КОМП`ЮТЕРНІ СИСТЕМИ ТА КОМПОНЕНТИ

UDC 004.85

Model-based learning of coordinators of the decentralized multi-zone objects control systems

Volodymyr M. Dubovoi

Doctor of Science (Engenearing), Professor,
Professor of Computer Control Systems Department
Vinnytsia National Technical University, Vinnytsia
ORCID 0000-0003-0440-3643

Abstract. Decentralized control systems are gaining more and more expansion, which is due to the increase in the availability and power of microcontrollers. Decentralized control of multi-zone objects is associated with the need to coordinate the local control systems of zones state. Learning systems are preferred for implementation of the coordination methods, as they are able for flexibly adjust to the specifics of control of each zone. However, the training of coordinators is complicated task by the absence at the stage of a system creating of marked datasets for controlled multi-zonal objects. This article considers the creation of a dataset based on a simulation of a decentralized system and four scenarios for training neural coordinators. A model for simulation of a decentralized system was been created on the Scilab/Xcos platform using a pre-built library of blocks for simulating decentralized systems. The scenarios differ depending on the structure of the neural coordinators: a segmented network according to the structure of the coordinator simulation model or an integrated one, as well as on the training strategy: train all the coordinators of the decentralized system in parallel or only one coordinator and then clone the results. Experimental studies of the proposed method of training neural network coordinators, implemented on Python TensorFlow, were conducted. The study showed greater effectiveness of segmented coordinators parallel training. However, in the course of the study, the last step of the scenarios – fine tuning on a real physical object, was not performed. A preliminary evaluation suggests that after such additional training, the advantages of mono-neural coordinators will become more visible, since such additional training will correct the shortcomings of imitation.

Key words: machine learning, distributed control system, decentralized coordination, model-based learning.

Модельно-орієнтоване навчання координаторів децентралізованої системи управління багатозональним об'єктом

Володимир Михайлович Дубовой

доктор технічних наук, професор,
професор кафедри комп'ютерних систем управління
Вінницький національний технічний університет, Вінниця

Анотація. Децентралізовані системи керування набирають все більшого розповсюдження, що зумовлено збільшенням доступності і потужності мікроконтролерів. Децентралізоване керування багатозональними об'єктами пов'язане з необхідністю координації локальних систем керування станом зон. Для реалізації методів координації перевагу мають системи, що навчаються, оскільки вони здатні гнучко налаштовуватися на особливості керування кожною зоною. Проте навчання координаторів ускладнюється відсутністю на стадії створення системи розмічених датасетів для керованих багатозональних об'єктів. У цій статті розглядається створення датасету на основі імітаційної моделі децентралізованої системи і чотири сценарії навчання нейронних координаторів. Імітаційна модель децентралізованої системи створена на платформі Scilab/Xcos з використанням попередньо створеної бібліотеки блоків для моделювання децентралізованих систем. Сценарії відрізняються залежно від структури нейронних координаторів: сегментована мережа відповідно до структури імітаційної моделі координатора або інтегрована мережа, а також від стратегії навчання: навчати паралельно усі координатори децентралізованої системи або тільки один і результати клонувати. Проведені експериментальні дослідження запропонованого методу навчання нейромережевих координаторів, реалізованих на Pyton TensorFlow. Дослідження показало більшу ефективність паралельного навчання сегментованих координаторів. Проте в ході дослідження не виконувався останній етап сценаріїв – донавчання на реальному фізичному об'єкті. Попередня оцінка дозволяє припустити, що після такого донавчання переваги інтегрованих нейронних координаторів стануть помітнішими, оскільки таке донавчання дозволить виправити недоліки імітації.

Ключові слова: машинне навчання, розподілена система керування, децентралізована координація, модельно-орієнтоване навчання.

DOI: https://doi.org/10.31649/1999-9941-2024-60-2-66-76

Introduction. Decentralized control systems (DCS) are becoming more and more widespread, which is due to the increase in the availability and power of microcontrollers. Decentralized control of multi-zone objects (MZO) is associated with the need to coordinate local systems for control the state of zones. The peculiarity of multi-zone objects is the mutual influence of zones. As a result, their control becomes much more complicated, since the zone control algorithm must take into account the consequences not only for the controlled zone, but also for other zones of the multi-zone object. There are also other tasks within the framework of the problem of decentralized control of MZO: ensuring the stability of a multi-connected system that arises as a result of the interaction of MZO zones and local control systems (LCS); ensuring the reliability and safety of the DCS, which may decrease due to failures of elements, communication means and coordinators, etc.; ensuring the necessary speed of action, which is limited by the speed of the spread of impacts between MZO zones, which requires high-quality forecasting of processes, etc.

State-of-the-Art. Classical approaches to coordination are based on a centralized (for a small number of LCS and a small distance between them) or a hierarchical (for a large number of LCS or a large distance between them) architecture of the coordination system. These approaches were developed mainly for organizational management systems: organization management, military management, etc. For the coordination of local control systems, these principles began to be applied with the development of technological automation (Katrenko & Savka, 2008; Ladanyuk et al., 2012). However, such systems have a rigid structure of connections and are difficult to scale. This makes it difficult to use them in objects with frequent and rapid changes in requirements.

The decrease in the cost of automation tools, the development of the Internet of Things (IoT), and the exponential growth of the microcontrollers power have opened up a promising way to solve the problem: the use of decentralized coordination with smart coordinators in each local control system.

In order to investigate the phenomenon of coordination, a consortium comprising 12 European research centers was established, tasked with conducting the Project Control for Coordination of Distributed Systems (CON4COORD or C4C) (Schuppen & Villa, 2015). An insightful review and analysis of distributed coordination systems is expounded in (Ge et al., 2020). These inquiries primarily focus on hierarchical systems, with particular emphasis on the allocation of functions across different levels.

Decentralized systems are also the subject of research. A feature of distributed decentralized systems is the significant uncertainty of subsystem interaction parameters, the incompleteness of the system, and the lack of complete information about the state of other subsystems that are in direct communication with a separate subsystem. The article (Bakule, 2008) reviews past and current results in the field of decentralized control of large-scale complex systems.

Depending on system typology, the responsibilities associated with decentralized system control encompass synchronization, decentralized stabilization, single-level coordination, peer control, among others (Gong & Aldeen, 1997). This spectrum of inquiry spans linear and nonlinear systems, continuous and discrete domains, and encompasses optimal and adaptive control paradigms, robust methodologies, and systems featuring elements of artificial intelligence (Shaikh et al., 2014).

The development of "small" energy, the appearance of a large number of solar wind and other power plants made it practically impossible to centralize control. As a result, a number of works dedicated to decentralized control in energy appeared (Zabet I. & Montazeri M. (2010); Aghdam et al., 2019).

The analysis of modern research in the direction of the creation and research of distributed systems of automatic control allows us to identify several urgent problems, in the solution of which great attention is paid to the application of decentralized systems:

- Scalability and efficiency. Distributed automatic control systems often work with a large amount of data and require high performance. The development of methods for optimizing the operation of such systems in order to increase speed and efficiency remains an urgent problem;
- Reliability and security. Ensuring the reliability of distributed systems is an important task. This covers both protection against accidental failures and ensuring resistance to failures of individual elements of the system;
- Adaptability to changes. Given the variability of conditions and inputs, it is important to develop methods that allow distributed automatic control systems to adapt to changes in real time and provide optimal control.

Related works. Decentralized control of distributed systems has certain advantages, but it raises a number of theoretical and practical problems. In many studies, attention is paid to the problem of stability and quality of control in decentralized systems. Works (Boyd et al., 1994; Šiljak & Stipanović, 2000). Use an approach based on the Lyapunov functions of the block-diagonal structure and the construction of systems of matrix linear inequalities based on them. Special tools for stability analysis are also being developed (Elmahdi et al., 2015).

Mirkin B. (1992) proposed the concept of adaptive decentralized control with model coordination is. At the same time, it is assumed that information about the state of reference models of all local subsystems is available to local controllers. In (Jianget et al., 2018). the problem of consensus for a class of heterogeneous linear multi-agent systems is investigated. The consensus problem is decomposed into a set of local tracking problems with local cost functions determined from the tracking errors. Based on game theory, the set of stable optimal policies of the entire network falls into a Nash equilibrium. In order to find the Nash solution, a distributed algorithm has been developed that calculates control strategies using an iterative process.

For the implementation of methods of coordination of control systems of multi-zone technological objects, learning systems are preferred, as they are able to flexibly adjust to the specifics of each zone control. However, a difficult problem for the application of such systems is the creation of training datasets. The article (Zhang et al., 2021). investigates model-based methods in multi-agent reinforcement learning (MARL). The complexity of dynamic sampling and the complexity of component sampling in MARL are determined.

In modern works on the coordination of decentralized control in technical systems, the training of agents in multi-agent systems based on neural networks is mainly considered. At the same time, model-based multi-agent reinforcement learning is used. Wang et al. (2022) provides an overview of existing research on model-based MARL, including theoretical analysis, algorithms, and applications, and analyzes the advantages and potential of model-based MARL.

The article (Akramizadeh et al., 2010). developed model-based reinforcement learning for a group of agents with self-interests and sequential action selection based on traditional priority sorting. The learning process is considered as an extensive Markov game.

In previous works, the author of this article presented decentralized distributed control systems (DCS) for multi-zonal objects (MZO) as two interacting layers (Fig. 1) (Dubovoi & Yukhymchuk, 2022): a layer of physical interaction of MZO zones and a layer of information interaction of local control systems (LCS). The communication between MZO zones and the corresponding LCS is carried out through regulators (executive devices) and sensors (feedback). The LCS implements both the implementation of a specific control law (relay or linear) and coordination with other LCSs in order to optimize the state of the MZO according to a given global criterion. In a decentralized system, optimization is not carried out simultaneously by all LCSs, but in a sliding mode based on a combined local-global criterion (Yukhymchuk & Dubovoi & Kovtun, 2022).

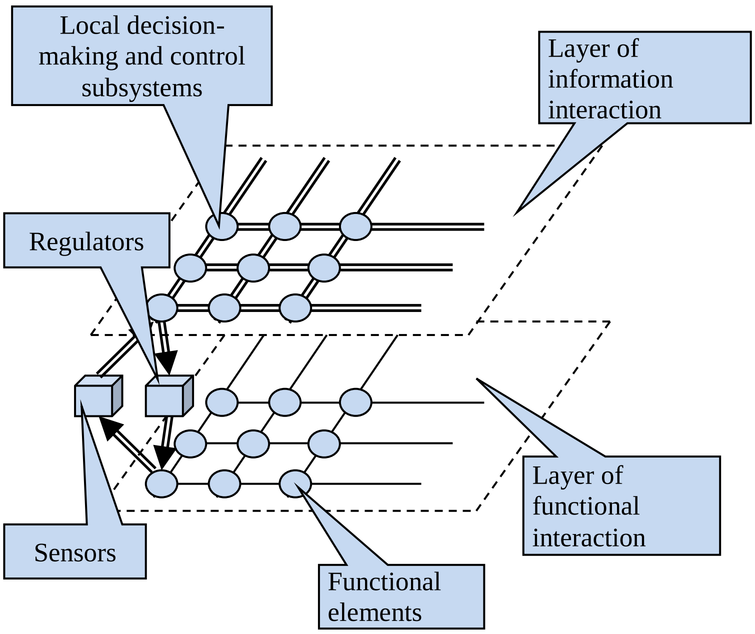


Figure 1. Generalized image of a decentralized control system

In preceding research (Dubovoi & Yukhymchuk, 2022) specific facets of the architecture pertaining to the decentralized coordination system governing multi-zone thermal entities were examined. The constituents comprising the state control system of the DCS utilizing neural networks are delineated as follows:

- DCS itself, which is divided into control and control zones;
- a set of agents implementing the sliding decentralized coordination algorithm;
- datasets characterizing the given and actual states of the DCS;
- training procedure for coordinators;
- user interface - system operator;
- stream processors for control of data exchange between system components.

The coordinator agent in (Dubovoi & Yukhymchuk, 2022) uses the following principles, methods and algorithms

- A model of a distributed cyber-physical system with decentralized control, in which each element of a multi-zone object is controlled by a local control system, which in turn receives optimal control settings from a local coordinator;
- A sliding algorithm for control of the coordination sequence to ensure the stability of the system;
- The principle of close action, which defines a set of coordinators (cluster), which provide information to the sliding coordination center;
- Methods of estimating uncertain parameters;
- Methods of forecasting the effects and states of the system and the use of forecasting during coordination control.

All the specified elements of the coordinator are combined in a synergistic interaction as modules shown in fig. 2. As a result, the coordination function is implemented by a software complex, but another version of the implementation of such an architecture using neural networks is also possible.

The initial data for optimal coordination for MZO with n zones are:

- Vector $\mathbf{F}[n]$ of given zone states;
- Vector $\mathbf{V}_0[n]$ of optimal states of zones (vector of LCS presettings is the initial data and objective of coordination recursively);
- Vector $\mathbf{V}[n]$ of actual zone states (resource accumulated in the zone);
- Vector $\mathbf{X}[n]$ of the amount of input raw materials to each MZO zone, for the processing of which the accumulated resource is spent;
- State $u[1]$ of the surrounding environment.

At the output of the coordinator, only one value $v_0[1]$ is been obtained - the presetting for the corresponding LCS.

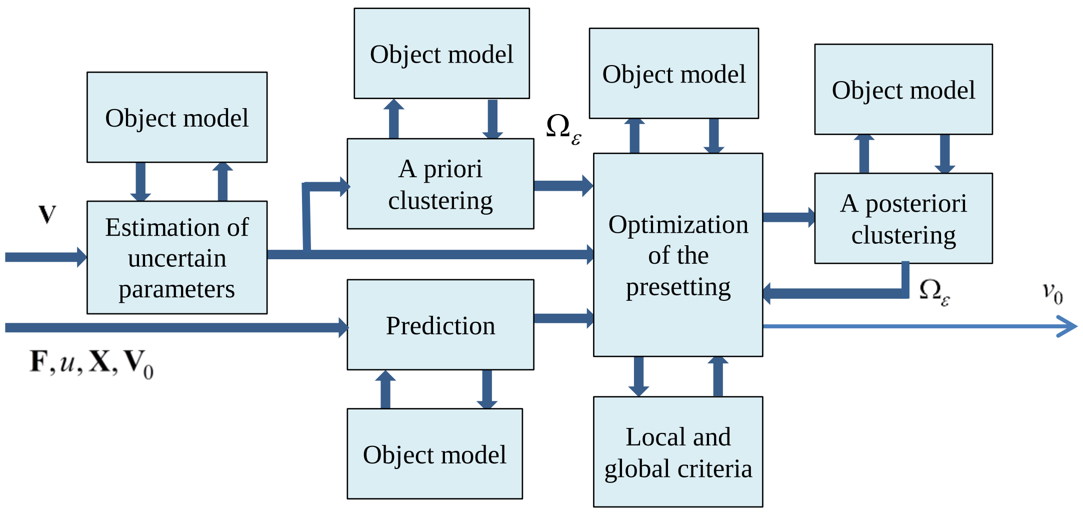


Figure 2. Scheme of interaction of coordinator modules

Objectives and problems. The conducted analysis showed that the existing approaches to solving the problem of creating effective decentralized control systems for multi-zonal objects have certain shortcomings. These shortcomings are mainly caused by too much idealization of the MZO model. And since the MZO model, according to the scheme of connections of the coordinator modules in Fig. 2, is used to perform all procedures: assessment, forecasting, optimization, clustering, their accuracy is insufficient.

The problem can potentially find resolution through the development of coordinators based on neural networks (NN). However, another problem arises here: the training of such coordinators. After all, training a neural network of coordinator requires a substantially extensive marked dataset due to the complexity of the coordination task. Unfortunately, in most cases such datasets do not exist for MZO.

The purpose of this work is to develop approaches to the creation of training datasets and training procedures for DCS neuron coordinators (NC).

Proposed Method.

Statement of research. The volume and structure of the training dataset depends on the type and size of the NN, which in turn depend on the characteristics of the problem for which they are intended to be solved. Therefore, the first task of the work is to formulate an approach to choosing the type and size of NN.The training method also affects the formation of the training dataset. Given that multi-zone objects can differ significantly from each other in terms of spatial structure, static and dynamic characteristics, final pre-operational training of the system is impossible. Therefore, in this work, we will plan a two-stage training: preliminary training based on the DCS simulation (model-based training) and further training in the process of operation. Accordingly, it is necessary to formulate an approach to the creation of datasets and training procedures at both stages.

Basic approach. Model-based training of the neural coordinator can be performed according to various scenarios, depending on the structure of the neural coordinator and the features of the structure of the multi-zone object.

According to the architectural depiction of the coordinator illustrated in Fig. 2, neural networks must implement the following functions:

- DCS modeling;
- Clustering;
- Optimal estimation of parameters;
- DCS state predicting;
- Optimization of the presetting according to the local-global criterion.

These functions can be implemented by separate neural networks (Separated Neural Coordinator, SNC) and the NN of the architecture most suitable for this task should be chosen for the performance of each function or with the help of one NN (Mono-Neural Coordinator, MNC) which has $4n+1$ inputs, 1 output and architecture united all the mentioned tasks in accordance with Fig. 2.

GRU (Gated Recurrent Unit) is the most suitable type of neural network for solving the problem of vector time series forecasting.

To solve the problem of optimal estimation of parameters, where one of the vectors is calculated by solving a system of differential equations, it is advisable to apply the modification of recurrent neural networks GRU (Gated Recurrent Unit)

Convolutional regression neural network is the most suitable type of neural network for solving the problem of presetting optimization.

For the task of clustering a set of zones given by a weighted graph, the most suitable type of neural network is GNN (Graph Neural Network). Among the modifications of GNN, GATs (Graph Attention Networks) are the most acceptable - these networks pay attention to the proximity of vertices in the graph and can take edge weights into account.

The most difficult task is the creation of a neural model of DCS, since it consists of interconnected models of zones, resource flows and LCS, and each of them are described by differential equations of different orders. As a result, the neural model should contain several recurrent layers, several (at least two) layers with lateral connections, as well as a Convolutional Neural Network with Vector Output.

The MNC coordinator solves the problem comprehensively, therefore, taking into account the order of the DCS dynamics model, the need to predict states, as well as the mutual influence of zones and coordinators, the MNC architecture should be built on the base of a GRU network with an additional recurrent layer and convolutional layer. Note that the MNC coordinator must receive all $4n+1$ input data and have an appropriate number of items in each layer, while NNs for individual tasks have a much smaller dimension.

Thus, there are 4 basic scenarios of NC learning, shown in Fig. 3, and their combinations in the case of partial merging of NNs.

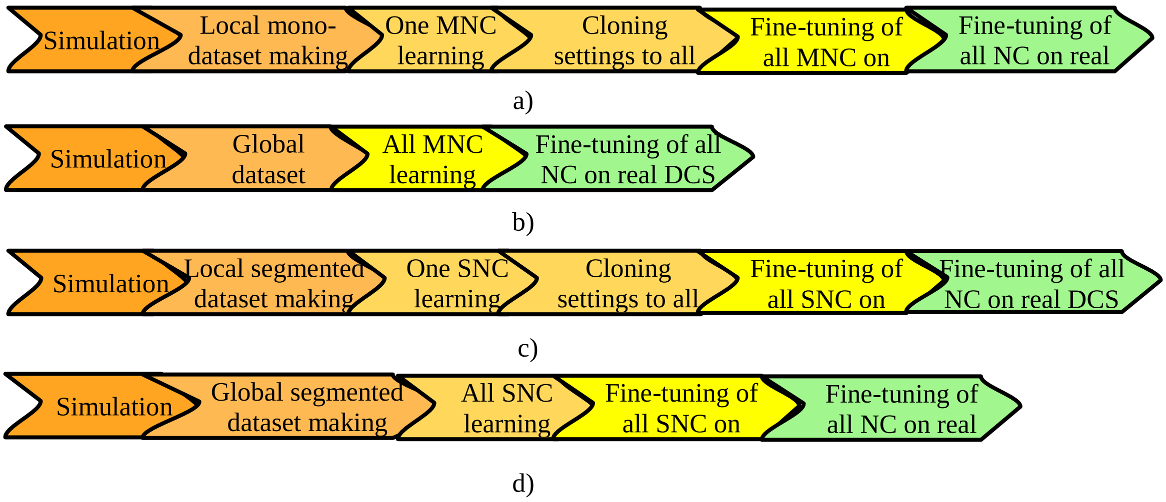


Figure 3. Basic scenarios for training neural coordinators

According to the scenario in Fig. 3a, one MNC coordinator is trained. The dataset is created for this purpose with the help of simulation of one NC of the most characteristic LCS, i.e., the one that is affected by the largest number of external influences. Based on the training results of this NC, other DCS coordinators are cloned. In the next step, individual adjustment of all MNCs is carried out on the simulation model and further training on the physical object. The last stage of additional training is already in the process of implementing the system.

According to the scenario in Fig. 3b, MNC coordinators of all LCS are simultaneously trained. For this, a global dataset of inputs and outputs of all NCs is created using simulation. This method does not require additional training on a simulation model, but requires a much larger dataset. In the last step, additional training is carried out at the physical object.

According to the scenario in Fig. 3c, one SNC coordinator is trained. Since the NC is segmented, that is, each function of the coordination task is performed by a separate NN, and a corresponding segmented dataset for the most characteristic LCS is created with the help of simulation for this. Based on the results of the training of all segments of the NN of this NK, other DCS coordinators are cloned. In the next step, individual fine-tuning of all relevant SNC segments is carried out on a simulation model and then fine-tuning on a physical object.

According to the scenario in Fig. 3d, SNC coordinators of all LCSs are simultaneously trained. For this, a global segmented dataset is created for each module of all NCs using simulation. This method requires fine-tuning the joint action of all modules of each SNC on a simulation model. In the last step, additional training is carried out at the physical object.

Numerical Experiments, Results and Discussion

The study of the effectiveness of these scenarios was carried out on the basis of a decentralized MZO control system with 4 zones located in series (Fig. 4).

Zones 1...4 affect each other by exchanging resource flows (heat, raw materials, etc.). Local zone control systems exchange information necessary for coordination through a data transmission system.

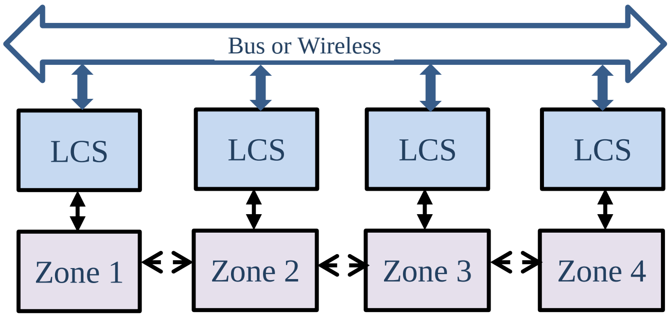
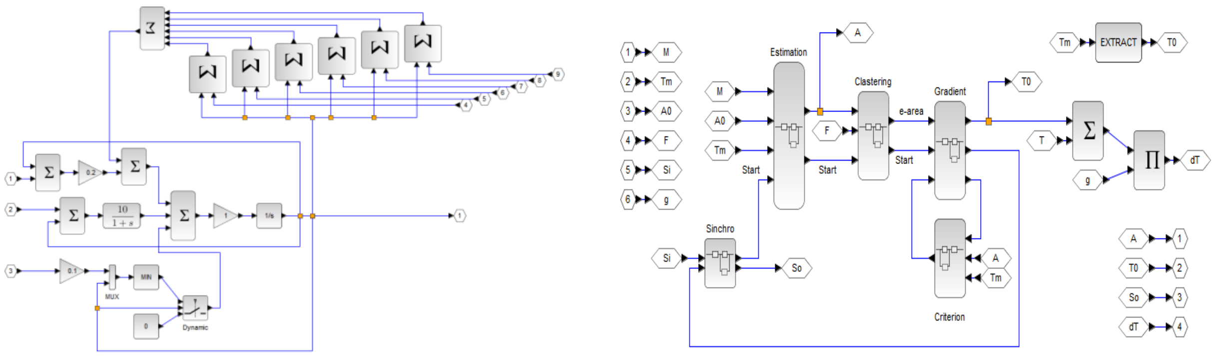


Figure 4. Basic structure for simulation

The simulation model of the system was created on the Scilab/Xcos platform using the block library for modeling distributed control systems (Dubovoi & Yukhymchuk, 2022; Yukhymchuk, 2022). Fig. 5a shows the model of one zone with LCS, Fig. 5b shows the model of the coordinator.



a) b)

Figure 5. Simulation models of the controlled zone and the coordinator

Neural coordinators MNC were created using ChatGPT with the Tensor-flow library. An example of the network generation function is shown in Fig. 6.

```
def create_model():
    model = Sequential([
        Input(shape=(5, 17)),  # Input layer for historical data
        GRU(64, return_sequences=True),  # Recurrent layer GRU
        GRU(32),  # The second recurrent layer GRU
        Dense(1)  # Output layer with one neuron
    ])
    return model
```

Figure 6. Function for MNC generation from ChatGPT

Let's calculate the number of configurable parameters for the proposed model:

1) First layer GRU:
   - Parameters for the input vector: $17\times64+64=108817\times64+64=1088$;
   - Parameters for recurrent matrix: $64\times64+64=416064\times64+64=4160$;

   Total: $N_1=1088+4160=52481088+4160=5248$.
2) Second layer GRU:

- Parameters for the input vector: 64×32+32=208064×32+32=2080;
- Parameters for recurrent matrix: 32×32+32=105632×32+32=1056;

Total: N_2=2080+1056=31362080+1056=3136.

3) Output fully connected layer:
- Parameters for scales: N_3=32×1+1=3332×1+1=33

Thus, the total number of configurable parameters for this model is:

N_p=5248+3136+33=84175248+3136+33=8417.

To create a dataset from the input, output and internal parameters of the coordinators, the module for outputting simulation data to a file shown in Fig. 7 was used. Pre-normalization of the simulation data was not required because the stability of the system had been verified in previous work and the parameter ranges were specified in the simulation presetting. The resulting data array was divided into a training (80%) and testing (20%) subset.

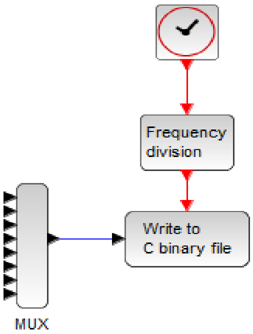


Figure 7. Module for outputting simulation data to a file

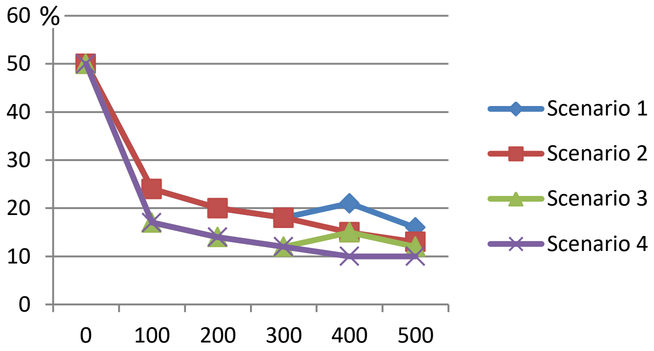


Figure 8. Comparison of the effectiveness of the scenarios

Dependencies of the RMSE of the output vector of presettings of the neural network relative to the same vector of presetting of the simulation on the testing sample and the number of training epochs were studied. The results of the study are shown in Fig. 8. The dataset was generated by a simulation with random input influences $\mathbf{F}[n]$, $\mathbf{X}[n]$, u with a uniform distribution and an exponential correlation function. The probability distribution of the components of the vector $\mathbf{F}$ of a given state of the zones is normal, with mathematical expectation $m_f = 37$, variation $\frac{\sigma_f}{m_f} = 20\%$, correlation interval τ_f =5 simulation cycles. The probability distribution of the components of the raw material flow vector $\mathbf{X}$ is uniform, with mathematical expectation $m_x = 10$, variation $\frac{\sigma_x}{m_x} = 10\%$, correlation interval τ_x =5 simulation cycles. The probability distribution of the state of the environment u is uniform, with a mathematical expectation $m_f = 15$, range is $[0; 30]$, correlation interval τ_u =15 simulation cycles.

Let's estimate the required number of dataset blocks. To prevent overfitting, we will use the "10x rule". This means that the number of training data should be 80,000, and the total number of simulation cycles and, accordingly, data blocks (including test ones) should be $N_d=10^5$.

Research results have shown that with a sufficiently large number of epochs, all scenarios lead to approximately the same result. However, segmented NCs approach the final learning level faster. In the cases of Scenarios 1 and 3, where one coordinator was first trained for 300 epochs and then the results were cloned to other coordinators, the overall RMSE increased immediately after cloning, but quickly decreased to close to Scenarios 2 and 4 as a result of fine-tuning.

At the same time, the training of segmented coordinators requires the creation of similarly segmented datasets: separately for each task which complicates the training process.

Conclusions. The studies allow us to conclude about the possibility of transition from the algorithmic solution of the problem of coordination of local control systems of multi-zonal objects to the solution using neural networks. Two variants of neural coordinators are considered: segmented and non-segmented. The advantage of the segmented version is a slightly higher learning speed. However, the non-segmented version is more versatile because it is not tied to a defined coordinator structure.

In the course of the study, the last stage of the scenarios - supplementary training on a real physical object - was not performed. However, it can be assumed that after such fine-tuning, the advantages of MNC will become even more noticeable, since such fine-tuning will correct the shortcomings of imitation.

Further research is expected to be directed to the task of choosing the optimal architecture of a neural network for decentralized coordination of MZO.

Acknowledgements. The author gratefully acknowledge the invaluable support and expertise provided by the staff of the Department of Computer Control Systems at Vinnytsia National Technical University. Their help and resources were essential to the successful completion of this research. Special thanks are due to colleagues who offered constructive insights that significantly enhanced the work.

Conflicts of Interest. The author declares no conflict of interest.

Стаття надійшла: 05.05.2024.

References

Aghdam A.G., Miller D.E., & Davison E.J. (2019). *Decentralized Control of Large-Scale Systems*, Springer, ISBN/EAN: 9781441960146.

Akramizadeh, A., Afshar, A., B Menhaj, M., & Jafari, S. (2010). Model-based Reinforcement Learning in Multiagent Systems in Extensive Forms with Perfect Information. *IFAC Proceedings Volumes*, Vol. 43, Issue 8, pp. 487–494. Elsevier BV. https://doi.org/10.3182/20100712-3-fr-2020.00080

Bakule, L. (2008). Decentralized control: An overview. Annu. Rev. Control., 32, pp. 87-98. [Online]. Available: https://api.semanticscholar.org/CorpusID:205659095.

Boyd, S., El Ghaoui, L., Feron, E., & Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory. Society for Industrial and Applied Mathematics*. SIAM, Philadelphia, 1994. https://doi.org/10.1137/1.9781611970777

Coordination Control of Distributed Systems. (2015). In J. H. van Schuppen & T. Villa (Eds.), Lecture Notes in Control and Information Sciences. Springer International Publishing. https://doi.org/10.1007/978-3-319-10407-2

Dubovoi V.M. & Yukhymchuk M.S. (2022). *Decentralized coordination control of distributed cyber-physical systems with continuous objects*, VNTU, Vinnytsia. [in Ukrainian]

Elmahdi, A., Taha, A. F., Sun, D., & Panchal, J. H. (2015). Decentralized Control Framework and Stability Analysis for Networked Control Systems. *Journal of Dynamic Systems Measurement and Control*, vol. 137. doi: 10.1115/1.4028789.

Ge, X., Han, Q.-L., Ding, L., Wang, Y.-L. & Zhang, X.-M. (2020). Dynamic Event-Triggered Distributed Coordination Control and its Applications: A Survey of Trends and

Techniques. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. Vol. 50, Number 9, pp. 3112–3125. https://doi.org/10.1109/tsmc.2020.3010825

Gong Z. & Aldeen M. (1997). Stabilization of Decentralized Control Systems, *Journal of Mathematical Systems, Estimation, and Control*, Vol. 7, No. 1, pp. 1–16,

Jiang, H., Liu, X.-K., He, H., Yuan, C., & Prokhorov, D. (2018). Neural Network Based Distributed Consensus Control for Heterogeneous Multi-agent Systems. In *2018 Annual American Control Conference (ACC). 2018 Annual American Control Conference (ACC).* IEEE. pp. 5175–5180. https://doi.org/10.23919/acc.2018.8431744

Katrenko, A.V., & Savka I.V., (2008). Mechanisms of coordination in complex hierarchical systems. *Bulletin of Lviv Polytechnic National University* . Vol. 631, pp. 156–166. [Online]. Available: http://vlp.com.ua/files/16_1.pdf. [in Ukrainian]

Ladanyuk, A., Shumygai, D. & Boiko, R. (2012). System task of co-ordination in the technological complexes of continuous type. *Collected Works of Kirovohrad National Technical University Machinery in agricultural production, industry machine building, automation*. Vol. 25-1. pp. 288-294. [Online]. Available: http://dspace.nuft.edu.ua/jspui/bitstream/123456789/4444/1/Sh_3.pdf. [in Ukrainian]

Mirkin B.M. (1992). A new decentralized model reference adaptive control scheme for large scale systems. in *Prepr. 4th IFAC International Symposium on Adaptive systems in control and signal processing*, Grenoble, France, pp. 645–650.

Shaikh P.H., Nor.B.M., Nallagownden P., Elamvazuthi I., & Ibrahim T. (2014). A Review on Optimized Control Systems for Building Energy and Comfort Control of Smart Sustainable Buildings. *Renewable and Sustainable Energy Reviews*, Vol. 34, pp. 409–429. https://doi.org/10.1016/j.rser.2014.03.027.

Šiljak, D.D., & Stipanović, D.M. (2000). Robust stabilization of nonlinear systems: The LMI approach. Mathematical Problems in Engineering, vol. 6, pp. 461–493.

Wang, X., Zhang, Z., & Zhang, W. (2022). Model-based Multi-agent Reinforcement Learning: Recent Progress and Prospects (Version 1). *arXiv*. https://doi.org/10.48550/ARXIV.2203.10603

Yukhymchuk, M., Dubovoi, V., & Kovtun, V. (2022). Decentralized Coordination of Temperature Control in Multiarea Premises. *Complexity*, vol. 2022, Article ID 2588364. https://doi.org/10.1155/2022/2588364.

Zabet I. & Montazeri M. (2010). Decentralized control and control systems for power industry via multiagent systems technology, in *PEOCO 2010 - 4th International Power Engineering and Optimization Conference, Program and Abstracts*, pp. 549–556, IEEE. doi: 10.1109/PEOCO.2010.5559254.

Zhang, W., Wang, X., Shen, J., & Zhou, M. (2021). Model-based Multi-agent Policy Optimization with Adaptive Opponent-wise Rollouts (Version 3). *arXiv*. https://doi.org/10.48550/ARXIV.2105.03363.