advance) and the probability that the servers will be able to execute the next task of any type (depends on the current load, but is known at the time of receiving the next task, because computing power is fixed).

3.1 Client data and functions

Clients have the following set of data with unknown accuracy:

current time within client timezone;
location (global geoposition);
location in space (local pose of the device at the location point or change of pose relative to the starting one); for some clients:
pictures of the space obtained by the cameras of this device;
text / audio / video / 3D models of augmented reality objects for placement.

Clients perform the following tasks:

take images of real space from camera according to a predetermined algorithm for scene 3D-model creation;
assessment of the quality of the reconstructed scene and the correctness of its placement in real space;
placement of objects on the reconstructed scene;
localization in the reconstructed scene;
assessment of the quality of localization;
receiving objects on active scene according to the access level;
assessment of the quality of placement of an object in real space;
search for the nearest server from the list of available;
making a decision on the amount of transmitted information depending on the network speed and response time from the selected server;
switch to another server or self-organize in case of loss of connection with the server.

Client Restrictions:

Scenes received from real space can be both public and private.
Groups of objects with different access levels can be placed on the same scene.
The client should only see those scenes, objects and other clients that his access rights allow.

3.2 Server data and functions

Server have the following set of data:

Impersonal key frames along with GPS data received from clients;
Reconstructions (3D-cloud of points);
AR-objects.
Information about rights and limits.

Servers perform the following tasks:

Accept http requests from clients and send responses.
Carry out complex calculations that are not available to customers.
Carry out the construction of reconstructions.
Carry out localization.
Store, process data.
Provide differentiation of access rights and limits.
Ensure the security of data placement.
Redistribute tasks between servers to ensure optimal response time to requests
Sync data and share it with according to query statistics.

4 Math model and problem setting

As we have already described, we have some set of values of output parameters of the observed system. There are also internal services of the system, each of which operates according to its own algorithm and has a criterion for evaluating a particular algorithm. We are interested in managing the algorithms of services to achieve consensus, since it is the joint work of services that affects the final result. Let us consider the parameters of the services as a computational vector, or more precisely, as an associated optimal predictor. The predictor can be estimated by the main metric, namely, the vector of system identification parameters, which is determined by the LSCR algorithm, the essence of which is described in [Dalai et al., 2007], and one of the ideas of its application in [Moseiko and Granichin, 2023].

The peculiarity of using this particular identification algorithm is that it provides not a specific model of the system, but a class of models. According to the algorithm, we can obtain such values of θ that guarantee that the observation results fall into the confidence interval with any accuracy specified in advance by the user of the system.

Thus in this work we use LSCR as SLA metric for estimate current server solution and try to predict the result for new version of system before integration and real usage by clients.

Then describe the control process over our real augmented reality system with the strong math ideas.

The mathematical model of this problem is a dynamic system with a control of the form

$$\dot{x} = f(x, u, w, \theta),$$

where x is the input data vector of the system, u is the control vector, w is the disturbance, and θ is the parameter vector we want to estimate.

Because the input data for our system is a sequence of requests, then we can consider our system as a discrete system with noise and control

$$x_{t+1} = f(x_t, u_t, w_{t+1}, \theta),$$

where x_t is a vector of input values transmitted by the user of the system at time t, w_{t+1} is a new disturbance.

We already have a working system in which we can make observations. Let g be some model that describes how to obtain a vector of values y_t from the observation vector x_t. In other words, we have a model of observations with noise

$$y_t = g(x_t, u_t, v_t, \theta),$$

where y_t is the observed result and v_t is unmeasured noise that describes all sources other than u_t that cause changes in y_t.

The mathematical model of observations for each connection is a nonlinear dynamical system. Consider a nonlinear system S which maps a non-measured noise process w_t into a measured signal y_t. Say that S is belongs to a parameterized system class $\{S\}$. Lets v_t is an independent sequence of random variables whose distribution is symmetric around zero. As in [Chou et al., 2022] we don't need in the other assumptions on v_t. The distribution of v_t can as well be time-varying. We aim at finding a region confidence for the parameter vector θ by observing the output y_t. In other words we have a system of the form $y_t = u_t\theta + v_t$, where u_t are control inputs, θ is a parameter that needs to be estimated under conditions of uncertainty, v_t is a random noise that occurs during measurements. According to the theory described in papers [Dalai et al., 2007] and [Amelin and Granichin, 2016], in this problem setting it is possible to apply the LSCR method. By this theory is a good idea to find the Algorithm which always produces a region that contains θ with a probability chosen by the user. In a situation of uncertainty, noise is distributed in such a way as to guarantee that the result falls into a predetermined confidence interval.

The LSCR-algorithm[Dalai et al., 2007] can be applied to this control system to construct a confidence interval for the parameter θ, where θ is some of control parameters u_t.

For our application, we can use a combination of our input data u_t, the computed online metrics y_t, the total vector $\begin{pmatrix} v \\ w \end{pmatrix}$ for known step as vector $\widetilde{w}$, and θ is the result of the micro-services logic, i.e., the quality of the finished output. Thus, it turns out that we have a metric to compare the quality of the results.

5 Data system and predictors

In the common case the data system is given in the matrix-as-vector form.

$$\begin{aligned} & y_t = A_t^T \theta^0 + \widetilde{w_t}, \\ & \text{where} \\ & A_t = [-y_{t-1}, \ldots, -y_{t_{n_a}}, u_{t-1}, \ldots, u_{t_{n_b}}], \\ & \theta^0 = [a_1^0, \ldots, a_{n_a}^0, b_1^0, \ldots, b_{n_b}^0] \end{aligned} \quad (1)$$

In our terms the user can choose the signal u_t and choice of u doesn't affect unmeasured and not known values of w_t for any time instant t. So we know constant values of n_a and n_b for our model order.

We will use

$$\widetilde{y_t}(\theta) = A_t^T \cdot \theta$$

as a predictor, where

$$\theta = \theta^0$$

. Then the prediction error is given by

$$\varepsilon_t(\theta) = y_t(\theta) - \widetilde{y_t}(\theta).$$

We realize that some of the parameters called a or b are known to us (depending on whether we are working with the client side or the server side).

The goal is to construct from the observations a set of confidence intervals Θ_i that contain the unknown parameter $\theta_i = b_i$ with a given probability.

6 Non-linear LSCR algorithm

Let t is a "time"-sequence and N its length.
Describes algorithm:

1: Compute the prediction error $\varepsilon_t(\theta)$;
2: Compute the new vector

$$F[t](\theta) = [-y_{t-1}, \ldots, -y_{t_{n_a}}, u_{t-1}, \ldots, u_{t_{n_b}}] \cdot \varepsilon_t(\theta)$$

3: Select an integer M and construct binary stochastic string of lenght as follows: every element of the remainings strings takes one of true/false value with the probably 0.5 and all elements are independent both in pairs and in any aggregate. Name the elements of resulting strings $h[i][t]$, where $i = 1, \ldots M, t = 1, ldotsN$.
Compute

$$G[i](\theta) = \sum_{t=1}^{N} h[i][t] \cdot F[t](\theta).$$

4: Let's plot the graphs of dependence of functions $G[i](b_i)$. Since it is very unlikely that all values of this function have the same sign, let us exclude all regions in which the values have all or except one of the signs. It follows from Campy's theorem that the remaining interval is the region of given validity for $b[i]$.

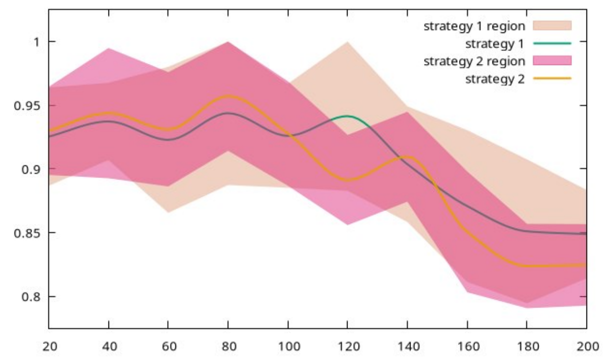

Figure 1. Counterexample: one parameter is not in confidence region of system (intersection) but in the local confidence region

7 LSCR for tyme-varying system

This extension is described in [Layton et al., 2009]

We assume that the variation in θ_t^0 is bounded by

$$|\theta_{t+1}^0 - \theta_t^0| \leq K,$$

where $K = [\triangle a_1^0, \ldots, \triangle a_{n_a}^0, \triangle b_1^0, \ldots, \triangle b_{n_b}^0]$.

In other words, we work with changes to values instead of the values themselves. The order of the model depends on the number of parameters, the fact that we will look at the system linear in each parameter otherwise it will not cease to be linear.

We have obtained a procedure for identifying the parameters of the dynamic system equation described by a moving average autoregressive model. If we can redefine the parameters so that they are in a mutually unambiguous correspondence, then the reparameterization allows us to use the LSCR procedure reasonably and when in the feedback channel with varyiting coefficients.

8 Main result

Let all the conditions for the theorem from [Dalai et al., 2007] hold. Then this theorem is true. According to the theorem in [Layton et al., 2009] and the corollary from it, for each parameter of the system it is clear whether it falls within the confidence interval with a given probability, and or whether the algorithm will work incorrectly (going beyond the boundaries of the confidence interval, convergence is too slow, etc.).

Let us describe the system adjustment algorithm:

1 For each parameter, we will use an algorithm with a fixed, pre-known probability (it is common to the system, regardless of which parameter we are estimating).

2 If all confidence intervals turn out to be narrower than the target one and around the given probability, adjustment is applied.

3 If some confidence interval worsened any one indicator of the system by no more than $\frac{1}{N}$ (25% for 4 parameters, depending on the number of parameters), then we will apply the change at this step and in the next iteration we will discard the parameter that made this change. If the hit is still preserved, then we will recalculate the entire system and work with it.

4 If more than one parameter made a negative contribution, then such a change is not applied.

Theorem. Let's call a strategy a set of parameters and the boundaries of confidence intervals for them. Then, if the theorem from [Layton et al., 2009] was fulfilled for each parameter separately, then after applying the algorithm described above, it will also be fulfilled for all parameters in the aggregate.

The converse theorem is not true, see counterexample (see Figure 1).

Prof: By consequence of Campi's theorem, for each parameter θ_i^0 there is a maximal number of trials M_i such that the confidence interval for each parameter separately contains the theta parameter with a given probability p (not less than $1 - \frac{1}{N}$ and common for all parameters). Let us consider the set of all subsets of the set of selected parameters. The strategy from the theorem corresponds to one of these subsets. Because for each parameter we observe not decreasing (as the amount of data increases, the system stabilizes), then for each parameter we can calculate the mathematical expectation as

$$E|\triangle_{t-1}(b_i)| = (b_i^0 - b)E|\triangle_{t-1}^2| + E|\triangle_{t-1}\widetilde{w_t}| = b_i^0 - b_i.$$

For the "full" system we can decrease model order by the theorem as for system with the control action with test signal [Amelin and Granichin, 2016]. In the approach described in the work, the confidence set (intersection) is constructed by excluding from the set of parameters those for which empirical correlations too often take only positive or only negative values.

9 Simulations

To demonstrate the solution, the qemu-image based on Linux Debian 12 OS and free Android mobile applications using AR-core functions were prepared. Any image has all parts of solution.

Image clones are installed on 9 physical servers geographically located at different points. The images are intended to be deployed on a host system or as a virtual machine hard drive.

Mobile phones and modern computers with a browser are used as client devices. Clients are applications for various purposes that demonstrate the execution of typical AR tasks, or, alternatively, shell-scripts that execute specified sequences of http requests.

The changing of localization service quality by iterations of system modification using LSCR θ-prediction is on the figure below.

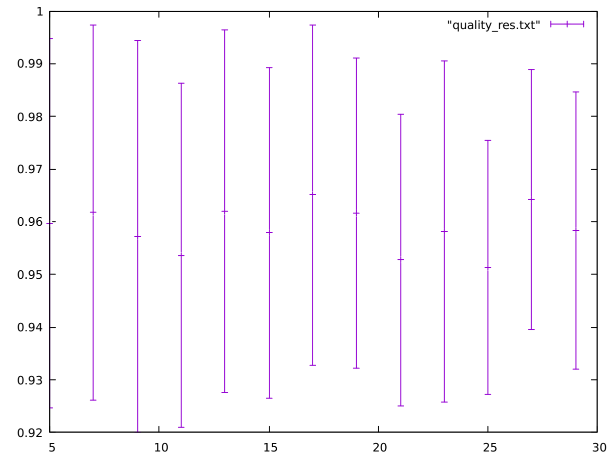

Figure 2. Ninety five percent confidence region of client satisfaction increased by number of keyframes

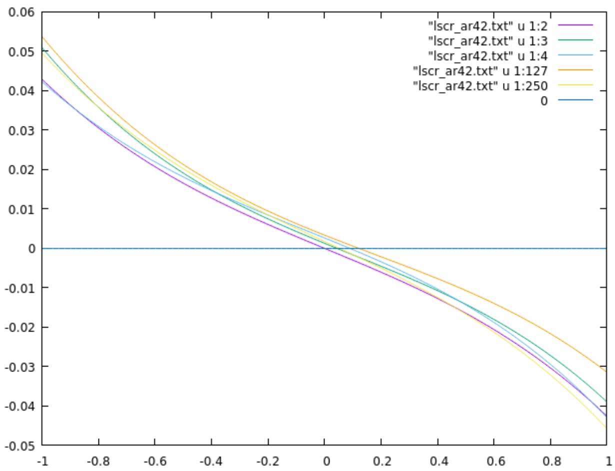

Figure 3. Some of the model g-functions obtained in the real system

The Figure 2 shows that the quality specified by the user always falls within the confidence interval, and the width of the confidence interval narrows with time. This allows you to select input data more qualitatively and reduce amount of data for processing. By experiments the length of confidence interval is reduced after 40 key frames that corresponds second or third minutes of tracking. It proves that quality of localization service is getting better.

This Figure 3 shows that we can find something close to an exact solution, but we only need a good (not exact) solution for it to work in practice. Why always or sometimes, we will try to analyze in our next papers.

10 Application example

One of the application for described method is comparision of the stability and performance of a distributed server solution in comparison with a centralized one, existing public server is used.

For testing, the most popular test scenarios for using AR were selected, namely:

1. Simultaneous creation of scenes (reconstructions) by several users.
2. Simultaneous placement of objects a) in different scenes; b) in the same scene.
3. Updating the scene with new data in the process of localizing independent users in this scene.
4. Localization of a group of clients in the scene in different situations: a) clients are not connected with each other; b) clients see each other within the scene.
5. Simultaneous solution of problems from groups 1-4 by random within restrictions described upper.

Other example. Let's consider a situation where the number of provider resources is known, but strictly limited. The main development goals are to improve the quality of service for existing users and increase the number and geography of new clients. Sooner or later, there comes a point in the system when resources of at least one type become insufficient. The method will help you choose the most suitable strategy for the provider.

Another example is communication between agents in a multi-agent environment. The algorithm described above allows you to compensate for insufficiently good, but not very bad, values of parameters of arbitrary origin.

11 Hardware platform

The developed system was also integrated into a decentralized hardware and software platform for deployment and support of the network operation, based on a universal embedded module for evaluating aggregate system characteristics and providing decentralized communication within the network. The underlying methodology of the multi-agent interactions through aggregate system characteristics is described in [Granichin et al., 2022; Erofeeva et al., 2023]. The network protocol also relies on the approach given in [Amelin and Ershov, 2021]. We have tested the platform in different conditions: to work with autopilots via FTDI, USB or COM directly; to work with protocols of different autopilots; to work with different wireless communication modules: Bluetooth, Xbee (Zigbee), Wi-Fi, GSM (GPRS); to work with different types of additional equipment such as camera, thermal imager, additional telemetry sensors, etc.; work with different types of batteries; work with actuators by PWM signal; work in group interaction mode with support for new protocols of decentralized data exchange (without a single decision-making center).

12 Conclusion

Possible approaches to handle control of multiserver system with multiagent emerging clusterization are discussed.

Developed prototype of multiserver solution with adaptive exchange protocol for multiagent coordination.

According to the test results the use of a multi-server architecture with a local voting protocol shows greater stability in solving localization tasks and less waiting time in reconstruction tasks compared to a centralized server solution.

The combination of these advances makes it possible to use the distributed server solution for a lot of AR-concept problem classes worldwide.

Future work will include the integration this prototype as experimental feature to the current solution, find and study the bottlenecks by real usage statistic, optimization works of the proposed exchange. Also this research will actual in the case where the number and geography of operate queries are huge or too slow for one computation server.

Acknowledgements

This work was supported by Russian Science Foundation (project 21-19-00516, IPME RAS).

References

Amelin, K. and Ershov, V. (2021). Data transfer in a decentralized network of robots using a local voting protocol. *Cybernetics and Physics*, **10** (4), pp. 219–223.

Amelin, K. and Granichin, O. (2016). Randomized control strategies under arbitrary external noise. *IEEE Transactions on Automatic Control*, **16** (5), pp. 1328–1333.

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.

Chou, Y.-Y., Wu, P.-F., Huang, C.-Y., Chang, S.-H., Huang, H.-S., Lin, W.-M., and Lin, M.-L. (2022). Effect of digital learning using augmented reality with multidimensional concept map in elementary science course. *The Asia-Pacific Education Researcher*, **31** (4), pp. 383–393.

Dalai, M., Weyer, E., and Campi, M. C. (2007). Parameter identification for nonlinear systems: Guaranteed confidence regions through lscr. *Automatica*, **43** (8), pp. 1418–1425.

Erofeeva, V., Ershov, V., Granichin, O., Pankov, V., Uzhva, D., and Granichina, O. (2023). Adaptive distributed cluster flow control for a group of autonomous robots. *IFAC-PapersOnLine*, **56** (2), pp. 8690–8695.

Granichin, O., Uzhva, D., Amelin, K., Leonova, A., Pankov, V., Erofeeva, V., and Ershov, V. (2022). A new method of adaptive mesoscale control in complex multiagent networked dynamical systems. *Cybernetics and Physics*, **11** (4), pp. 175–189.

Layton, K., J., Weyer, E., and Campy M., C. (2009). Online algorithms for the construction of guaranteed confidence sets for the parameters of time-varying systems. *in 15th IFAC Symposium on System Identification*, **42(10)**, pp. 426–431.

Moseiko, E. and Granichin, O. (2023). Application the general method for system identification for the best quality of augmented reality providing.

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

Rastrigin, L. (1963). The convergence of the random search method in the external control of many-parameter system. *Automation and Remote Control*, (24), pp. 1337–1342.

Uzhva, D. and Granichin, O. (2021). Cluster control of complex cyber-physical systems. *Cybernetics and Physics*, **10** (3), pp. 191–200.

Vakhitov, A., Vlasov, V., and Granichin, O. (2010). Adaptive control of siso plant with time-varying coefficients based on random test perturbation. 08, pp. 4004 – 4009.

Wang, D., Wang, Z., Chen, M., and Wang, W. (2018). Distributed optimization for multi-agent systems with constraints set and communication time-delay over a directed graph. *Information Sciences*, **438**, pp. 1–14.

Xia, Z., Du, J., Wang, J., Jiang, C., Ren, Y., Li, G., and Han, Z. (2021). Multi-agent reinforcement learning aided intelligent uav swarm for target tracking. *IEEE Transactions on Vehicular Technology*, **71** (1), pp. 931–945.