

# THE MULTIDISCIPLINARY JOURNAL OF SCIENCE AND TECHNOLOGY

## VOLUME-5, ISSUE-5

### DEVELOPMENT OF A MATHEMATICAL MODEL FOR SIMULATING A DECENTRALIZED CONTROL SYSTEM FOR COLLABORATIVE ROBOT NETWORKS

Vladyslav Yevsieiev <sup>1</sup>, Jafar Ababneh <sup>2</sup>, Svitlana Maksymova <sup>1</sup>, Amer Abu-Jassar <sup>3</sup>

<sup>1</sup>Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv National University of Radio Electronics, Ukraine

<sup>2</sup>Cyber Security department, Faculty of Information Technology, Zarqa University, Zarqa, Jordan

<sup>3</sup>Department of Computer Science, College of Information Technology, Amman Arab University, Amman, Jordan

#### ABSTRACT

The article presents the development of a mathematical model for modeling a decentralized control system for a network of collaborative robots. The main attention is paid to the analysis of the behavior of agents in a distributed environment, taking into account the level of trust, risk, adaptability and probability of failures. The proposed model is implemented in the Python programming language and allows for simulations with the ability to visualize changes in the main parameters over time. The modeling demonstrated that a high level of risk and a decrease in trust critically affect the efficiency of interaction in the network, especially with an increase in the frequency of failures. The experiments confirmed the dependence of the stability of the system on the initial conditions and values of key parameters, which allows optimizing the architecture of such systems to increase reliability. The results obtained are important for further improvement of decentralized control algorithms in the field of collaborative robotics.

**Keywords:** Decentralized Control, Collaborative Robots, Robot Networks, Trust Level, Risk, Mathematical Modeling, Fault Tolerance, Adaptability, Python, Simulation, Agent Interaction, Autonomous Systems.

#### INTRODUCTION

In the current conditions of Industry 5.0 development, there is a rapid growth of interest in decentralized control systems, in particular in the field of robotics, where networks of collaborative robots serve as the basis for building flexible, scalable and adaptive production systems [1]-[12]. The rejection of centralized approaches in favor of decentralization provides increased resistance to failures, reduced load on individual nodes, accelerated decision-making processes and improved autonomy of robotic agents.

The relevance of the research lies in the need to develop mathematical models that allow simulating and analyzing the behavior of complex distributed systems in a dynamic environment with a high level of uncertainty, in particular with the possibility of failure of individual elements, changes in the structure of the communication network, or fluctuations in trust between agents. The concept of trust as a mechanism that ensures effective cooperation between robots, allows assessing risks and making informed decisions in conditions of partial information is of particular interest. As robotic systems are increasingly deployed for mission-critical tasks in search and rescue, security, medical, and logistics operations [13]-[25], the ability to predict the dynamics of trust and risk between agents becomes increasingly important. In this context, a model that takes into account the probability of failure, variable adaptability, initial trust, and risk tolerance provides a deeper understanding of the behavior of the system as a whole and lays the foundation for improving its performance.

This suggests the need to develop more robust adaptation strategies, and also justifies the importance of this research for designing robotic systems of the future that can function under uncertainty, autonomously respond to environmental changes, and maintain stable interaction without centralized control. Different methods and approaches can also be used here [26]-[49].

#### LITERATURE REVIEW

In the process of developing a mathematical model for modeling a decentralized control system for a network of collaborative robots, it is important to analyze existing scientific approaches, technologies and architectures used in related research. A review of previous works allows us to identify key trends, limitations and promising directions for the development of decentralized control systems in multi-robot networks.

Authors K. V. R. Devi offers in his study an overview of modern approaches to robot swarms and cooperative control in multi-component systems, focusing on the principles of self-organization, decentralization and scalability. This allows us to better understand the basics of coordination between agents, reducing the need for centralized control [50]. The results obtained can be directly used when building basic trust and interaction algorithms in a mathematical model of decentralized control.

In his work, A.E. Celik investigates synchronization between collaborative robots using 5G technologies. The authors demonstrate the high efficiency of decentralized synchronization due to low latency and reliable connection [51]. Such results can be used for network simulations with realistic communication characteristics when building a model.

In the article, O. Omotuyi proposes a method for training scalable decentralized controllers for heterogeneous robot swarms using graph neural networks, which provides effective coordination without centralized control [52]. The results of this study can be used as a theoretical basis for building an adaptive level of interaction model in robot networks.

In the study, M. Bashabsheh describes a centralized control model for transport robots, focused on optimizing movement and task distribution [53]. Although the study focuses on centralized systems, it provides a contrasting basis for justifying the need and advantages of decentralization in modeling.

Authors C. Urrea proposes an algorithmic implementation of manipulator coordination using RRT (Rapidly Exploring Random Trees), which provides adaptive behavior under constraints [54]. This approach can be integrated into a mathematical model to take into account the local adaptation of each agent.

X. Ma develops a robust constraint-based control based on the Udvadiya–Kalaba approach for collaborative robot modules, which allows to ensure the accuracy of actions during cooperation [55]. The proposed control methods can be used to model physical constraints in collaborative robotics systems.

In the article Y. Chen investigates the use of large language models (LLM) in centralized and decentralized multi-robot interaction systems, evaluating their performance in complex tasks [56]. This allows adapting decision-making models in a decentralized approach to modern conditions and technologies.

In the study, J. Gao proposes a three-level distributed simulation architecture for the coordination of heterogeneous robots, which allows modeling the interaction of different types of agents in a shared environment [57]. The results obtained are especially useful for expanding the capabilities of the mathematical model taking into account the heterogeneity of agents.

The analysis of the research of modern authors confirms the high relevance of developing a mathematical model of a decentralized control system for networks of collaborative robots. In particular, the importance of self-organization, adaptability, scalability and fault tolerance, which are key requirements for decentralized systems, is emphasized. Works devoted to the use of 5G, graph neural networks, motion planning algorithms and new approaches to control demonstrate the potential of integrating such solutions into simulation models. Most publications emphasize the limitations of centralized approaches and the advantages of distributed decision-making in dynamic and uncertain environments. This creates a scientific basis and justification for the need to build a mathematically based model of decentralized interaction, which will become the basis for software simulation and further practical implementation in cyber-physical production systems.

#### **DEVELOPMENT OF A MATHEMATICAL MODEL OF A DECENTRALIZED CONTROL SYSTEM FOR A NETWORK OF COLLABORATIVE ROBOTS**

## THE MULTIDISCIPLINARY JOURNAL OF SCIENCE AND TECHNOLOGY

### VOLUME-5, ISSUE-5

Within the framework of these studies, the following mathematical model of a decentralized collaborative robot network management system is proposed, based on the principles of self-organization, swarm intelligence, distributed decision-making and flexible interaction protocols, taking into account scalability, fault tolerance, adaptability and reduction of decision-making time:

– collaborative robot system model, represented as a set:

$$\mathcal{R} = \{R_1, R_2, \dots, R_n\}, \quad (1)$$

$\mathcal{R}$  – a set of collaborative robots;

$R_i$  – collaborative robot ( $R_i \in \mathcal{R}$ ).

Each robot  $R_i$ , is described by a state vector. A state vector is a set of variables that completely describe the state of the robot, necessary for decision-making, control, and interaction with other agents in the system.

$$x_i(t) \in \mathbb{R}^m, \quad (2)$$

$$x_i(t) = \begin{bmatrix} p_i(t) \\ v_i(t) \\ e_i(t) \\ s_i(t) \\ l_i(t) \\ r_i(t) \\ t_i(t) \end{bmatrix} \in \mathbb{R}^m, \quad (3)$$

$p_i(t) \in \mathbb{R}^3$  – spatial position, coordinates of the robot's position in the workspace (e.g.  $x, y, z$  y 3D or  $x, y, z$  y 2D) it determines location for navigation, collision avoidance, and interaction with others;

$v_i(t) \in \mathbb{R}^3$  – velocity/orientation vector, the speed of movement or orientation in space that is taken into account for coordinating movement. Used in alignment algorithms, collision avoidance, and trajectory planning;

$e_i(t) \in \mathbb{R}$  – remaining energy, i.e. battery or charge level (% or relative value). Determines whether the robot can continue working or must return to the base/to charge;

$s_i(t) \in \mathbb{R}^k$  – sensor data, values from sensors (e.g. distances to objects, temperature, human detection, etc.). Provides perception of the environment for decision-making, dynamic adaptation, and interaction with the environment;

$l_i(t) \in \mathbb{R}$  – current logical load or task load. A value that reflects how loaded the robot is with current tasks (can be 0–1 or in tasks in the queue). Helps in balancing the load in the network, determining the optimal distribution of tasks;

$r_i(t) \in \mathbb{R}^q$  – risk/trust parameters, assessing the reliability of the robot in specific conditions (e.g., trust factor when interacting with a person, risk of failure). Provides priorities in choosing a partner for joint actions or task delegation;

$t_i(t) \in \mathbb{R}$  – response time or decision lag, the delay between receiving information and generating action. Critical for real-time systems to reduce latency in coordination.

$m = 3$  (position) + 3 (speed) + 1 (energy) +  $k$  (sensors) + 1 (load) +  $q$  (risk/trust) + 1 (delay). Total vector  $x_i(t)$  size depends on hardware specifications and number of sensor inputs (i.e.  $m = 10-20$ ).

The developed representation in 2-3 allows: to carry out full-fledged remote modeling and control; to create adaptive behavioral models; to implement distributed control without the need for a central node; to ensure the system's resistance to failures, environmental changes, and flexible interaction with a person:

## THE MULTIDISCIPLINARY JOURNAL OF SCIENCE AND TECHNOLOGY

### VOLUME-5, ISSUE-5

– the dynamics of the state change of a collaborative robot, we will describe through the equations of a nonlinear dynamic system that models how the state of each robot  $R_i$  changes over time, taking into account control, environmental influences, interaction with other agents, and noise:

$$\dot{x}(t) = f_i(x_i(t), u_i(t), \eta_i(t), \xi_i(t)), \quad (4)$$

$\dot{x}(t)$  – derivative of the state vector, or rate of change of the robot's state. This is the rate of change in the physical and informational characteristics of the robot at time  $t$ . It determines how the parameters  $x_i(t)$  (e.g., position, velocity, energy, sensor data) change over time;

$x_i(t) \in \mathbb{R}^m$  – current state vector of the robot. Describes the current state of the robot (position, speed, energy, sensors, trust, etc.). The state is the starting point for evaluating the dynamics of the system, which the control function  $f_i$  should change;

$u_i(t) \in \mathbb{R}^p$  – vector of control influences. This is a set of commands or actions that are sent to the robot – for example, commands to move, change the trajectory, perform a task. Controls the behavior of the robot in accordance with the set goal, ensures adaptation to current conditions;

$\eta_i(t) \in \mathbb{R}^q$  – environmental inputs. Contains information about external conditions that affect the robot's behavior: obstacles, location of people, noise, temperature, humidity, movement of other objects. Allows the robot to adapt to changes in the environment, to work in the real world;

$\xi_i(t) \in \mathbb{R}^r$  – noise/uncertainties/disturbances. Represents uncontrolled, random, or unexpected variables that can affect the system: sensor noise, transmission delays, communication failures, measurement errors. Taking noise into account allows you to build robust algorithms that can function in conditions of unpredictability;

$f(\cdot)$  – a function of the robot dynamics that determines how the state of the robot changes depending on all the listed factors. Can be described linearly, nonlinearly, stochastically or logical-linguistically (for example, fuzzy). A key control component, allows you to develop control, learning, coordination algorithms.

– swarm interaction and self-organization of collaborative robots, it is proposed to use local rules based on the Bosds algorithm to maintain formation and avoid collisions. The BOSDS algorithm (Bio-inspired Optimization for Swarm-based Decentralized Systems) is an intelligent optimization method based on the principles of bio-inspired swarm behavior and decentralized decision-making. It imitates the mechanisms of interaction between agents in natural systems, such as ants or bees, to achieve effective coordination of collaborative robots without centralized control. BOSDS provides flexibility, adaptability and stability of the system in conditions of unpredictable environmental changes.

$$u_i(t) = w_1 \cdot Cohesion_i(t) + w_2 \cdot Separation_i(t) + w_3 \cdot Alignment_i(t) + w_4 \cdot Goal_i(t). \quad (5)$$

$Cohesion_i(t)$  - attraction to the center of mass of neighbors:

$$Cohesion_i = \sum_{j \in N_i} (x_j - x_i);$$

$Separation_i$  - avoiding crowds:

$$Separation_i = \sum_{j \in N_i} \frac{x_i - x_j}{||x_i - x_j||^2};$$

$Alignment_i$  - speed coordination:

$$Alignment_i = \sum_{j \in N_i} (v_j - v_i);$$

$Goal_i$  - movement towards a goal or local task;

$w_1, \dots, w_4$  – coefficients that determine behavior (can adapt dynamically).

– distributed decision making, will be performed by each agent independently. The agent independently decides which action to perform, using the principles of collective choice:

## THE MULTIDISCIPLINARY JOURNAL OF SCIENCE AND TECHNOLOGY

### VOLUME-5, ISSUE-5

$$u_i(t) = \arg \max_{a \in A_i} [U_i(a, x_i, \eta_i)], \quad (6)$$

$u_i(t)$  – this is the control action (control) that the robot chooses  $R_i$  at a point in time  $t$ , to achieve the best result in your current circumstances;

$a \in A_i$  – set of available actions for the robot  $R_i$ , which can change depending on its state, task, or environment;

$A_i$  – set of possible actions between which the robot  $R_i$  chooses at time  $t$ ;

$U_i(a, x_i, \eta_i)$  – a utility function that evaluates the appropriateness of action  $a$ , given the current state of the robot  $x_i$  and information from the environment or from other agents  $\eta_i$ . It allows the robot to make optimal decisions in a decentralized environment;

$x_i$  – robot  $R_i$  state vector, which displays its current position, speed, energy level, and other internal parameters;

$\eta_i$  – contextual information, which may include signals from other robots, sensor data, or environmental characteristics that influence the choice of actions.

Expression 6 formalizes how each robot in the system independently, based on its own information and limited context, makes a decision that ensures decentralized but coordinated behavior of the swarm.

– the communication topology of the network of collaborative robots, can be represented as a graph:

$$G(t) = (V, E(t)), \quad (7)$$

$V$  – collaborative robots;

$E(t) \subseteq V \times V$  – variable set of connections (depending on environmental conditions, the presence of a connection).

Provided that the connection graph  $G(t)$  has to be connected at all times ( $t$ ):

$$\forall t: G(t) \text{ is connected.} \quad (8)$$

Requirement (8), that the connection graph  $G(t)$  must be connected at every time moment ( $t$ ), is critical for the efficient functioning of a decentralized collaborative robot control system. If the graph is disconnected, that is, there are robots or groups of robots that do not have a communication path to each other, this leads to information loss, unavailability of the global context, and disruption of coordination between agents. Connectivity ensures that all robots can transmit or receive data - directly or through intermediate nodes - which allows maintaining consistency in decision-making, supporting common system goals, and quickly adapting to changes in the environment. This requirement also ensures resilience to failures of individual agents, since information paths can be redirected through alternative routes in the network. In the context of Industry 5.0, where trust and security of human-robot interaction are important, constant connectivity is the foundation for reliable, adaptive, and intelligent collective control.

– failure adaptability model, implemented using the local control recovery function upon loss of communication:

$$\eta_i(t) = \begin{cases} \sum_{j \in N_i} w_{ij} x_j(t), & \text{connection is present,} \\ \hat{\eta}_i(t) = \text{"prediction if connection is lost".} & \end{cases} \quad (9)$$

In the failure adaptability model (9), the parameter  $\eta_i(t)$  plays the role of an assessment or adaptive response of agent  $R_i$  to information from neighbors in the network. This parameter allows

## THE MULTIDISCIPLINARY JOURNAL OF SCIENCE AND TECHNOLOGY

### VOLUME-5, ISSUE-5

to ensure the stability of the system in case of loss of communication with other agents or partial failure of communications. Its structure describes two scenarios:

- if the connection with neighbors  $j \in N_i$  is preserved, then:

$$\eta_i(t) = \sum_{j \in N_i} w_{ij} x_j(t), \quad (10)$$

$w_{ij}$  – weights of trust in a neighbor  $j$ ;

$x_j(t)$  – neighbor state vector at time  $(t)$ ;

$N_i$  – the set of all neighbors of an agent  $i$ .

– if the connection is lost, then  $\eta_i(t) = \hat{\eta}_i(t)$  – a predicted value based on a local model or historical data. This can be obtained through extrapolation methods, local predictors, or even machine learning to maintain an estimate of the overall network state.

Thus, the parameter  $\eta_i(t)$  allows the agent to continue adaptive work even in cases of communication disruption, maintaining consistency in distributed decision-making.

– the fault tolerance metric allows you to assess how effectively a collaborative robot system is able to continue stable and coordinated work in the event of partial or complete failures of individual robots, loss of communication, or unpredictable changes in the environment. It allows you to quantitatively measure the system's ability to:

- quickly adapt to changes in network topology;
- restore functionality without centralized intervention;
- maintain consistency of decisions in a group of robots even with limited information;
- minimize efficiency losses in the process of interaction with a person.

Thanks to this metric, you can optimize control algorithms, design more reliable distributed systems, and ensure a high level of security within the framework of Industry 5.0.

$$R(t) = \frac{|\mathcal{R}_a(t)|}{|\mathcal{R}|}, \quad (11)$$

$\mathcal{R}_a(t)$  – number of active (not broken) robots.

- the aggregate objective function is the global cost function that the system minimizes:

$$J = \sum_{i=1}^n [\lambda_1 \cdot Energy_i + \lambda_2 \cdot Time_i + \lambda_3 \cdot Risk_i + \lambda_4 \cdot Trust_i], \quad (12)$$

$Energy_i$  – robot  $R_i$  energy consumption, which characterizes the efficiency of task performance and is important for the autonomy of the system;

$Time_i$  – time spent by the robot  $R_i$  to perform tasks, which affects overall system performance;

$Risk_i$  –  $p$ th level of risk associated with the robot's  $R_i$  actions, that takes into account the possibility of accidents, errors or dangers to humans;

$Trust_i$  – level of trust in the robot  $R_i$ , which reflects the quality of human-machine interaction in the context of safety and reliability;

$\lambda_1, \lambda_2, \lambda_3, \lambda_4$  – Weights that determine the relative importance of each criterion in the overall objective function. They allow you to adjust the model to specific conditions: for example, increase the value  $\lambda_3$  for systems with high safety requirements or  $\lambda_4$  for human-oriented tasks.

General function  $J$  allows you to find optimal solutions that balance resource costs, speed of execution, security, and quality of human interaction.

**DEVELOPMENT OF A PROGRAM FOR MODELING DECENTRALIZED CONTROL SYSTEMS FOR COLLABORATIVE ROBOTS**

The choice of the Python programming language for modeling decentralized control systems for a network of collaborative robots is quite justified given its versatility, clear syntax, and powerful ecosystem of scientific and technical libraries [58]-[60]. Python allows you to quickly implement complex mathematical models, integrate artificial intelligence modules, and effectively visualize data, which is extremely important in the field of Industry 5.0, where the interpretation of modeling results is key to analyzing the effectiveness of robotic systems. The PyCharm development environment was chosen because of its convenience in debugging code, support for working with virtual environments, autocompletion, integration with Git, and the ability to organize large projects with many modules. PyCharm also provides a convenient graphical interface for viewing errors, working with libraries, and structured development, which greatly facilitates the implementation and testing of decentralized control models. The numpy library is used for efficient work with numerical arrays, data generation, calculations on state vectors, and processing mathematical operations in real time. It is an integral part of most scientific calculations in Python. The networkx library allows you to create, modify, and analyze graphs, structures that are ideal for describing a network of robot interactions. In this context, graphs are the basic model of communication between agents in a decentralized system, where each robot is a node and links are edges with or without weights. It is thanks to network that you can track the change in network topology over time, the loss of links, and their impact on decision-making. The matplotlib.pyplot library is necessary for visualizing the interaction process, trust, risk, efficiency, and other variables, which allows the researcher to quickly analyze and present simulation results. Visual graphs are an integral part of assessing the behavior of complex systems over time. Finally, the random module is used to simulate random events, such as link failures, variations in agent behavior, or changes in trust levels, which makes the model more realistic and adaptive.

Together, these tools allow you to build a flexible, scalable, and fault-tolerant model of a decentralized control system that can take into account the human factor and interaction within modern cyber-physical systems of Industry 5.0. Below is a fragment of software code that implements the basic structure of such a model:

```
class Robot:
    def __init__(self, idx):
        self.id = idx
        self.state = np.random.rand(4) # [Energy, Time, Risk, Trust]
        self.active = True
        self.neighbors = []
        self.trust = trust_initial
    def update_state(self, neighbor_states):
        if not self.active:
            return

        if neighbor_states:
            avg_state = np.mean(neighbor_states, axis=0)
            diff = np.linalg.norm(self.state - avg_state)
            if diff > adaptivity_threshold:
                self.state = (self.state + avg_state) / 2
        # Risk and Trust Dynamics
        self.state[2] = min(1.0, self.state[2] + np.random.uniform(-0.05, 0.05)) # Risk
        self.state[3] = max(0, min(1.0, self.trust - 0.01 * self.state[2])) # Trust
```

This code fragment is necessary for modeling the individual behavior of each collaborative robot in a decentralized control system. It allows taking into account the influence of neighbors, adaptive response to changes in the environment, as well as risk dynamics and the level of trust in other robots. This provides a realistic approximation to the conditions of Industry 5.0, where flexibility, self-organization and safe interaction with humans play a key role. Thanks to such an implementation, it is possible to model fault tolerance and evaluate the effectiveness of the network in conditions of partial loss of communication.

```
# ----- Building a network of interaction -----
def create_communication_graph(n, p):
    G = nx.erdos_renyi_graph(n, p)
    while not nx.is_connected(G):
        G = nx.erdos_renyi_graph(n, p)
    return G
```

This code snippet is responsible for creating a network of interactions between collaborative robots in the form of a connection graph. It uses the Erdős-Rényi model to generate a random graph with the probability of connection between nodes. The main condition is that the graph must be connected, that is, each robot must have a path to any other in the network. This is critical for implementing decentralized control, where information must be freely distributed among all participants. This approach allows us to investigate the scalability and resilience of the system to loss of connection.

```
# Failure testing
for r in robots:
    if r.active and random.random() < failure_rate:
        r.active = False
```

This code snippet simulates the probability of failure of each robot in the system at each step of the simulation. If the robot is active and the random number is less than the specified failure rate, the robot becomes inactive, which simulates its failure. This mechanism allows us to assess the system's resilience to failures and the network's adaptability to the disappearance of individual nodes. This is important for testing the reliability of a decentralized control system in conditions of dynamic changes.

```
# Collecting neighbor states
for r in robots:
    if r.active:
        for neighbor_id in r.neighbors:
            if robots[neighbor_id].active:
                neighbor_states[r.id].append(robots[neighbor_id].state)
```

This code snippet implements the process of collecting information about the state of neighboring robots in the network for each active robot. It goes through each robot, and if it is active, it checks its neighbors. If the neighbor is also active, its current state vector is added to the list of neighbor states. This mechanism allows each robot to take into account the behavior of its closest network participants, which is critical for implementing distributed control, adapting to changes, and forming collective decisions.

### **EXPERIMENTAL STUDIES AND ANALYSIS OF THE RESULTS OBTAINED**

The purpose of conducting experiments on modeling a decentralized collaborative robot network management system is to study the impact of key parameters - such as the number of robots,

# THE MULTIDISCIPLINARY JOURNAL OF SCIENCE AND TECHNOLOGY

## VOLUME-5, ISSUE-5

the probability of communication, the failure rate, the adaptability threshold, the initial level of trust and risk tolerance – on the overall efficiency, stability, adaptability and safety of the collective work of the robotic system in conditions of uncertainty and dynamic changes, which meets the challenges of Industry 5.0.

The task is to create conditions for analyzing the behavior of the system when changing the specified parameters in order to identify patterns that contribute to the optimization of decentralized management and increasing the level of trust between agents and humans.

To implement the experimental modeling, a Microsoft Surface Pro 4 laptop is used, processor: Intel Core i7-6650U, 3.4 GHz, 2 cores, 4 MB cache; Memory. RAM: 16 GB; Hard drive: 512 GB SSD; Graphics adapter: Intel Iris Graphics 540.

Software: Windows 10 OS, the program was created in Python in the PyCharm 2022.2.3 environment.

The expected results are:

- assessment of the scalability of the system when changing the number of robots and the probability of connection;
- determination of failure resistance thresholds for different failure rate levels;
- analysis of the influence of the level of adaptability and trust on the stability of agent behavior and the effectiveness of collective decision-making.

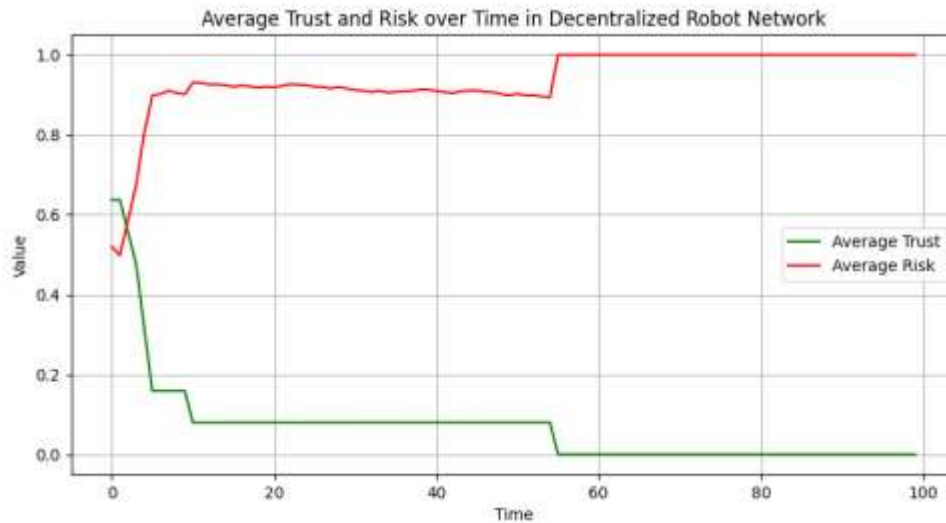
Such a modeling experiment will allow identifying optimal parameter configurations for the development of reliable and safe collaborative robotics systems that can be used in new generation industrial production, logistics complexes, rescue operations, smart factories and in conditions where autonomy and the ability to collectively adaptively respond to environmental changes are critical.

The input parameters for modeling decentralized collaborative robot control systems and analysis of the results obtained are presented in Table 1.

**Table 1:** Input parameters for modeling decentralized control systems for collaborative robots

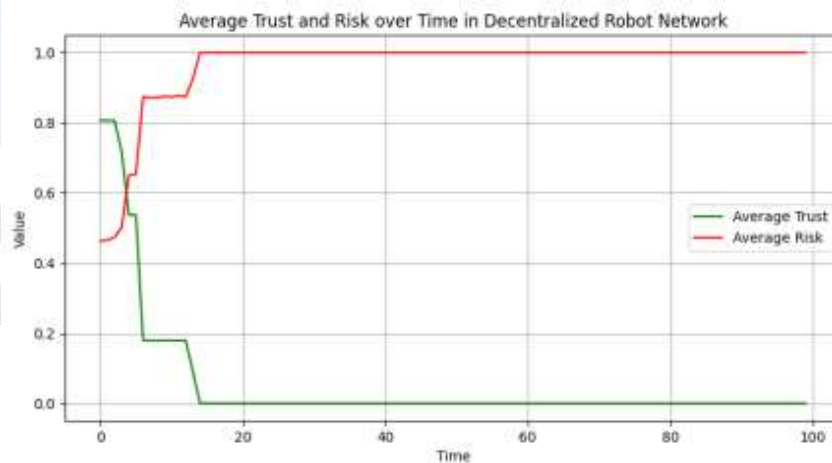
№	Input parameters						
	Number of robots	Simulation time	Probability of connection between robots	Robot failure probability	Adaptability threshold	Initial level of trust	Risk tolerance
1	10	100	0.4	0.1	0.2	0.8	0.3
2			0.6	0.2	0.6	0.9	0.5
3			0.3	0.3	0.3	0.6	0.4
4			0.7	0.5	0.8	0.4	0.7
5			0.5	0.01	0.1	0.3	0.8

The obtained results of modeling a decentralized control system for collaborative robots with different input parameters (Table 1) are presented in Figures 1-5.



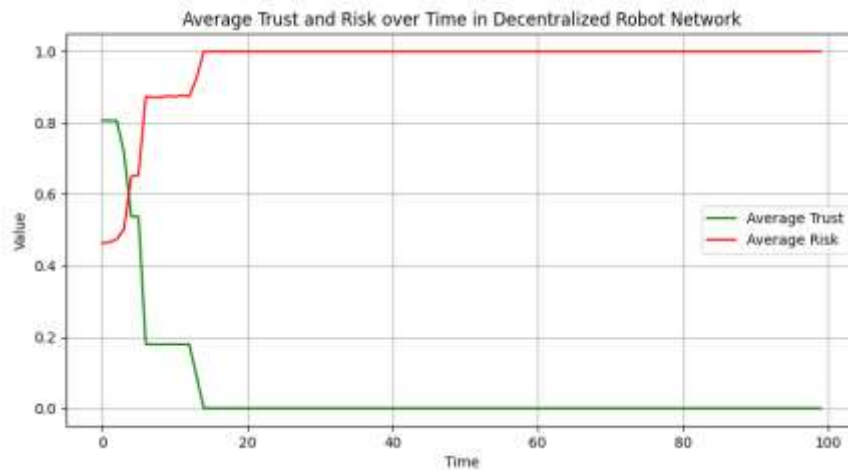
**Figure 1:** Graph of average trust and risk in a decentralized robot network (first simulation experiment)

Figure 1 shows the change in the average level of trust (green line) and risk (red line) in a decentralized network of collaborative robots over 100 time units. There is a rapid increase in risk in the first steps of the simulation to a level close to 0.9, after which it stabilizes and gradually reaches a maximum of 1.0. At the same time, trust sharply decreases from the initial value of about 0.65 to almost zero, which indicates a strong influence of increased risk on the decline in trust between agents. After the 60th iteration, both metrics stabilize, which indicates the completion of adaptation processes or the complete loss of functionality of some robots. This result demonstrates the vulnerability of the system to increased risk and the need to improve trust support mechanisms in conditions of high uncertainty.



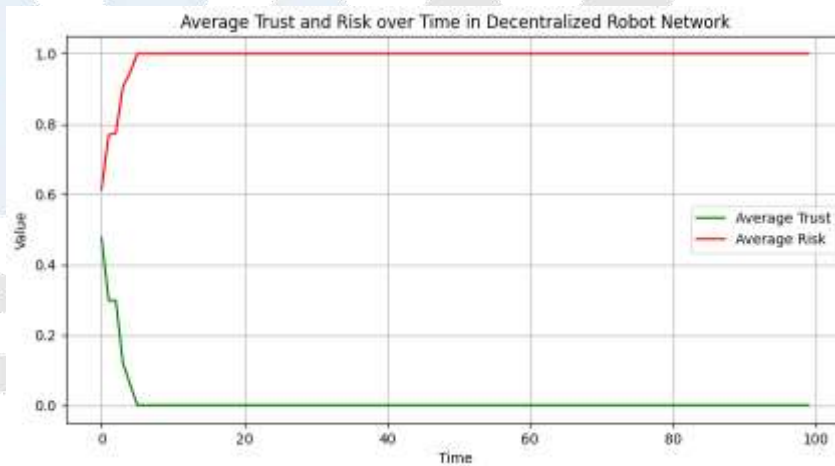
**Figure 2:** Graph of average trust and risk in a decentralized robot network (second simulation experiment)

Figure 2 shows a sharp increase in the average risk level in the collaborative robot network to a maximum value of 1.0 already in the early stages of the simulation, after which the risk stabilizes at this level throughout the simulation time. At the same time, the average trust level rapidly decreases to zero at approximately the 15th step, indicating a critical loss of trust in the system due to the increase in risk or instability in the interaction between agents. This behavior indicates the lack of effective adaptation or self-healing mechanisms in conditions of failures or destabilization, and indicates the need to revise the models of risk assessment and trust maintenance in decentralized systems to ensure their resilience.



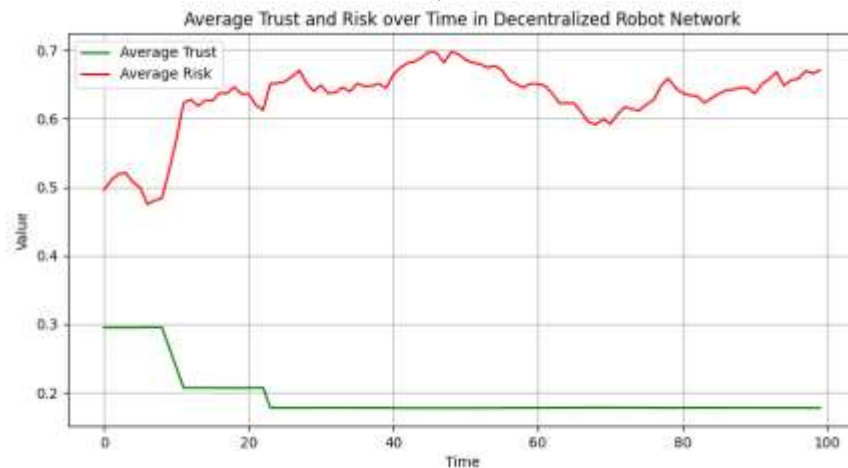
**Figure 3:** Graph of average trust and risk in a decentralized robot network (third simulation experiment)

Figure 3 illustrates the change in the average level of trust (green line) and risk (red line) in a decentralized network of collaborative robots during the simulation. At the beginning, a high level of trust is observed with a relatively low risk, but already in the first ten units of time there is a rapid drop in trust to zero, which is accompanied by a sharp increase in risk to a single value. In the future, the level of trust remains zero, while the risk is stably high. This indicates a violation or degradation of communication or behavior of individual agents in the system, which emphasizes the importance of developing mechanisms for adaptation, self-healing or checking the reliability of information within the mathematical model to ensure the stability of decentralized interaction.



**Figure 4:** Graph of average trust and risk in a decentralized robot network (fourth simulation experiment)

Figure 4 shows the average level of trust and risk in a decentralized network of collaborative robots over 100 time units. At the initial stage, trust decreases from a level of about 0.5 to zero by the 10th step of the simulation, while risk increases rapidly from about 0.6 to a maximum value of 1.0. After that, both indicators remain stable: trust is at zero, and risk is at its maximum. This behavior of the network indicates the emergence of situations in which the interaction between agents becomes unreliable or contradictory, which leads to a complete loss of trust between the elements of the system, in parallel with an increase in the probability of failures. This emphasizes the need to improve the mathematical model to maintain stability in a dynamic environment.



**Figure 5:** Graph of average trust and risk in a decentralized robot network (fifth simulation experiment)

Figure 5 shows the dynamics of changes in the average level of trust and risk in a decentralized network of collaborative robots over 100 time intervals. The average level of trust gradually decreases from an initial value of approximately 0.3 to a stable value of approximately 0.18, after which it remains unchanged. At the same time, the average risk fluctuates between 0.5 and 0.7, demonstrating the unstable behavior of the system. This result indicates the partial preservation of trust between agents, but under conditions of increased uncertainty, which emphasizes the complexity of managing interaction in decentralized conditions. Such behavior indicates the potential need for adaptive risk control mechanisms while maintaining the basic level of trust.

### CONCLUSION

In the course of the work, a mathematical model was developed for modeling a decentralized control system for a network of collaborative robots, which takes into account the key requirements of Industry 5.0, in particular scalability, fault tolerance, adaptability, speed of decision-making, as well as interaction with a person based on trust and security. To implement the task, the Python programming language was used as a flexible tool for quickly building simulation models, using libraries such as NumPy for mathematical calculations, NetworkX for building communication graphs between agents, matplotlib for visualizing the dynamics of indicators over time, and random for modeling random processes in the environment. The result of the modeling was the creation of a system in which each agent (robot) makes decisions based on its own state and the states of its neighbors, which allows reflecting the real principles of decentralized interaction. The probability of robot failure, the ability to adapt to environmental changes, the threshold for responding to risks, and trust as a key factor for safe interaction in a shared environment were taken into account. Analysis of the simulation results showed that with a certain combination of input parameters, such as a high level of risk and frequent failures, the system demonstrates a sharp drop in the level of trust between agents to zero, which leads to a loss of efficiency of the entire network. At the same time, the risk reaches critical values, demonstrating a weak ability to self-heal and limited adaptability of the model. Thus, the results of experimental modeling indicate that the basic model requires further improvement to ensure long-term stability and flexibility of collaborative robot networks. The revealed dependence of the level of trust on risk confirms the importance of developing mechanisms to reduce uncertainty, increase transparency of decision-making and maintain the integrity of information exchange between agents.

The presented model can be used as a basis for designing control systems in Industry 5.0 conditions, where autonomy, interaction with people, and reliability of decision-making play a critical role. It allows not only to study the behavior of the system under different parameters, but also to test new strategies for communication, adaptation, and fault tolerance, which can be extremely useful

## THE MULTIDISCIPLINARY JOURNAL OF SCIENCE AND TECHNOLOGY

### VOLUME-5, ISSUE-5

when creating complex distributed robotic systems in industry, search and rescue operations, or robotic systems of the future.

#### REFERENCES

1. Abu-Jassar, A., Al-Sukhni, H., Al-Sharo, Y., Maksymova, S., Yevsieiev, V., & Lyashenko, V. (2024). Building a Route for a Mobile Robot Based on the BRRT and A\*(H-BRRT) Algorithms for the Effective Development of Technological Innovations. *International Journal of Engineering Trends and Technology*, 72(11), 294-306.
2. Chala, O., Yevsieiev, V., Maksymova, S., & Abu-Jassar, A. (2025). Mathematical Model Based on Multi-Agent Reinforcement Learning (MARL) And Partially Observable Markov Decision Process (POMDP) for Modeling Cargo Movement for a Mobile Robots Group. *Multidisciplinary Journal of Science and Technology*, 5(4), 480-489.
3. Maksymova, S., Abu-Jassar, A., Gurin, D., & Yevsieiev, V. (2024). Comparative Analysis of methods for Predicting the Trajectory of Object Movement in a Collaborative Robot-Manipulator Working Area. *Multidisciplinary Journal of Science and Technology*, 4(10), 38-48.
4. Yevsieiev, V., Abu-Jassar, A., Maksymova, S., & Gurin, D. (2024). Human Operator Identification in a Collaborative Robot Workspace within the Industry 5.0 Concept. *Multidisciplinary Journal of Science and Technology*, 4(9), 95-105.
5. Nevliudov, I., Yevsieiev, V., Maksymova, S., & Chala, O. (2025). A Small-Sized Robot Prototype Development Using 3D Printing. *acta mechanica et automatica*, 19(1).
6. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2021). Neural networks as a tool for pattern recognition of fasteners. *International Journal of Engineering Trends and Technology*, 69(10), 151-160.
7. Abu-Jassar, A. T., Al-Sharo, Y. M., Lyashenko, V., & Sotnik, S. (2021). Some Features of Classifiers Implementation for Object Recognition in Specialized Computer systems. *TEM Journal: Technology, Education, Management, Informatics*, 10(4), 1645-1654.
8. Sotnik, S., Mustafa, S. K., Ahmad, M. A., Lyashenko, V., & Zeleniy, O. (2020). Some features of route planning as the basis in a mobile robot. *International Journal of Emerging Trends in Engineering Research*, 8(5), 2074-2079.
9. Baker, J. H., Laariedh, F., Ahmad, M. A., Lyashenko, V., Sotnik, S., & Mustafa, S. K. (2021). Some interesting features of semantic model in Robotic Science. *SSRG International Journal of Engineering Trends and Technology*, 69(7), 38-44.
10. Lyashenko, V., Abu-Jassar, A. T., Yevsieiev, V., & Maksymova, S. (2023). Automated Monitoring and Visualization System in Production. *International Research Journal of Multidisciplinary Technovation*, 5(6), 9-18.
11. Ahmad, M. A., Sinelnikova, T., Lyashenko, V., & Mustafa, S. K. (2020). Features of the construction and control of the navigation system of a mobile robot. *International Journal of Emerging Trends in Engineering Research*, 8(4), 1445-1449.
12. Lyashenko, V., Laariedh, F., Ayaz, A. M., & Sotnik, S. (2021). Recognition of Voice Commands Based on Neural Network. *TEM Journal: Technology, Education, Management, Informatics*, 10(2), 583-591.
13. Gurin, D., Yevsieiev, V., Abu-Jassar, A., & Maksymova, S. (2024). Using the Kalman Filter to Represent Probabilistic Models for Determining the Location of a Person in Collaborative Robot Working Area. *Multidisciplinary Journal of Science and Technology*, 4(8), 66-75.
14. Yevsieiev, V., Maksymova, S., Gurin, D., & Alkhalaileh, A. (2024). Data Fusion Research for Collaborative Robots-Manipulators within Industry 5.0. *ACUMEN: International journal of multidisciplinary research*, 1(4), 125-137.
15. Maksymova, S., Yevsieiev, V., Nevliudov, I., & Bahlai, O. (2024). Balancing System For A Zoomorphic Spot Type Mobile Robot Development Using An Accelerometer MPU 6050

## THE MULTIDISCIPLINARY JOURNAL OF SCIENCE AND TECHNOLOGY

### VOLUME-5, ISSUE-5

- (GY-521). In 2024 IEEE 19th International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), IEEE, 39-42.
16. Chala, O., Maksymova, S., & Alkhalaileh, A. (2024). Analysis of Systems for Coordination of Enterprise Subsystems Control. *Journal of universal science research*, 2(10), 127-137.
  17. Kuzmenko, O., Yevsieiev, V., Maksymova, S., & Abu-Jassar, A. (2024). Robot Model for Mines Searching Development. *Multidisciplinary Journal of Science and Technology*, 4(6), 347-355.
  18. Yevsieiev, V., Maksymova, S., & Alkhalaileh, A. (2024). Capturing Human Movements in Real Time in Collaborative Robots Workspace within Industry 5.0. *Journal of universal science research*, 2(10), 232-247.
  19. Sotnik S., & et al.. (2022). Key Directions for Development of Modern Expert Systems. *International Journal of Engineering and Information Systems (IJEAIS)*, 6(5), 4-10.
  20. Abu-Jassar, A. T., Attar, H., Amer, A., Lyashenko, V., Yevsieiev, V., & Solyman, A. (2025). Development and Investigation of Vision System for a Small-Sized Mobile Humanoid Robot in a Smart Environment. *International Journal of Crowd Science*, 9(1), 29-43.
  21. Abu-Jassar, A. T., Attar, H., Amer, A., Lyashenko, V., Yevsieiev, V., & Solyman, A. (2024). Remote Monitoring System of Patient Status in Social IoT Environments Using Amazon Web Services (AWS) Technologies and Smart Health Care. *International Journal of Crowd Science*, 8.
  22. Hamdan, M., Kamal, I. W., Abu-Jassar, A., Maksymova, S., & Lyashenko, V. (2025). Prototyping of a two-wheeled mobile robot for sustainable manufacturing development based on triangulation method and software development. *Journal of Theoretical and Applied Information Technology*, 103(8), 3357-3370.
  23. Tahseen A. J. A., & et al.. (2023). Binarization Methods in Multimedia Systems when Recognizing License Plates of Cars. *International Journal of Academic Engineering Research (IJAER)*, 7(2), 1-9.
  24. Sotnik, S., & Lyashenko, V. (2022). Prospects for Introduction of Robotics in Service. *Prospects*, 6(5), 4-9.
  25. Sotnik, S., & Lyashenko, V. (2022). Prospects for Introduction of Robotics in Service. *Prospects*, 6(5), 4-9.
  26. Ahmad, M. A., Baker, J. H., Tvoroshenko, I., & Lyashenko, V. (2019). Computational complexity of the accessory function setting mechanism in fuzzy intellectual systems. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(5), 2370-2377.
  27. Lyashenko, V., Kobylin, O., & Ahmad, M. A. (2014). General methodology for implementation of image normalization procedure using its wavelet transform. *International Journal of Science and Research (IJSR)*, 3(11), 2870-2877.
  28. Lyashenko, V. V., Babker, A. M. A. A., & Kobylin, O. A. (2016). The methodology of wavelet analysis as a tool for cytology preparations image processing. *Cukurova Medical Journal*, 41(3), 453-463.
  29. Lyashenko, V. V., Matarneh, R., & Deineko, Z. V. (2016). Using the Properties of Wavelet Coefficients of Time Series for Image Analysis and Processing. *Journal of Computer Sciences and Applications*, 4(2), 27-34.
  30. Ababneh, J., Abu-Jassar, A., Abuowaida, S., Liubchenko, V., & Lyashenko, V. (2024, December). Evaluation of Three Different Operators for Object Highlighting in Medical RGB Images: Canny, Roberts, and LoG in Independent Color Spaces. In 2024 25th International Arab Conference on Information Technology (ACIT) (pp. 1-7). IEEE.

## THE MULTIDISCIPLINARY JOURNAL OF SCIENCE AND TECHNOLOGY

### VOLUME-5, ISSUE-5

31. Orobinskyi, P., Deineko, Z., & Lyashenko, V. (2020). Comparative Characteristics of Filtration Methods in the Processing of Medical Images. *American Journal of Engineering Research*, 9(4), 20-25.
32. Mousavi, S. M. H., Lyashenko, V., & Prasath, V. B. S. (2019). Analysis of a robust edge detection system in different color spaces using color and depth images. *Computer Optics*, 43(4), 632-646.
33. Lyubchenko, V., Veretelnyk, K., Kots, P., & Lyashenko, V. (2024). Digital image segmentation procedure as an example of an NP-problem. *Multidisciplinary Journal of Science and Technology*, 4(4), 170-177.
34. Orobinskyi, P., Petrenko, D., & Lyashenko, V. (2019, February). Novel approach to computer-aided detection of lung nodules of difficult location with use of multifactorial models and deep neural networks. In *2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)* (pp. 1-5). IEEE.
35. Mousavi, S. M. H., MiriNezhad, S. Y., & Lyashenko, V. An Evolutionary-Based Adaptive Neuro-Fuzzy Expert System as a Family Counselor before Marriage with the Aim of Divorce Rate Reduction. *Education*, 1, 5.
36. Sotnik, S. Overview: PHP and MySQL Features for Creating Modern Web Projects / S Sotnik, V. Manakov, V. Lyashenko // *International Journal of Academic Information Systems Research (IJASIR)*. – 2023. – Vol. 7, Issue 1. – P. 11-17.
37. Lyashenko, V. V., Matarneh, R., Baranova, V., & Deineko, Z. V. (2016). Hurst Exponent as a Part of Wavelet Decomposition Coefficients to Measure Long-term Memory Time Series Based on Multiresolution Analysis. *American Journal of Systems and Software*, 4(2), 51-56.
38. Lyashenko, V. V., Matarneh, R., & Deineko, Z. V. (2016). Using the Properties of Wavelet Coefficients of Time Series for Image Analysis and Processing. *Journal of Computer Sciences and Applications*, 4(2), 27-34.
39. Tvoroshenko, I., Lyashenko, V., Ayaz, A. M., Mustafa, S. K., & Alharbi, A. R. (2020). Modification of models intensive development ontologies by fuzzy logic. *International Journal of Emerging Trends in Engineering Research*, 8(3), 939-944.
40. Matarneh, R., Tvoroshenko, I., & Lyashenko, V. (2019). Improving Fuzzy Network Models For the Analysis of Dynamic Interacting Processes in the State Space. *International Journal of Recent Technology and Engineering*, 8(4), 1687-1693.
41. Lyashenko, V., & et al.. (2016). The Methodology of Image Processing in the Study of the Properties of Fiber as a Reinforcing Agent in Polymer Compositions. *International Journal of Advanced Research in Computer Science*, 7(1), 15-18.
42. Kuzemin, A., Lyashenko, V., Bulavina, E., & Torojev, A. (2005). Analysis of movement of financial flows of economical agents as the basis for designing the system of economical security (general conception). In *Third international conference «Information research, applications, and education* (pp. 27-30).
43. Deineko, Zh., & et al.. (2021). Features of Database Types. *International Journal of Engineering and Information Systems (IJEAIS)*, 5(10), 73-80.
44. Kobylin, O., & Lyashenko, V. (2020). Time Series Clustering Based on the K-Means Algorithm. *Journal La Multiapp*, 1(3), 1-7.
45. Vasiurenko, O., Baranova, V., & Lyashenko, V. (2024). Probability distributions of interest rates on loans and deposits in a study of banking activities. *Multidisciplinary Journal of Science and Technology*, 4(1), 49-56.
46. Omarov, M., Tykha, T., & Lyashenko, V. (2019). Use of Wavelet Techniques in the Study of Internet Marketing Metrics. *Eskişehir Technical University Journal of Science and Technology A-Applied Sciences and Engineering*, 20, 157-163.
47. Ahmad, M. A., Baker, J. H., Tvoroshenko, I., Kochura, L., & Lyashenko, V. (2020). Interactive Geoinformation Three-Dimensional Model of a Landscape Park Using

## THE MULTIDISCIPLINARY JOURNAL OF SCIENCE AND TECHNOLOGY

### VOLUME-5, ISSUE-5

- Geoinformatics Tools. International Journal on Advanced Science, Engineering and Information Technology, 10(5), 2005-2013.
48. Al-Sherrawi, M. H., Lyashenko, V., Edaan, E. M., & Sotnik, S. (2018). Corrosion as a source of destruction in construction. International Journal of Civil Engineering and Technology, 9(5), 306-314.
  49. Lyashenko, V. V., Deineko, Z. V., & Ahmad, M. A. Properties of wavelet coefficients of self-similar time series. In other words, 9, 16.
  50. Devi, K. V. R., & et al. (2024). A review: Swarm robotics: Cooperative control in multi-agent systems. In E3S Web of Conferences, EDP Sciences, 505, 03013.
  51. Celik, A. E., & et al. (2024). Decentralized System Synchronization among Collaborative Robots via 5G Technology. Sensors (Basel, Switzerland), 24(16), 5382.
  52. Omotuyi, O., & Kumar, M. (2024). Learning Scalable Decentralized Controllers for Heterogeneous Robot Swarms with Graph Neural Networks. Journal of Dynamic Systems, Measurement, and Control, 146(6).
  53. Bashabsheh, M. (2024). Comprehensive and Simulated Modeling of a Centralized Transport Robot Control System. International Journal of Advanced Computer Science & Applications, 15(5).
  54. Urrea, C., & et al. (2024). Enhancing Adaptability and Autonomy in Cooperative Selective Compliance Assembly Robot Arm Robots: Implementation of Coordination and Rapidly Exploring Random Tree Algorithms for Safe and Efficient Manipulation Tasks. Applied Sciences, 14(15), 6804.
  55. Ma, X., & et al. (2024). Robust approximate constraint-following control design based on Udwadia–Kalaba approach and experimental validation for the joint module of cooperative robot. Nonlinear Dynamics, 112(3), 1931-1949.
  56. Chen, Y., & et al. (2024). Scalable multi-robot collaboration with large language models: Centralized or decentralized systems?. In 2024 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 4311-4317.
  57. Gao, J., & et al. (2024). Digital Battle: A Three-Layer Distributed Simulation Architecture for Heterogeneous Robot System Collaboration. Drones, 8(4), 156.
  58. Yevsieiev, V., & et al. (2024). A Program for Analyzing the Structure of a Web site Development Using the Parsing Method Based on the Python. Journal of Universal Science Research, 2(4), 172-183.
  59. Gurin, D., & et al. (2024). Web site reliability analysis using the python parsing method. Journal of Universal Science Research, 2(5), 113-126.
  60. Yevsieiev, V., & et al. (2024). Building a traffic route taking into account obstacles based on the A-star algorithm using the python language. Technical Science Research In Uzbekistan, 2(3), 103-112.