

# Machine Learning-Based Optimization of Robotic Perception and Control Processing

Akshith Jitendra Hedge

Faculty of Engineering, Emirates Aviation University, Dubai, UAE

Husnain Raza Qasim

Faculty of Engineering, Emirates Aviation University, Dubai, UAE

Muhammad Abdullah

Faculty of Engineering, Emirates Aviation University, Dubai, UAE

**Abstract:** Real-time systems, such as robotics, industrial automation, SCADA, testing and measurement apparatus, and technical process control systems, are extensively utilized in industry. In many situations, how well an intelligent robot completes its task depends on the characteristics of the robot's sensor and control systems to provide obstacle avoidance, trajectory planning, object recognition, gripper adaptation, and the appropriate clamping force, among other functions. The strategies and techniques for using machine learning to handle sensors and control data in real-time are analyzed in this study, along with successful examples of machine learning applications in the integration of robot sensor and control systems. The robotic systems being studied include (a) adaptive robots with slip displacement sensors and fuzzy logic implementation for processing sensor data, (b) mobile robots that are controlled by magnetism and can move on ceiling and inclined surfaces using neuro-fuzzy observers and neuro controllers, and (c) robots operating in uncharted territory that use statistical learning theory to predict the control system state. All the results gained pertain to the primary components of the two-component robotic system, which consists of a mobile robot and an adaptive manipulation robot on a fixed base for carrying out intricate tasks in unpredictable or non-stationary environments. Creating a structure diagram and describing the chosen technologies, training a neural network to recognize and classify geometric objects, and implementing control system components software are all part of the design and software implementation step. The design of the control system is done using the Swift programming language, and the neural network is made using the Create framework. The primary findings include: (a) extending the intelligent control system's capabilities by increasing the number of recognition classes from three (cube, cylinder, and sphere) to five (cube, cylinder, sphere, pyramid, and cone); (b) using CreateML (YOLOv2 architecture), increasing the validation accuracy (to 100%) for the recognition of five different classes; (c) using Torch library (ResNet34 architecture) to increase the training accuracy (to 98.02%) and testing accuracy (to 98.0%) for the recognition of five different classes in less time and epochs than with Create ML (YOLOv2 architecture); (d) increasing the training accuracy (to 99.75%) and testing accuracy (to 99.2%) for the recognition of five different classes using Torch library (ResNet34 architecture) and fine-tuning technology; and (e) examining the impact of dataset size on recognition accuracy using ResNet34 architecture and fine-tuning technology. The findings can be used to select effective (a) robotic device control design methodologies, (b) machine learning techniques for pattern identification and classification, and (c) computer technologies for robotic device simulation and control system design.

**Keywords:** robotics; sensor; control system; real-time system; machine learning; pattern recognition; classification; fuzzy logic; neural network; canonical decomposition

## 1. OVERVIEW

As technology advances, real-time systems find use in a wide range of industries. Industry uses real-time systems extensively, such as robotics, industrial automation, SCADA, testing and measurement apparatus, and technical process control systems [1–4]. The interest in and amount of study being done on robotic systems has grown because of modern technologies. Human life can be made easier with the help of several studies of such systems in various fields. Robotics, for instance, is now a crucial technology in the automation sector. When completing jobs, robots guarantee maximal accuracy without human error [5]. Current intelligent robots operate efficiently in specific modes and have high dynamic indicator values. Because robots typically lack full capability, controlling them in unknown circumstances is a challenging challenge. Significant functionality and technological potential are provided by the availability of robots equipped with effective remote and tactile sensor systems [3,6].

For contemporary robots to acquire experience and adjust to naturally occurring nonstationary working situations to carry out a variety of tasks, intellectual property is crucial. In recent years, the utilization of service robots operating in unpredictable environments has increased even further [7–10]. Numerous contemporary robots operate in offices, clinics, supermarkets, movie theatres, businesses, etc. [11–13]. Robots must move effectively and safely in the target area [14–16] to collaborate with others and assist with work in a variety of contexts, especially in dynamic environments where people are present. The capabilities of the robot's sensor and control systems, which include trajectory planning, object recognition, gripper adaptation, obstacle avoidance, and more, are often what determine how well an intelligent robot completes its task (drones, unmanned underwater robots, etc.) [17–21].

The exacting specifications of the robot's sensor and control system parameters (indicators) must be met in order to achieve the effective robot's performance in real-time, especially in unfamiliar or uncertain settings. Its primary concerns are: improving the accuracy of the tactile or remote sensors' sensor information; reducing the time it takes for sensor signals to form; processing sensor and control information faster; reducing the time it takes for the robot's control system to make decisions in unpredictable situations or in a dynamic working environment with obstacles; and expanding the functional capabilities of the robots based on the use of effective sensors and fast calculation algorithms.

The perspective instrument for creating sensors and control systems for robots with enhanced technological features is artificial intelligence (AI) techniques and algorithms. Artificial intelligence includes machine learning. Without using conventional programming techniques, machine learning (ML) algorithms create a model for predictions and/or decision-making based on training data (sample data) [22]. Applications for machine learning (ML) include picture recognition, email filtering, speech recognition, computer vision, and human activity recognition, when it is either impossible or difficult to use traditional algorithms to carry out the required tasks [23].

Because robotics and artificial intelligence, including machine learning, increase and amplify human facilities, enhance productive capacity, and progress from simple thinking to human cognitive skills, special attention must be given to the implementation of various machine learning algorithms and approaches to robotics. It creates new possibilities for improving sensor information processing efficiency, identifying the present state of the robot's working area, regulating signal processing to achieve the intended trajectories, automatically generating alternative hypotheses, and making decisions in real time. Neural networks, fuzzy sets, fuzzy logic, reinforcement learning, deep learning, semi-supervised learning, time series analysis, unsupervised learning, and regression analysis are some of the most widely used machine learning algorithm techniques and methodologies. To increase the efficiency of sensor and control information processing in sophisticated multi-component robotic complexes, this work aims to develop, investigate, and implement various machine learning techniques, such as fuzzy logic, neuro systems and networks, combined neuro-fuzzy approaches, and methods of statistical learning theory. These autonomous two-robot systems of a unique type that operate in non-stationary, unpredictable, or unknown working environments are known as multi-component robotic complexes (MCRC). A moving mobile robot, an adaptive robot with a fixed base (a manipulator with an adaptive gripper), sensor systems, and control systems make up MCRC. It is possible to mount the adaptable robot on the moving mobile robot's hull. In MCRC, mobile robots act as mobile motherhips for a fixed-base adaptive robot, delivering the adaptive robot to any target location on the work surface so it may carry out the associated task. Depending on the MCRC missions, the sensor system may include different kinds of remote sensors, video sensors, and tactile sensors. The application of contemporary machine learning techniques can meet the following key requirements: high functional reliability of mobile and adaptive robots; high accuracy of manipulation operations; and high speed of sensor and control information processing.

Several facets of the topic discussed are covered in the remainder of the article. The creation of the problem statement and the examination of published related papers are covered in Section 2. A broad illustration of the suggested fuzzy information processing method for the adaptive robot's sensor system that detects the slip displacement signal and determines the direction of the unknown object slippage is provided in Section 3. The control system of the mobile robot, which can move on both vertical and inclined ferromagnetic surfaces, is presented in Section 4 together with a neuro-controller and a neuro-fuzzy clamping force observer. Based on the canonical decomposition of statistical data, the authors describe in full the prediction process for delivering dependable operating MCRS, comprising robot sensors and control systems, in Section 5. In Sections 6 and 7, the open-source software for building the control system of the adaptive robot is implemented. A convolution neural network is trained to recognize the shape of objects in the robot's working zone using video-sensor data. Section 8 provides a conclusion for the paper.

## 2. THE PROBLEM STATEMENT AND RELATED WORK

Machine learning has become much more important in recent years. Numerous fields of human endeavor, including medical [4,24–31], agriculture [32–36], transportation [37–42], energy production [43,44], finance markets [45], investment policy [46], and research [47], have successfully implemented machine learning techniques. Real-time sensor data processing based on efficient multi-output Gaussian processes [48] and technical object state prediction utilizing canonical decomposition of a random sequence [49] are two practical applications of statistical learning theory.

Since the focus of this article is on improving the efficiency of robot sensor and control information processing through the use of suitable machine learning algorithms, let's examine the peculiarities of applying ML approaches in robotics. Robot control using model-based reinforcement learning [50], robotics and automation using simulation-driven machine learning [51], intelligent and autonomous surgical robotics [28], speed control when building robots of the type "leader-follower" using fuzzy sets and fuzzy logic, and supervised machine learning [52], computer-aided design based on machine learning for space research and control of autonomous aerial robots [53], and robotics and automation using simulation-driven machine learning

### 2.1. Robotics Machine Learning Methods for Industrial Automation

Using robotic devices, Rajawat et al. [55] presents a more recent method of process automation that combines human flexibility and functionality with robotic control and repetitiveness to improve efficiency and product quality using artificial intelligence and machine learning techniques. The use of machine learning techniques (artificial neural networks, random forests) to precisely estimate surface roughness in wire arc additive manufacturing is covered in the paper [56]. In [57], Wang et al. suggest an image processing technique based on machine learning techniques, specifically for the robotic assembly system. Mayr et al. talk about using machine learning and sensor integration to optimize the linear winding process in the production of electric motors [58]. In [59], Al-Mousawi creates a magnetic explosives detection system using a wireless sensor network and machine learning techniques. The use of machine learning approaches for cognitive robotic process automation is examined by Martins et al. in [60]. Using numerous sensor monitoring, Segreto et al. suggest applying several machine learning algorithms for in-process end-point recognition in robot-assisted polishing [61].

### 2.2. Robot Path Planning and Control Using Machine Learning

To strengthen the link between low-level and high-level representations of sensing and planning, respectively, the method (with application to mobile robots) described in [62] makes use of machine learning techniques. A planning path method based on quickly exploring random trees and reinforcement learning SARSA is proposed by Qijie et al. [21] for mobile robots operating in unfamiliar and uncertain surroundings. To improve timeliness, motion accuracy, and safety of gait planning, the paper [63] discusses exoskeleton robot applications and offers a variety of data modes as input parameters to machine learning models. The authors of [64] give an overview of tracking and gait planning issues, as well as deep learning and robotic capture. The basketball-training robot avoids impediments on its way to the target point and offers clever path autonomous planning [65]. In [20], the deep reinforcement learning method is used to describe the robot route planning methodology.

### 2.3. Robot Tactile and Remote Sensors Using Machine Learning for Information Processing

The coupling of machine learning methods and electronic skins is investigated in the review in [66]. The authors show how scientists might build autonomous robots with deployable capabilities by utilizing the most recent advancements in the two fields. To deal with complex situations in real life, they were equipped with proprioceptive and informational sensory capacities. Ibrahim et al. introduce embedded machine learning techniques [67] for tactile data processing from close sensors. Keser et al. recognizes surface roughness using machine learning approaches based on input from fiber optic touch sensors [68]. In [69], proprioceptive sensing is used to train machine learning regression algorithms that anticipate individual wheel slippage in off-road mobile robots. For robot target identification and recognition employing multi-sensor information processing, Wei et al. suggest a fusion approach that applies evidence theory and support vector machines [70]. Using learning from sensory predictions, Martinez-Hernandez et al. employs a tactile robot to explore item shape autonomously and adaptively [71]. Two machine learning methods, specifically a support vector machine and a neural network, are used in [72] to propose a smart capacitive sensor skin with embedded data quality indicator for improved safety in human robot interaction.

#### 2.4. Robot Computer Vision Using Machine Learning

In [73], Joshi et al. show how to use vision-motor feedback to solve a robotic gripping problem using a deep reinforcement learning approach. For robotic machining, a posture assessment system based on the "eye-to-hand" camera was created in [74]. Two distinct methods—sparse regression and LSTM neural networks—are used to increase the accuracy of the estimated pose. A stereo camera and convolutional neural networks—a deep learning technique—are the foundation of Inoue et al.'s [19] machine vision strategy for autonomously navigating robots that avoid obstacles. Mishra et al. examine deep learning-based robotic vision solutions for pedestrian detection in mobile robots' operating zones [75].

#### 2.5. Using Machine Learning to Improve Fault Diagnostics and Reliability

The current condition of the production machinery and robots must be tracked and estimated to boost the productivity of automated industrial operations. Long et al. employ a deep hybrid learning framework (support vector machine and sparse auto-encoder) to exploit attitude data for intelligent fault diagnosis of multi-joint industrial robots [76]. Using an extreme learning machine, Subha et al. examine the issue of sensor defect diagnostics for autonomous underwater robots [77]. In [78], Severo de Souza examines how to improve the production system's dependability by identifying anomalous sensors using data from the wireless sensor network and machine learning algorithms. Based on the real-time use of IoT data and machine learning algorithms, a predictive maintenance system for manufacturing production lines can identify possible failures before they happen in [2]. In order to prevent a robotic error, Kakizono et al. in [79] provide a neural network-based defect detection and classification method with a harmonic sensor. Researchers are continuously refining machine learning techniques [47,80,81] and creating new machine learning solutions [40,82–85] for intelligent robots using both established and novel design methods, approaches, and methodologies, according to the previously discussed analysis of recent publications on the application of machine learning in robotics. It first addresses the novel and specialized tasks of mobile robots on the ground, underwater, and in the air, as well as the high degree of information uncertainty regarding the characteristics of the robot environment, the shifting nature of the parameters of manipulated objects, and the unpredictable behavior of dynamic obstacles. Research into novel strategies for integrating machine learning techniques and algorithms into contemporary robots is also prompted by additional design constraints. Therefore, to provide effective sensor and control information processing, it is reasonably necessary to build new design approaches, algorithms, and models. It will concurrently raise the control indexes of robot navigation-control systems' operation in unpredictable contexts and enhance their design procedures.

To increase the efficiency of robot sensors and control information processing in multi-component robotic complexes that operate in nonstationary, uncertain, or unknown working environments, the article's problem statement focusses on the implementation and investigation of advanced machine learning techniques based on fuzzy sets theory, theory of neuro systems, statistical learning theory, and others. The tasks of the MCRC under consideration might address the initial unknown or variable mass of the objects being managed, as well as the requirement to steer clear of obstacles or adjust the adaptive robot's trajectory when it collides with them [3,6,7,86–93]. In this scenario, the fixed-based adaptive robot ought to be able to recognize the mass of the object and the directions in which the handled object slips. Such adaptable robots can be put at the moving mothership mobile robot as the second robotic component of MCRC, or they can be outfitted with various touch sensors and function independently. It is crucial to guarantee high indicators of control and maneuvering characteristics and to supply the necessary magnetic clamping force between the mobile robot and the working surface for mobile robots that can move on inclined, vertical, or ceiling ferromagnetic surfaces (such as ship hulls) [13,15,17,94–98]. Given that a ferromagnetic surface may be covered in nonmagnetic layers of dead microorganisms, the mobile robot should monitor the working surface parameters and generate a dependable clamping force value for high-reliable MCRC operation in the event of surface disturbances. In some situations, the gap's size might not be stationary. By managing and forecasting the technical status of every MCRC component, it is also essential to maintain and sustain the high functionality of MCRC [34,49,84]. The operability of the associated equipment for sensor and control information processing must be assessed in real-time during routine MCRC operations to guarantee their continued operation and anticipate any problems. The video camera on the manipulator's arm can be used to enable real-time object detection for several crucial MCRC operations. In these situations, it is necessary to classify the identified objects and send the pictures to the control panel, such as a cell phone. A potential method for identifying and classifying various items in photos is to build machine learning models using neural networks using the YOLOv2 and ResNet34 designs. Implementing a simulation approach based on a MoveIt environment that enables the retrieval of manipulator arm configuration files and their transfer to the MCRC's control system is essential for the development of the ideal structure of the control system during design processes.

Finally, let us formulate the aims of this research as: implementing the machine learning algorithms for extension of functional features of adaptive robots; in particular, using fuzzy and neuro net approaches for sensor information

processing within the recognition of the slippage direction of manipulated objects in the robot gripper during its contact with the obstacles; approximating the “clamping force—air gap” nonstationary functional dependence based on a neuro-fuzzy technique for the mobile robot control system, which provides increased reliability for robot movement on inclined electromagnetic surfaces; implementing the statistical learning theory for increasing the efficiency of a robot’s sensor system based on the developed algorithms of prediction control; and developing the machine learning models and corresponding software for recognizing manipulated objects [99] using video—sensor information processing with a discussion of the peculiarities of the convolutional-neural network’s training process.

### 3. THE MACHINE LEARNING TECHNIQUES FOR ADDING SLIP DISPLACEMENT SENSORS TO ADAPTIVE ROBOTS TO EXPAND THEIR FUNCTIONAL PROPERTIES

The use of tactile sensors that detect object sliding between the gripper fingers is one of the effective ways to ascertain the unknown mass of a controlled object and the intended clamping force value [3,6,12,86–93,100–102]. In addition, slip displacement information can be used to correct the trajectory of the robot gripper and the control algorithm, as well as to identify items from a list of options. The implementation of various detection techniques, such as rolling motion, vibration, or altering the configuration of sensitive elements, friction registration, oscillation of the circuit parameters, displacement of the fixed sensitive elements, and others, forms the basis of the design process for the slip displacement sensors (SDS) [89,92,103].

Let's talk about the task of identifying object slippage and slippage direction (in the gripper of the adaptive robot) using machine learning algorithms to process information from tactile sensors. Most SDS merely use slippage as an event (1—true, 0—false) when creating sensor data for the robot control system. Another issue is determining the direction of slippage, which is crucial in cases where a robot gripper meets an unidentified obstruction within the robot's working area [89,103,104]. The obstacles' presence frequently has a random quality. The trajectory of the robot's gripper and the locations of the obstacles in the adaptive robot's working area determine the slippage direction in collision scenarios. (a) Multi-component slips displacement sensors and (b) machine learning methods (fuzzy logic, neural networks) for sensor information processing can be used to obtain information about the direction of slippage. Using a collection of Hall sensors [90], capacitive sensors [89,102,103], or resistance sensors based on the electro-conductive rubber [88], multi-component slip displacement sensors may identify the sensitive rod movement in the unique cavity.

Let's look at a fuzzy logic method that uses the capacitive slip displacement sensor [89,103] shown in Figure 1 to determine the direction of slippage.

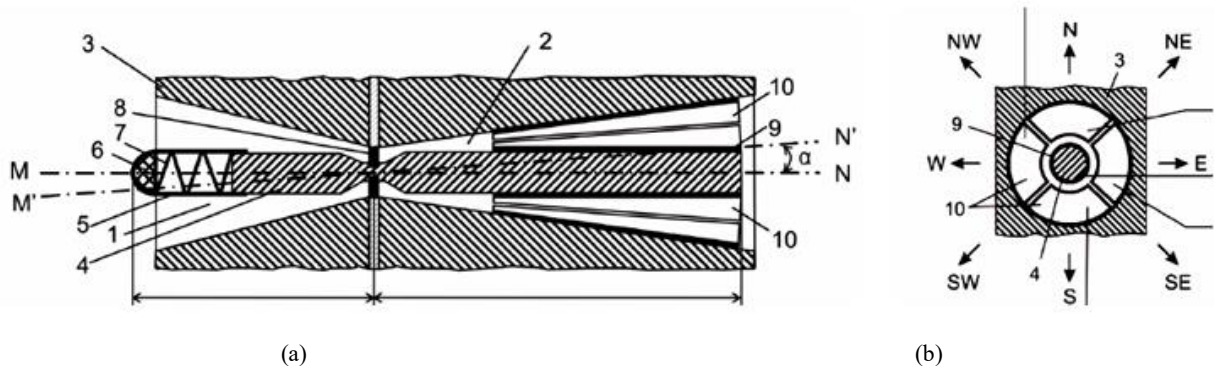


Figure 1. Two-part sensitive rod capacitive slip displacement sensor: (a) front view; (b) right view: 1, 2—the robot's finger's first and second cavities; three—the finger itself; four—a two-part rod (sensitive element); five—the elastic tip; six—the elastic contact surface; seven—the spring; eight—the resilient element; nine—ten— capacitor plates with several components.

The SDS is placed on at least one of the gripper fingers (Figure 1). The recording element consists of four capacitors distributed across the conical surface of the special cavity.

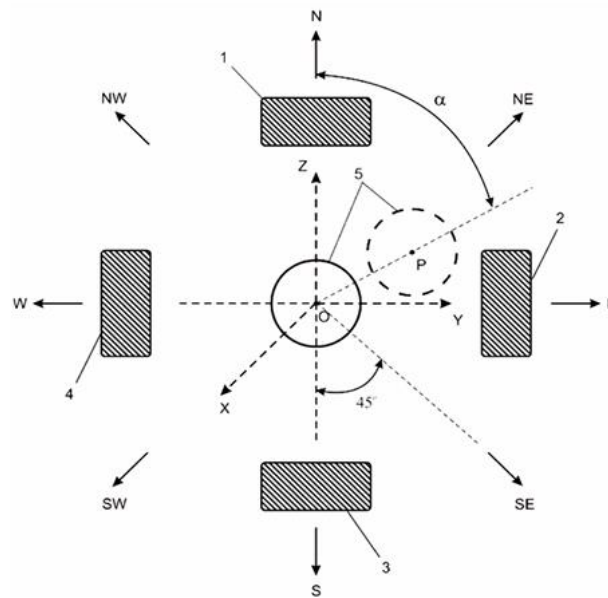


Figure 2: Sensor for determining the direction of slippage point O, the sensitive rod's beginning position prior to object slippage; point P, the sensitive rod's final position following object slippage; 1, 2, 3, and 4—capacitors C1, C2, C3, and C4; 5—deviating rod; The object slippage directions are N ( $0^\circ/360^\circ$ ), NE ( $45^\circ$ ), E ( $90^\circ$ ), SE ( $135^\circ$ ), S ( $180^\circ$ ), SW ( $225^\circ$ ), W ( $270^\circ$ ), and NW ( $315^\circ$ ); The coordinate system OXYZ

Depending on which way the rod moves, the capacitances C1, C2, C3, and C4 change because of the reciprocal movements of plates (9) and (10) in all capacitive elements. A neural network with four input signals (C1, C2, C3, C4) and one output signal (direction  $\alpha$  in the interval  $[0^\circ, 360^\circ]$ ) can be trained using experimental data for various displacement directions and the accompanying changes in C1–C4. The same experimental or simulated data can be utilized to modify the rule consequents of a Mamdani-type fuzzy system, which is an alternative to a fuzzy-rule-based method. Figure 3 depicts the topology of the Mamdani fuzzy system (FS) for processing input from slip-displacement sensors, while Figure 4 displays a portion of the fuzzy rule base.

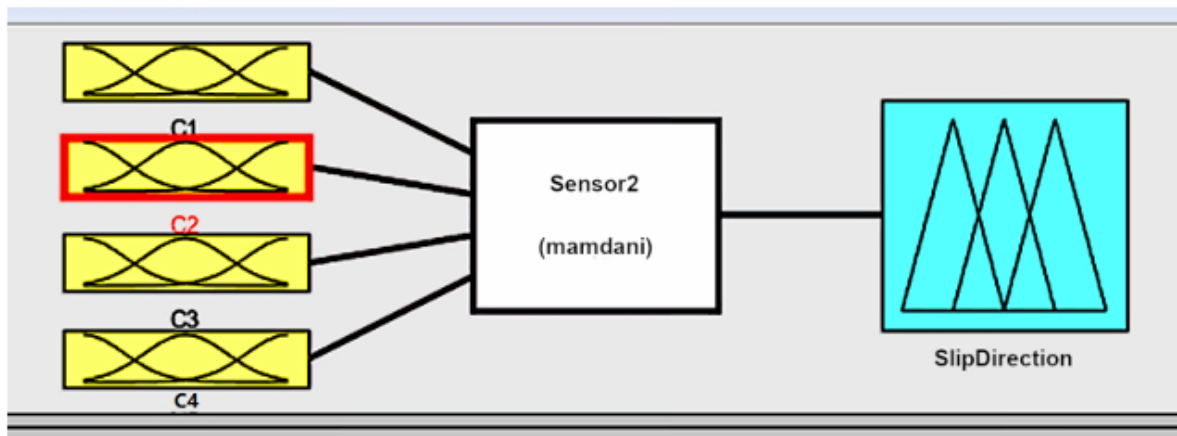


Figure 3. The fuzzy system's structure, which uses three language terms for input signals and nine linguistic terms for output signals

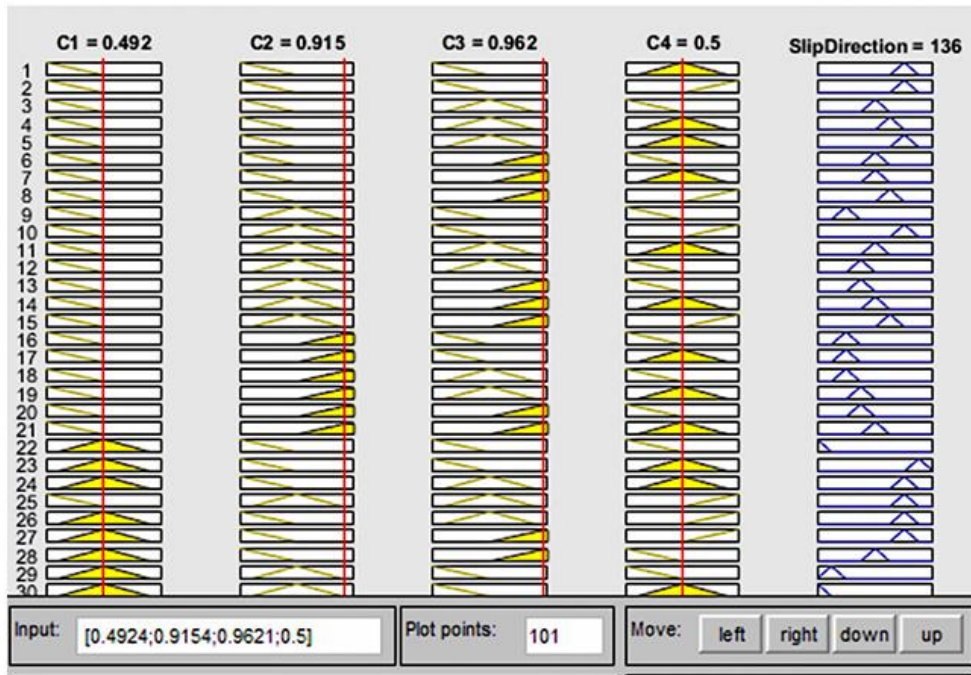


Figure 4. The fragment of the fuzzy rule is based for the identification of the slippage direction.

Each fuzzy rule in the rule base follows the general structure that defines the relationship between the input conditions and the resulting output:

IF (Condition–Antecedent), THEN (Result–Consequent).

This rule set determines the dependence between the slip displacement direction  $\alpha$  and the capacitive values  $C_1$ ,  $C_2$ ,  $C_3$ , and  $C_4$ , expressed as:

$$\alpha = f_{FS}(C_1, C_2, C_3, C_4)$$

In this case,  $C_1$ ,  $C_2$ ,  $C_3$ , and  $C_4$  stand for the output signals from the respective capacitive sensing components and are also used as input signals for the fuzzy system that was created. Figure 5 shows the characterized surfaces of the Mamdani-type fuzzy system that was designed for processing information from slip displacement sensors.

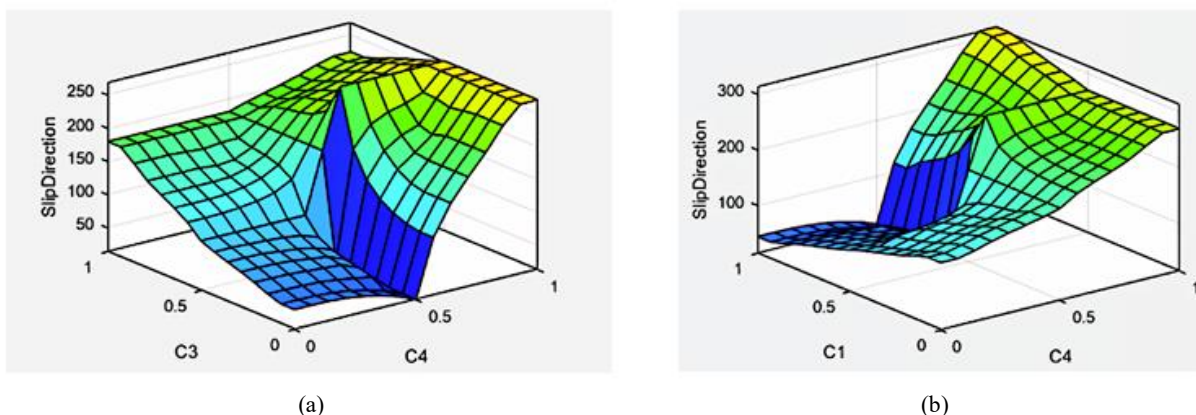


Figure 5. Characteristic surfaces of the fuzzy system  $\alpha = f_{FS}(C_1, C_2, C_3, C_4)$ : (a)  $C_1 = \text{const}$ ,  $C_2 = \text{const}$ ; (b)  $C_2 = \text{const}$ ,  $C_3 = \text{const}$

According to simulation results, the designed fuzzy system can accurately determine the slippage direction for any natural combination of the measured capacitive parameters  $C_1$ ,  $C_2$ ,  $C_3$ , and  $C_4$ , which correspond to the displacement of the sensitive element (rod 4 in Figure 1) and provides efficient sensor information processing. By using a fuzzy logic-based approach to handle sensor data, the adaptive robot's functional capabilities are improved and it can now adjust its trajectory to avoid obstacles. Additionally, a variety of structural and parametric optimization strategies for fuzzy systems can be used to enhance the accuracy and performance of the generated fuzzy system [107–112].

A) the fuzzy system's intelligent rule base and developed structure for processing sensor data during slip-displacement detection and identification of the object's slippage direction within the  $0^\circ$ – $360^\circ$  range, and (b) the suggested machine-learning algorithm and information-communication framework between the fuzzy system and the original capacitive multi-component slip-displacement sensor are what make the results presented novel.

#### 4. NEURO-FUZZY METHODS FOR MOBILE ROBOT CONTROL SYSTEMS ON VERTICAL, INCLINED, AND CEILING FERROMAGNETIC SURFACES

Mobile robots (MR) that can move across several surface orientations—horizontal, inclined, vertical, and even ceiling-mounted—are becoming increasingly necessary in contemporary applications. To operate in difficult conditions, these robots use a variety of propulsion and adhesion techniques. For example, they can automate labor-intensive shipbuilding procedures by cleaning the external surfaces of ships and other huge structures, whether they are afloat or in dry docks [98,113–115].

Cleaning large vertical or confined surfaces, decontaminating radioactive areas, installing dowels or explosive devices, fighting fires, painting, inspecting, diagnosing, and more are just a few of the complicated, resource-intensive, and potentially dangerous tasks that these robots are made to do. Ensuring dependable adherence (or gripping) to the surface and preserving stability without slippage during operation are crucial components of managing such robots [98]. Using electromagnets, clamping systems with magnetic fastening mechanisms provide efficient attachment. Using such systems on mobile robots, as shown in Figures 6 and 7, improves the efficiency and dependability of technological activities on ferromagnetic surfaces [15,17,98,116,117].

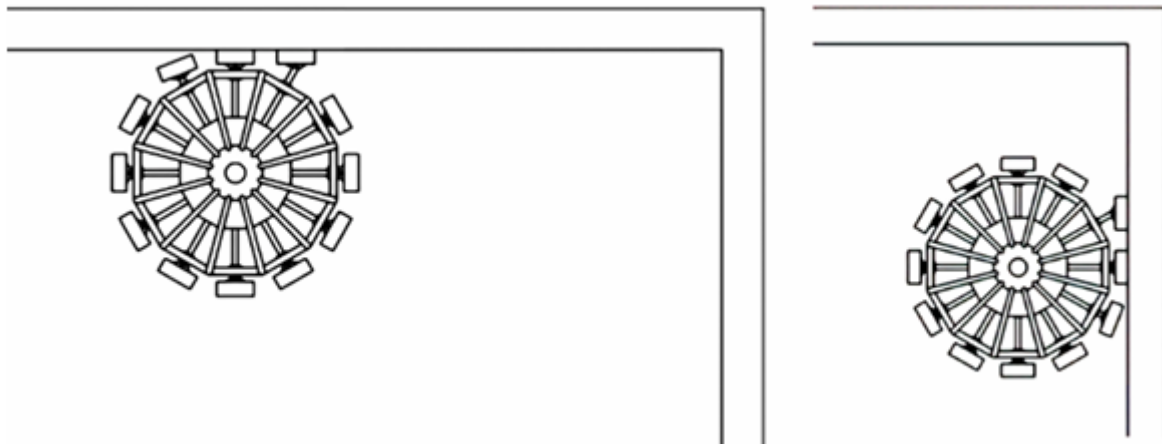
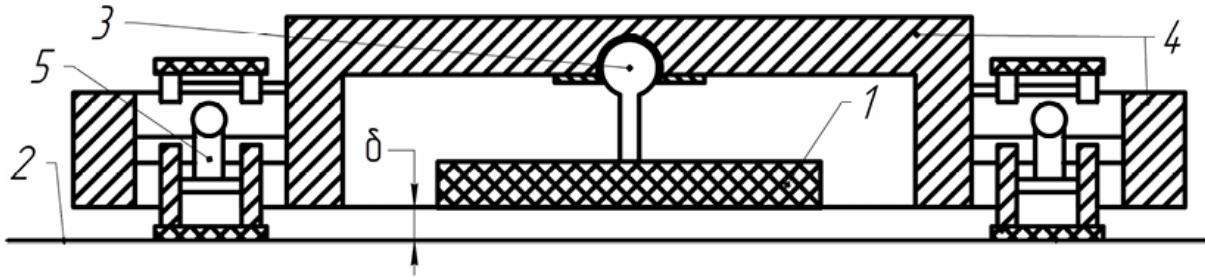


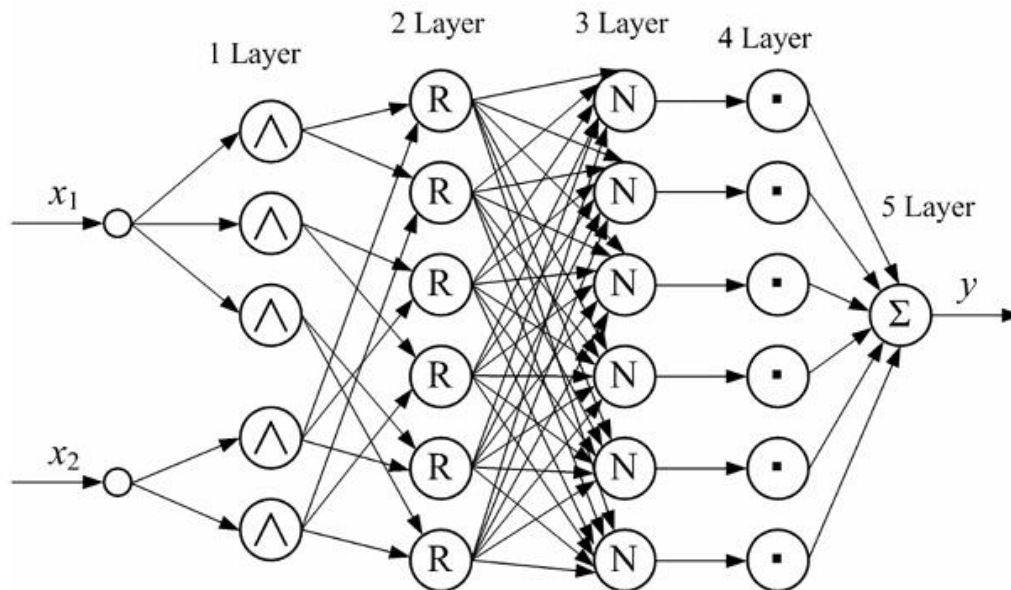
Figure 6 shows the movement on the ceiling (left) and vertical (right) electro-conductive surfaces in two intermediate stages of the magnet-controlled wheel of mobile robots with different stepping leg positions. Mobile robots with magnet-controlled wheels in two separate states—moving on vertical (right) and ceiling (left) electroconductive surfaces—have stepping legs positioned differently.





. Figure 7. Multipurpose caterpillar mobile robot (MR): 1 — main clamping magnet; 2 — ferromagnetic surface; 3 — spherical joint; 4 — frame; 5 — right and left tracks;  $\delta$  — clearance.

The clamping force observer [17,98] for the mobile robot (MR) shown in Figure 6 can be synthesized using machine learning techniques based on the adaptive neuro-fuzzy inference engine. An artificial neural network based on a fuzzy inference system (FIS) is called an adaptive network-based fuzzy inference system (ANFIS) [98]. ANFIS is set up as a five-layer neural network that propagates signals forward (Figure 8). A linguistic term (LT) of a certain input signal is represented by each node in the first ANFIS layer. In Figure 8, for instance, there are five nodes in the first layer since the first input signal contains two LTs (Small, Large) and the second input signal has three LTs (Small, Middle, Large). The number of fuzzy rules, degrees of antecedent realization, and contributions of the rules are represented by the six nodes found in the second, third, and fourth layers of ANFIS. The output signal is created in the fifth layer by combining their contributions in a single node.



**Figure 8.** Functional structure of typical ANFIS with two inputs  $x_1$ ,  $x_2$ , and one output  $y$ .

By monitoring the distance between the robot's clamping magnet and a non-stationary ferromagnetic surface coated with non-ferromagnetic components, ANFIS enables better sensor information processing (Figure 8). The accuracy of the clamping force observer and the ANFIS training procedure are both strongly impacted by the form of the linguistic term membership functions of the input variables.

The developed fuzzy-neuro observer's great accuracy in determining the necessary clamping force is demonstrated by comparative data. ANFIS's training error for input signal linguistic phrases employing Gaussian-2 membership functions is 0.187, which is 1.23 and 2.75 times lower than that of  $\pi$ -like and trapezoidal membership functions, respectively. Figure 9 displays the ANFIS characteristic surface using Gaussian-2 membership functions for the input variables.

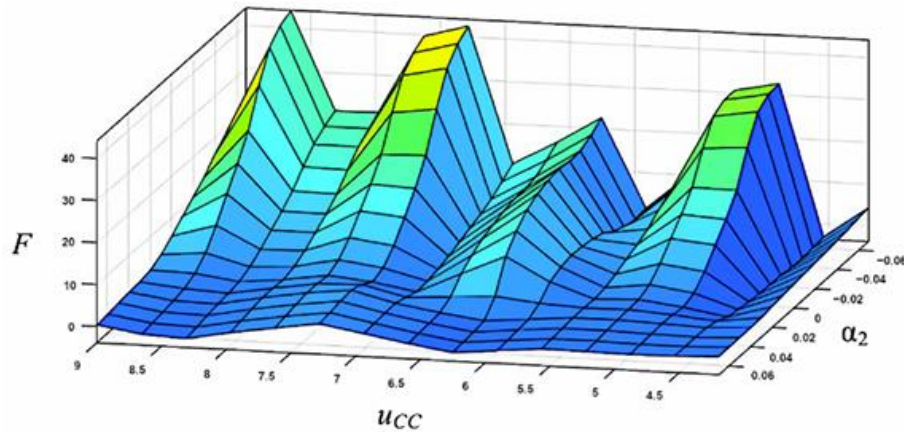


Figure 9. Characteristic surface based on Gaussian-2 membership functions. F

## 5. PREDICTIVE REGULATION OF ROBOTIC SENSORS AND CONTROL SYSTEMS UTILIZING CANONICAL DECOMPOSITION OF STATISTICAL DATA

Sensor system failures arise from the inherent degradation of the sensors and the idiosyncrasies of their operation [118–122]: Adverse operational environments (extreme temperatures, humidity, pressure, pollution, lighting, etc.); Work autonomy; Variations in the relative positioning of the sensor and the recognition object; Real-time operation (almost continuously); Resource constraints. Deficiencies in sensor systems markedly diminish the operational efficacy of robotic control systems, potentially resulting in catastrophic outcomes at critical infrastructure sites. The assessment of the control system's operability in real-time is a critical and pressing task, considering the nuances of varying operating conditions and their analysis using machine learning.

To guarantee the operational reliability of control systems, it is recommended to incorporate a predictive control module within the overall framework—this involves estimating the system's condition at future time intervals, followed by a determination of its applicability. In general, parameter  $S$ , which defines the quality of the system's performance (operational duration, frequency of operations per time unit, accuracy of execution, etc.), is stochastic. Consequently, to address the issue of estimating the system's future state, it is essential to employ the methodologies of random function and random sequence theory.

The canonical decomposition of a random sequence  $S = S(i)$ , where  $i = 1, I$ , of the variable parameter  $S$  at the time points  $t_i$ ,  $i = 1, I$ , is the most universal mathematical model in terms of constraints [120]:

The random sequence  $\{S\} = S(i)$ ,  $i = 1, 2, \dots, I$  am represented by the following canonical expansion:

$$S(i) = E[S(i)] + \sum_{v=1}^i \sum_{\lambda=1}^N .P_v^{(\lambda)} \rho(\lambda, v; 1, i), i = \overline{1, I} \quad (1)$$

where  $E[\ ]$  denotes the mathematical expectation.

Elements of the Expansion

The expansion is defined by two key elements:

- $P_v^{(s)}$ : Centered, uncorrelated random coefficients
- $\rho(\lambda, v; h, i)$ : Coordinate (weight) functions describing probabilistic relationships

1. Recurrence Relation for Random Coefficients:

$$P_v^{(\lambda)} = S^\lambda(v) - E[S^\lambda(v)] - \sum_{\mu=1}^{v-1} \sum_{j=1}^N \cdot P_\mu^{(j)} \rho(j, \mu; \lambda, v) - \sum_{j=1} \cdot P_v^{(j)} \rho(j, v; \lambda, v), \quad \begin{matrix} \lambda = 1, \bar{N}, \\ v = 1, \bar{I}. \end{matrix} \quad (2)$$

2. Coordinate Functions Definition:

$$\rho(\lambda, v; h, i) = \frac{E[P_v^{(\lambda)}(S^h(i) - E[S^h(i)])]}{E[(P_v^{(\lambda)})^2]} \quad (3a)$$

Recursive Computational Form:

$$\rho(\lambda, v; h, i) = \frac{1}{D_\lambda(v)} \left\{ \begin{aligned} & E[S^\lambda(v)S^h(i)] - E[S^\lambda(v)]E[S^h(i)] \\ & - \sum_{\mu=1}^{v-1} \sum_{j=1}^N \cdot D_j(\mu)\rho(j, \mu; \lambda, v)\rho(j, \mu; h, i) \\ & - \sum_{j=1} \cdot D_j(v)\rho(j, v; \lambda, v)\rho(j, v; h, i) \end{aligned} \right\}, \quad (3b)$$

$\lambda = 1, \bar{h}, v = 1, \bar{v}, h = 1, \bar{N}, i = 1, \bar{I}.$

Orthogonal Properties:

$$\rho(\lambda, v; h, i) = \begin{cases} 1, & \text{if } (h = \lambda) \wedge (v = i); \\ 0, & \text{for } (i < v) \vee ((h < \lambda) \wedge (v = i)). \end{cases} \quad (4)$$

3. Variance of Random Coefficients:

$$D_\lambda(v) = E[S^{2\lambda}(v)] - (E[S^\lambda(v)])^2 - \sum_{\mu=1}^{v-1} \sum_{j=1}^N \cdot D_j(\mu)(\rho(j, \mu; \lambda, v))^2 - \sum_{j=1} \cdot D_j(v)(\rho(j, v; \lambda, v))^2, \quad \begin{matrix} \lambda = 1, \bar{N}, \\ v = 1, \bar{I}. \end{matrix} \quad (5)$$

Prediction and Control Framework

Prediction Algorithm

The recursive estimation formula for future parameter estimates:

$$\hat{s}^{(\mu,l)}(h, i) = \begin{cases} E[S^h(i)], & \text{for } \mu = 0 \\ \hat{s}^{(\mu,l-1)}(h, i) + (s^l(\mu) - \hat{s}^{(\mu,l-1)}(l, \mu))\rho(l, \mu; h, i), & \text{for } l \neq 1 \\ \hat{s}^{(\mu-1,N)}(h, i) + (s^l(\mu) - \hat{s}^{(\mu-1,N)}(l, \mu))\rho(l, \mu; h, i), & \text{for } l = 1 \end{cases} \quad (6)$$

Coordinate Function Updates

$$T_\lambda^{(\alpha)}(\xi) = \begin{cases} T_\lambda^{(\alpha-1)}(\xi) - T_\lambda^{(\alpha-1)}(\alpha)\tau_\alpha(\xi), & \text{if } \lambda \leq \alpha - 1; \\ \gamma_\alpha(\xi), & \text{for } \lambda = \alpha; \end{cases} \quad (7)$$

$$\tau_\alpha(\xi) = \begin{cases} \rho(\text{mod}_N(\alpha), \lfloor \frac{\alpha}{N} \rfloor + 1; 1, \lfloor \frac{\alpha}{N} \rfloor + 1), & \text{for } \xi \leq kN; \\ \rho(\text{mod}_N(\alpha), \lfloor \frac{\alpha}{N} \rfloor + 1; 1, i), & \text{if } \xi = (i - 1)N + 1. \end{cases} \quad (8)$$

Mean Square Forecast Error

$$E[\Delta^2(k, N, i)] = \begin{aligned} & E[S^{2\lambda}(v)] - (E[S^\lambda(v)])^2 \\ & - \sum_{\mu=k+1}^i \cdot \sum_{\lambda=1}^N \cdot D_\lambda(\mu)\rho^2(\lambda, \mu; 1, i)\beta_{1\mu}^{(\lambda)}(i), \quad i = k + 1, I. \end{aligned} \quad (9)$$

Predictive Control Condition

$$a \leq \hat{s}(i) \leq b, i = k + 1, I. \quad (10)$$

The system's operational requirement is defined as:

$$u \simeq s(t) \simeq v \quad (11)$$

Adaptive Learning During Operation

Recursive Model Training

Mathematical Expectation Update:

$$E_{(L+1)}[S^n(i)] = \frac{E_{(L)}[S^n(i)] \cdot L + s_{L+1}^n(i)}{L + 1} \quad (12)$$

Variance Update:

$$D_{n,(L+1)}(v) = \frac{D_{n,(L)}(v) \cdot (L - 1) + (\bar{s}_{L+1}^n(v) - \bar{s}_{(L)}^{(n-1,v)}(n, v))^2}{L} \quad (13)$$

Coordinate Functions Update:

$$\rho_{(L+1)}(n, v; \lambda, i) = \frac{\rho_{(L)}(n, v; \lambda, i) \cdot (L - 1) + (\bar{s}_{L+1}^n(v) - \bar{s}_{(L)}^{(n-1,v)}(n, v))(s_{L+1}^\lambda(i) - E_{(L+1)}[S^\lambda(i)])}{L \cdot D_{n,(L+1)}(v)} \quad (14)$$

## Experimental Validation

The method was validated using the random sequence model:

$$S(i + 1) = \frac{5S(i)}{1 + S^2(i)} - 0.5S(i) - 0.5S(i - 1) + 0.5S(i - 2) + \xi(i + 1) \quad (15)$$

Initial conditions: random sequences uniformly distributed on  $[-1,1]$ , with  $\xi(i)$  uniformly distributed on  $[-0.1, 0.1]$ . Through 100 extrapolation experiments comparing linear algorithms, 4th-order Kalman filters, and the proposed 4th-order nonlinear algorithm, standard deviation estimates were obtained (Figure 10).

When using the nonlinear methods (5) and (6) (curve "non-linear forecast" in Figure 10), which take into account the stochastic properties of the investigated random sequences as much as possible (nonlinearity, use of the full amount of a posteriori information, non-stationarity), the standard deviation of the forecast error (Figure 10) shows a high forecast accuracy. Because non-linear relationships are used, the extrapolation accuracy is 3–3.4 times higher than that of the Wiener–Hopf technique [126] ("linear forecast" curve in Figure 10), and it is 1.5–2.4 times higher than that of the Kalman method [126] because a greater amount of a posteriori data is used.

The characteristics of the predictive control module's operation are depicted in the diagram in Figure 11. Equations (5), (6), (9), and (10), as well as data regarding the operation of systems of this class, are used to derive the estimates  $\hat{s}(i)$ ,  $i = k+1$ , of future values at the time of system deployment. Equations (13)–(15) are used in subsequent operations to perform machine learning of the model and the extrapolator based on statistical data about the system being studied.

Given that the forecast algorithm's initial parameters (5), (6), (9), and (10) can be determined beforehand before the system begins operating and that the training (13)–(15) and extrapolation (7) and (8) formulas are computationally straightforward, the module can operate in real-time.

Design of Control Systems and Simulation of Robot Arms. Accurate object recognition, image categorization, and high-quality machine learning all depend on the creation of an efficient control system. Arm modelling and control system design both heavily rely on machine learning and intelligent control techniques. It can be utilized for computer vision, remote control of hazardous and/or dangerous goods, sorting and orienting items on a conveyor, and automatic object recognition based on learnt models employing machine learning technologies [24,27,42,105]. Generally speaking, software implementation is imposed on the weak points of analogue goods [1,127]. Prior to software implementation, a control system must be designed and a manipulator's arm must be simulated. An example structure diagram of the control system is presented in Figure 12. It displays a list of various control system services' attributes and capabilities. These services will only be utilized in the appropriate locations within the system and won't be linked to the management system itself. For instance, to view the log of every activity the robot has taken, the network service might be required. Let's take a closer look at the pertinent elements.

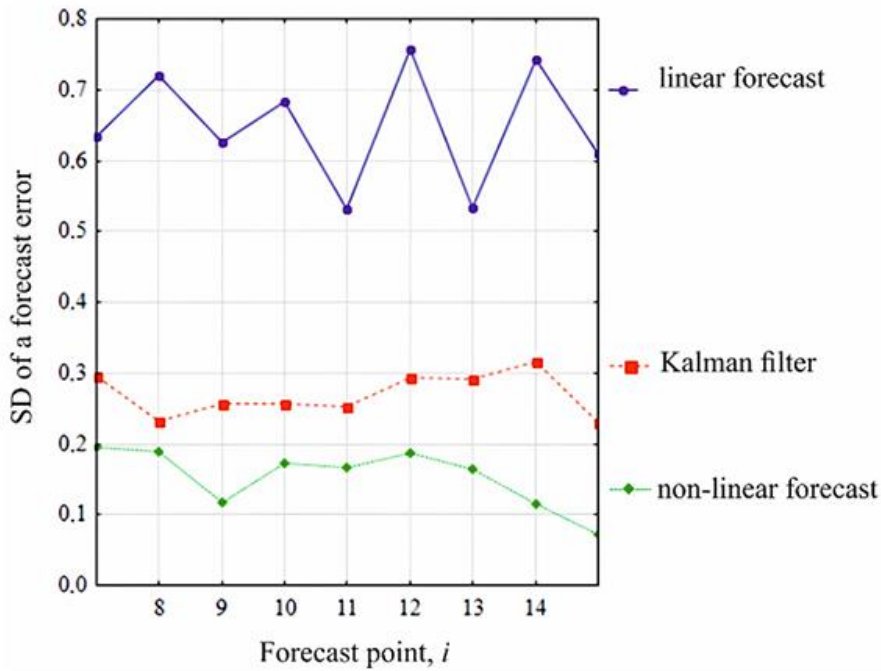


Figure 10. Forecast error standard deviation of the random sequence realizations for various extra potation algorithms: compar

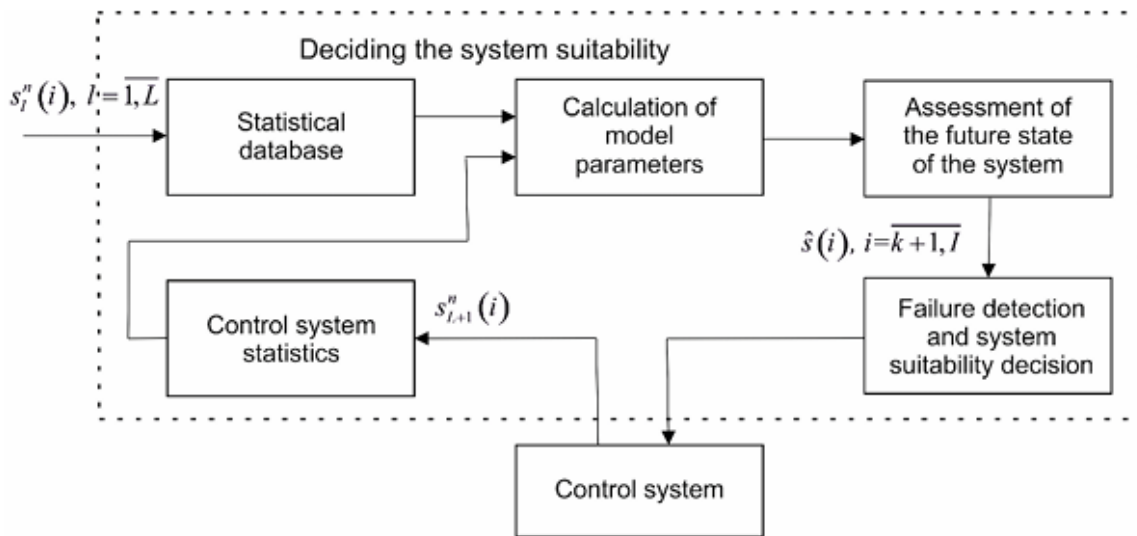


Figure11. Diagram of the predictive control module functioning

Preventing failures and, consequently, guaranteeing a control system's continuous operation in the future is a major benefit of the suggested approach for evaluating a control system's operability. The method's benefit is that it also considers the unique features of the control system: the process of operation assessment of operability based on current measurements of the system's state in real-time, which accumulates priori information about the system under study. In order to solve the problem of predictive operability monitoring as accurately as possible, the forecasting employed an algorithm that, in contrast to the known methods (Wiener–Hopf method, Kolmogorov polynomial, Kalman filter,

etc.), does not impose any restrictions on the random sequence of changes in system parameters (linearity, monotonicity, ergodicity, stationarity, Markov properties, etc.). The engineering solution of the approach under consideration is to forecast the technical state of the item

## 6 DESIGN OF CONTROL SYSTEMS AND SIMULATION OF ROBOT ARMS

Accurate object recognition, image categorization, and high-quality machine learning all depend on the creation of an efficient control system. Arm modelling and control system design both heavily rely on machine learning and intelligent control techniques. It can be utilized for computer vision, automatic object detection based on trained models employing machine learning technologies, sorting and orienting things on the conveyor, and remote control of hazardous and/or destructive objects [24,27,42,105].

Generally speaking, software implementation is imposed on the weak points of analogue goods [1,127]. Prior to software implementation, a control system must be designed, and a manipulator's arm must be simulated. An example structure diagram of the control system is presented in Figure 12. It displays a list of various control system services' attributes and functions. These services will only be utilized in the appropriate locations within the system and will not be allocated to the management system itself. For instance, the network service can be required to view the robot's action log. Let's examine the pertinent elements in greater detail.

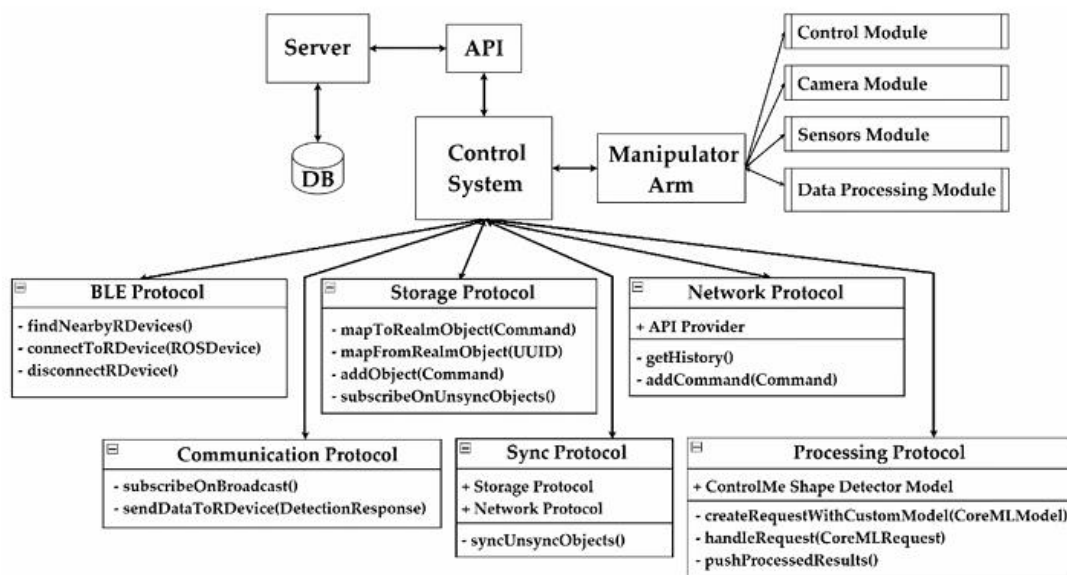


Figure 12. An extended structural diagram of the control system

### 6.1. "System of Control" Part

Some examples of general functions include: utilizing a Bluetooth connection to communicate with a robotic device; choosing the system operation mode (testing the model, working with the device); intelligently controlling the device with a trained neural network; storing information about the device's state and the task's current progress; synchronizing intermediate data with the server when the device completes the task; and communicating with the server via the REST (Representational State Transfer) API (Application Programming Interface). A BLE (Bluetooth Low Energy) service to connect to a robotic device, a service to save local data, a network service to communicate with the server, a communication service to communicate with a robotic device, and a data processing service with an artificial intelligence module make up the list of internal dependencies and extra services. To recognize and classify patterns, the artificial intelligence module is shown as a neural network. Convolutional neural networks with the YOLOv2 (You Only Look Once) structure make up this neural network's architecture. It consists of a fully connected neural network for classification, layers pooling for feature map construction, and sequential convolution layers with the function ReLU (Rectified Linear Unit) [127–130].

### 6.2. The "Server" Part

General capabilities include storing history of all device operating states, communicating with the control system via the REST API, and storing intermediate data. The list of internal requirements includes "SQLite," a relational database management system (DBMS) for data storage; "Fluent ORM" for transforming bespoke Swift programming language queries into raw SQL (Structured Query Language); and REST API with commands (GET, POST) [131].

### 6.3. The "Manipulator Arm" Part

General functions include using a Bluetooth connection to communicate with the control system, taking a picture of the surroundings with an integrated camera and sending it to the system, obtaining the object's coordinates from the system, and converting the coordinates that are obtained from the system into environmental coordinates. The ROS (Robot Operating System) real-time operating system and the MoveIt software are used to mimic a manipulator arm in the "Manipulator arm" component. It features a gripper at the end and five degrees of freedom [132–135]. The open-source manipulation program MoveIt (Figure 13) was created by Willow Garage by Sachin Chitta and Ioan A. Sucan [133–135]. The program provides answers to issues with mobile manipulation, including navigation, 3D perception, planning and motion control, and kinematics. Robotic systems frequently employ the MoveIt library, which is a component of the ROS package. Because it can be readily tailored for any task, MoveIt is excellent for developers [133–135].

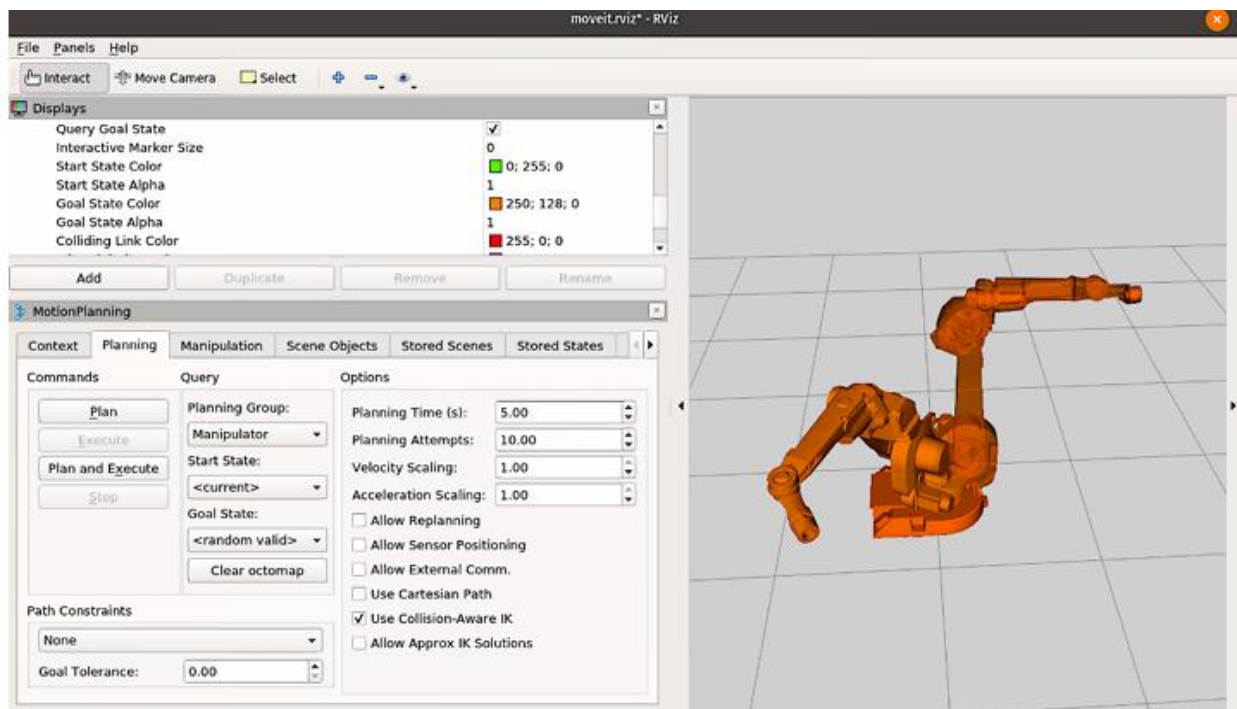


Figure 13. Motion planning with MoveIt software.

The URDF (Universal Robot Description Format) file of the device—in our case, the URDF of the manipulator's arm—must be obtained in order to construct a control system utilizing MoveIt software. Elements of arm composition, length, and joints are included in this file. MoveIt Assistant, which is included with MoveIt, greatly aids with configuration and simplification [133]. Movement planning, manipulation, and other operations can be carried out on a robotic device using the MoveIt Rviz (Figure 13) setup environment and the batch files containing the robot configurations [133–135].

The development of the original control system structure with the implementation of system component interaction protocols and the integration of an artificial intelligence module for object recognition and classification for manipulator arm missions constitute the novelty of the obtained results. (b) The developed control system is modeled in the MoveIt environment, with the corresponding parameters adjusted for various manipulator arm missions.



### 7. Using Convolutional Neural Networks for Object Recognition in Robotic Workspace

Selecting image sets to carry out the neural network's learning process is essential for creating an intelligent control module. Using artificial neural networks, one machine learning technique pattern recognition and classification performs intelligent control [1, 3, 24, 25, 26, 27, 42, 99]. The control system processes the image that is obtained and relays the processing results back to the robotic device when the robotic device sends the query to collect an image from the surroundings. The object's membership in one of the classes and its coordinates, in relation to the final image, are the outcome of this process. In our situation, three image datasets, the cube, cylinder, and sphere must be located for each class (see Figure 14 for partial image datasets).

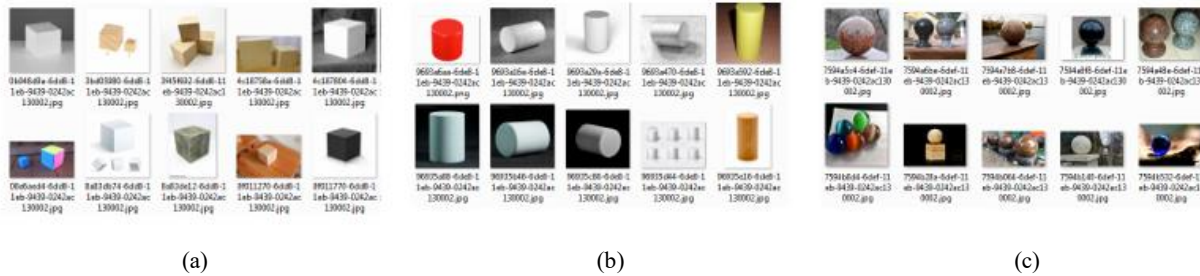


Figure 14. the datasets for: (a) cube; (b) cylinder; (c) sphere.

Images with one or more objects, such as three cubes and two cylinders or one to five spheres, were included in the dataset that was produced for this challenge. The dataset had 213 photographs in total: 70 images for the "Cube" class, 70 images for the "Cylinder" class, 71 images for the "Sphere" class, and 2 images with multiple classes. Annotations describing the coordinates of every object were necessary for the complete dataset because the neural network training procedure for this class of tasks is supervised. The authors made a project, uploaded the image sets, and annotated each class using the cloud.annotations.ai portal. Bounding rectangles were drawn around each instance of an object in an image as part of this procedure; for instance, 10 cubes in an image needed ten distinct annotations, each described by its coordinates (X, Y, width, and height). All the data was annotated before being exported in the Create ML format. This set of photos and the accompanying annotations in.json format, which provide comprehensive details about the placement of each class object in the 2D space, are necessary for training the convolutional neural network used in the study, which has a YOLOv2 architecture.

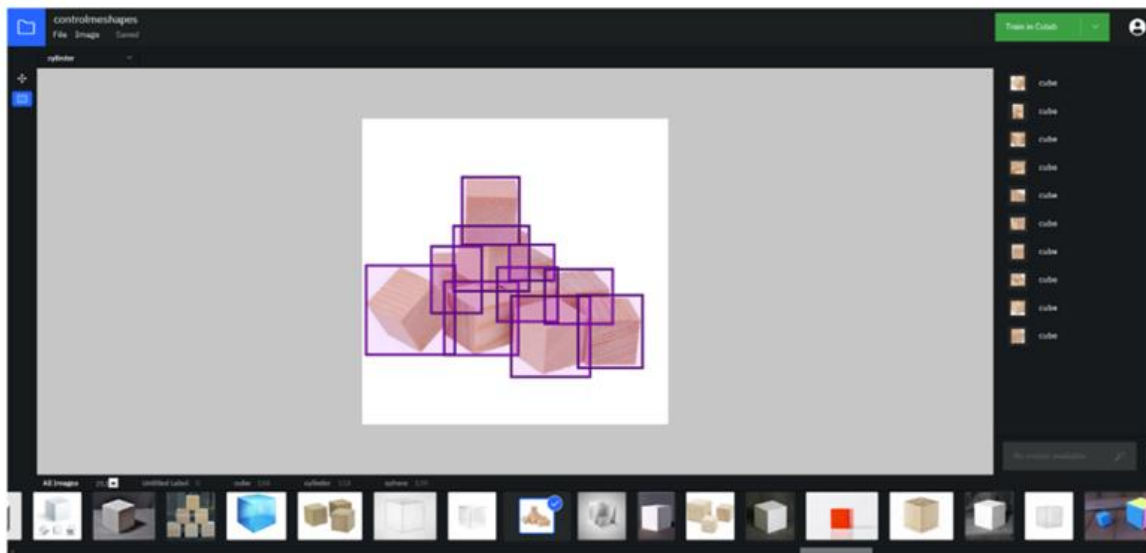


Figure15. The process of adding annotations to images

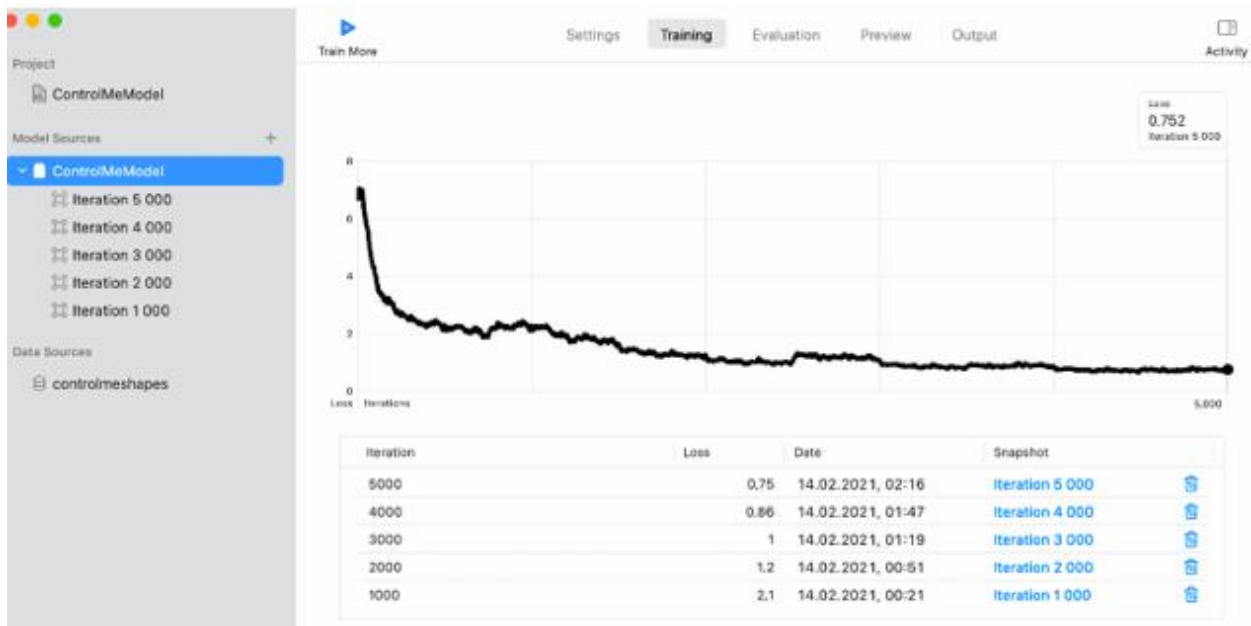


Figure 16. The training process in Create ML

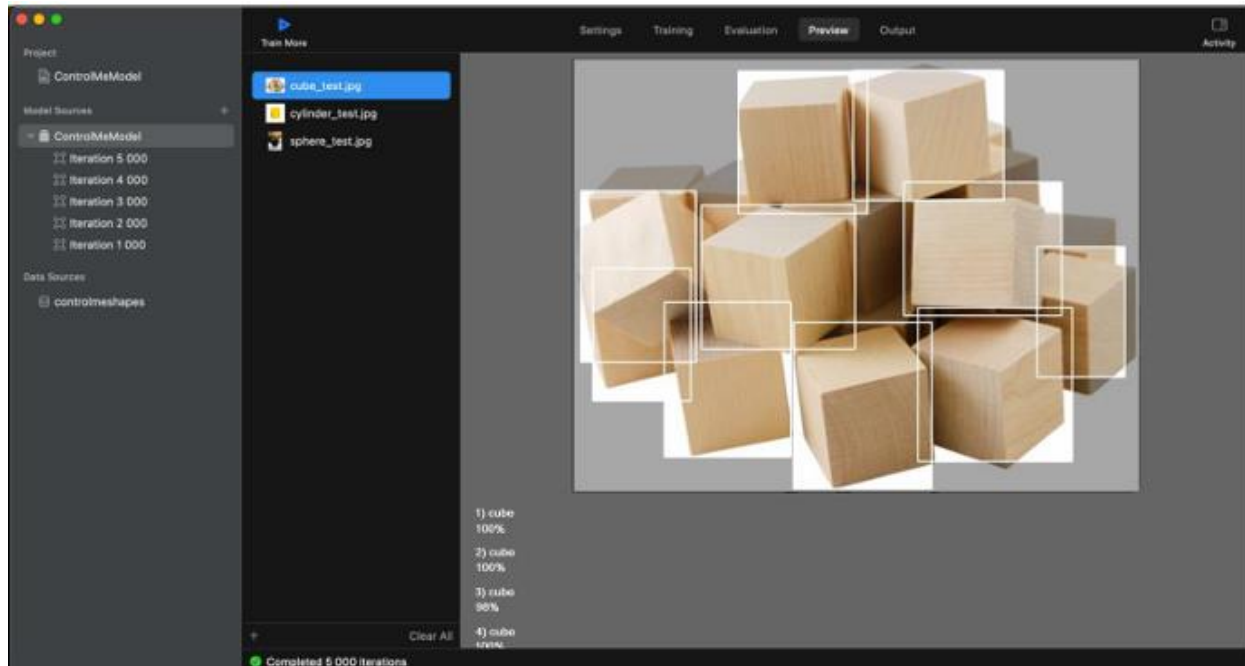
Additional information about learning (training) outcomes is in table 1

Table-1 Training outcomes

Number of Epochs	Loss	Time in Seconds	Training Accuracy in %	Validation Accuracy in %
1000	2.123	1860	87	82
2000	1.211	3660	95	90
3000	1.044	5340	95	93
4000	0.857	7020	99	93
5000	0.752	8760	100	95

After 5000 epochs, the neural network model's training accuracy was 100%. The 2016 MacBook Air with a 1.6 GHz i5 CPU was used to build, train, and test the model. We created a set of images that included objects from the designated classes that weren't in the training set of images to verify the neural network's training outcomes. According to the testing (identification) findings (Figure 17), the 10 cubes in the image were identified with 99.8%

accuracy (one cube was recognized with 98% accuracy, while the others were recognized with 100% accuracy). It should be mentioned, too, that occasionally the recognition accuracy decreases when the image is deformed. The overall accuracy was 99% across 50 photos with varying classes and items that were examined. The HTTP (HyperText Transfer Protocol) protocol serves as the foundation for client-server communication in the built software implementation. HTTP requests (GET, POST, PUT, and DELETE) provide communication. One of the possible endpoints receives a request from the client; the server receives it, processes the information, and responds [99].



**Figure 17.** Testing (recognition) results of the object class "Cube"

The authors have expanded the number of recognition classes from three ("Cube," "Sphere," and "Cylinder") to five ("Cube," "Sphere," "Cylinder," "Cone," and "Pyramid") to boost the intelligent control system's capabilities. 52 photos of the class "Sphere," 45 images of the class "Cone," 44 images of the class "Cube," 42 images of the class "Pyramid," and 42 images of the class "Cylinder" are among the 225 images that make up the final dataset. The resolution of the input photos was 416 by 416 pixels. Table 2 provides more details regarding the learning (training) outcomes for the five recognition classes. In this instance, a more potent 2017 MacBook Air with a 1.8 GHz i5 CPU was used to develop, train, and test the model. Based on data in Table 1, the neural network model trained with 100% accuracy in 5000 epochs with reduced time and loss. Furthermore, the recognition (classification) accuracy is really high. For instance, the network can identify identical items in different quantities that are members of the same class with 100% accuracy (Figure 18 a,b). When multiple objects of various classes are present in the same image, recognition accuracy declines (Figure 18c). For instance, with differing degrees of accuracy, the "Cylinder" object (Figure 18c) is a member of the following classes: 91% of the class "Cylinder," 4% of the class "Sphere," 3% of the class "Cube," and 1% of the classes "Pyramid" and "Cone."

Table 2 Training outcomes for recognition of five classes using Create ML

Number of Epochs	Loss	Time in Seconds	Training Accuracy in %	Validation Accuracy in %
450	1.7	780	85	80
510	1.7	900	86	81
1000	1.2	1800	87	82
2000	0.95	3540	95	90
3000	0.73	5220	95	93
4000	0.68	6960	99	96
5000	0.63	8700	100	100



(a)

(b)

(c)

**Figure 18.** Testing (recognition) results: (a) class “Pyramid”; (b) class “Sphere”; (c) several objects of different classes.

Five geometric figures (cone, cube, cylinder, sphere, and torus) from a similar dataset (<https://data.wielgosz.info>, accessed on December 10, 2021) were selected to examine the effects of sample size, neural network design, and learning parameters on recognition accuracy. There are test and training samples in the dataset. The training sample consists of 40,000 photos, with 8,000 images for each figure. Ten thousand photos make up the testing sample (two thousands of each object). For every sample, a different transformation was used. Normalization and cropping to 224 by 224 pixels were applied to both training and test photos. The training sample was also subjected to additional changes, such as random rotations and image center shifts, for increased accuracy. The ResNet34 architecture was selected by the authors. It was worked using the Torch library for Python. The GPU

was used to train the neural network at Google Colab [26,27]. The ResNet34 model was downloaded for training, and five outputs—each representing a figure class—were used instead of the default 1000. The procedure was halted in the third period because the testing accuracy declined (overfitting/overtraining occurred). 98% was the best result during the second epoch. Table 3 provides further details regarding training and testing results for five recognition classes. The accuracy of the tests was raised to 99.2% by the authors using fine-tuning technologies. Table 4 provides further details on training and testing results for five recognition classes with fine-tuning technologies.

**Table3.** Training and testing outcomes for recognition of 5 classes by Torch library (Python)

Number of Epochs	Training Loss	Testing Loss	Training Accuracy in %	Testing Accuracy in %
1	0.1187	0.0857	96.21	97.07
2	0.0599	0.0611	98.02	98.00
3	0.0515	0.0661	98.28	97.87

**Table 4.** Training and testing outcomes for recognition of 5 classes using fine-tuning technology by the Torch library (Python).

Number of Epochs	Training Loss	Testing Loss	Training Accuracy in %	Testing Accuracy in %
1	0.0469	0.0341	98.38	98.68
2	0.0092	0.0582	99.67	97.68
3	0.0060	0.0263	99.75	99.20
4	0.0058	0.0677	99.76	97.16

Overtraining occurs after the third epoch, according to the results (Table 4). With the training accuracy of 99.75% and the testing accuracy of 99.2%, let's concentrate on the third epoch. Comparing the acquired numerical findings to other similar research [68] that used a support vector machine algorithm and a K-nearest neighbor technique for recognition and classification, the accuracy values were 81.6% and 84.2%, respectively. This demonstrates how beneficial neural networks are for these kinds of issues.

The impact of dataset size on recognition accuracy was also examined by the authors. There were only 200 photos of each class in the training and test datasets for the study. So, there were still 1000 test images and 1000 training images. Table 5 displays the outcomes of using the neural network using the ResNet34 design and fine-tuning technology.

**Table 5.** Training and testing outcomes for object recognition

Number of Epochs	Training Loss	Testing Loss	Training Accuracy in %	Testing Accuracy in %
1	0.6128	0.3412	77.80	88.40
2	0.1623	0.1902	94.80	93.60
3	0.1330	0.2070	95.30	90.90
4	0.1011	0.1240	96.70	96.80
5	0.0302	0.0601	99.10	98.00

According to the findings (Table 5), the testing accuracy progressively rises from 88.4% to 98%. But compared to a whole dataset, the neural network trains more slowly. The number of epochs and the quality of training are significantly influenced by the sample size.

The server portion was implemented using the Vapor web framework. The overall list of server functions states that connection with the client portion is required. The HTTP protocol can be used for this. Working with Vapor requires the Vapor Toolbox. Use the following command in the Terminal window to download and install this set: `brew tap vapor/tap && brew install vapor/tap/vapor`. Once this command is run, the Vapor Toolbox is accessible [137].

To test the created application interface, use the Postman software. Postman makes it possible to confirm both the client's and the server's responses to queries. Next, draft a request for a command to be added to the scheme. Click send after choosing the POST method, entering the endpoint address and port, and creating the query body, which will include the model's attributes. Next, make a request to get all of the scheme's commands. Click send after choosing the GET method and entering the port and endpoint address. The list of received items and the successful query code are visible [131].

We can set up communication with the server with the aid of Moya, an abstract network layer. Set the HTTP method, request type, headers, endpoints, and server address to do this [126]. Modules "Dashboard," "Testing," and "Control" are the segments that make up the device control system (Figure 19) [99].

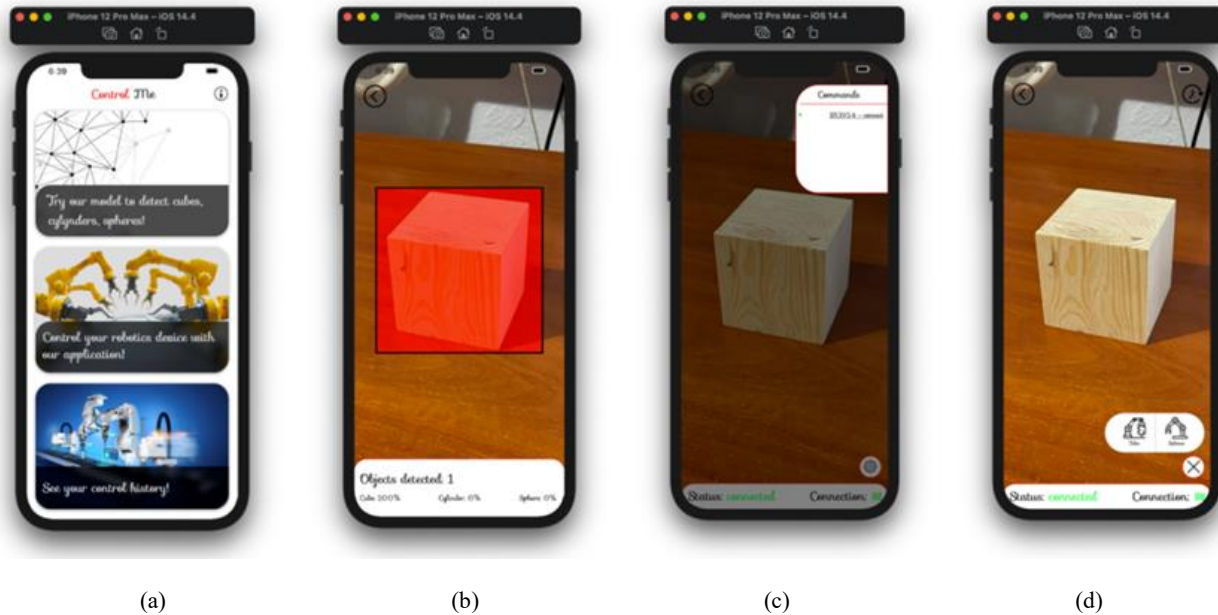


Figure 19. Device control system: (a) "Dashboard" module; (b) "Testing" module; (c) "Control" module: submodule "Device connection"; (d) "Control" module: submodule "Capture by device." Figure 19. System of Device Control: Modules "Dashboard" (a), "Testing" (b), "Control" (submodule "Deviceconnection"), and "Capturebydevice" (submodule "Control")

The "Dashboard" module (Figure 19a) is shown as a list of potential ways to use the system, including connecting to a robotic device and controlling it, viewing the history of activities on devices, and testing a trained model using a mobile device's camera (emulating a manipulator camera). A trained convolutional network using the YOLOv2 architecture and the ReLU activation function in the convolution layers is tested using the "Testing" module (Figure 19b). This module uses the designated network to process images and replicates the transmission of a video feed from a robotic device. The block with the number of items identified in the current image that belong to the designated class indicates the processing results. The essential component of this intelligent control system is the "Control" module (Figure 19c,d). Local data storage, network, communication, synchronization, and data processing are among the services it offers. The Swinject container is used to realize all of these services. The "Control" module requires the "Device connection" submodule (Figure 19c) in order to locate nearby robotic devices and establish a Bluetooth connection with them. Following a successful connection, a command containing the device's communication protocols is sent out. The purpose of Module "Control" and its submodule "Capture by device" (Figure 19d) is to directly capture and control the identified object.

Utilizing a Swinject container, the "History" module presents a set of things that were received from a network service. A drop-down list with potential sorting choices (by device ID, context, and timestamp) is included in the collection header. In contrast to the current model, which has three recognition classes (cube, cylinder, sphere, and pyramid), the new results include (a) expanding the capabilities of the intelligent control system by increasing the number of recognition classes to five classes (pyramid, cone, cube, and sphere); (b) further developing the neural network model with the ResNet34 architecture through the complex application of optimization techniques, such as random rotations and shifts of the image center, splitting the training sample into batches with 64 images, and using fine-tuning technology, which, in comparison to the existing model, provides increasing the training accuracy by 1.73% (from 98.02% to 99.75%) and the testing accuracy by 1.2% (from 98.0% to 99.2%).

## 8.CONCLUSION

With an emphasis on robotic systems, this article offers an analytical survey of machine learning techniques used in several spheres of human endeavor. In sophisticated multi-component robotic complexes operating in unpredictable, unknown, or non-stationary working environments, particular focus is given to improving the processing efficiency of sensor and control inputs. In addition to the sensor and control systems, MCRC is made up of a mobile robot that moves and an adaptive robot that has a fixed base (a manipulator with an adaptive gripper).

By using (a) fuzzy logic to identify the direction of slippage of a manipulated object in the robot fingers when they collide with an obstacle, (b) neuro and neuro-fuzzy approaches to design intelligent controllers and clamping force observers of mobile robots with magnetically controlled wheels that can move working tools or an adaptive robot with a fixed base on inclined or ceiling ferromagnetic surfaces (ship hull, etc.), and (c) a canonical decomposition approach from statistical learning theory to predict robot control system states during robot mission in the nonstationary environment, the authors' contributions show how to improve control quality and extend the functioning properties of MCRC's robotic components.

An operability monitoring module that enables the identification of potential system faults at later times is suggested to increase the dependability of control systems. The employed algorithm for forecasting control system parameters does not impose any constraints on the random sequence of changes in system parameters (linearity, monotonicity, ergodicity, stationarity, Markov properties, etc.), in contrast to the well-known methods (Wiener–Hopf method, Kolmogorov polynomial, Kalman filter, etc.). This allows for the highest level of accuracy in resolving the predictive operability monitoring problem. Its great efficiency was validated by the numerical experiment's results (the relative extrapolation error is 2–3%).

Additionally, using video-sensor information processing, the authors demonstrate the effectiveness of software used to train the proposed convolutional neural network and design robot control systems for object detection from various classes. During the system design phase, a general structural schematic of the control system is created, an entity-table with a relational database management system is displayed, and MoveIt software is used to construct a robotic arm. After the design was finished, the control system's software implementation phase was completed. The Vapor web framework is used to develop the server, the CreateML framework is used to configure and create a neural network, and Swift and other technologies are used to implement the control system. Compared to Create ML (YOLOv2 architecture), a neural network with the ResNet34 architecture trains faster (3 epochs to attain 99.2% testing accuracy). The authors' study of the effect of dataset size on training and testing accuracy yielded the following results: (a) training accuracy increased gradually from the first to the fifth epoch (77.8%, 94.8%, 95.3%, 96.7%, and 99.1%, respectively); (b) testing accuracy increased from the first (88.4%) to the fifth epoch (98.0%); and (c) the neural network trained more slowly (122 s in 5 epochs) on the small size dataset (1000 training images) than on the full dataset (75 s in 3 epochs), while using 40,000 training images. Using the Torch library (Python), fine-tuning technology with the ResNet34 architecture improves training accuracy to 99.75 percent and testing accuracy to 99.2% for the recognition of five distinct classes.

## 9. REFERENCES

- [1] Kawana, E., & Yasunobu, S. (2007). An intelligent control system using object model by real-time learning. In Proceedings of the SICE Annual Conference (pp. 2792–2797). IEEE. <https://doi.org/10.1109/SICE.2007.4421416>
- [2] Ayvaz, S., & Alpay, K. (2021). Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. *Expert Systems with Applications*, 173, 114598. <https://doi.org/10.1016/j.eswa.2021.114598>
- [3] Kondratenko, Y., & Duro, R. (Eds.). (2015). *Advances in intelligent robotics and collaborative automation*. River Publishers.
- [4] Kondratenko, Y., Khalaf, P., Richter, H., & Simon, D. (2019). Fuzzy real-time multi-objective optimization of a prosthesis test robot control system. In *Advanced control techniques in complex engineering systems: Theory and applications* (pp. 165–185). Springer. [https://doi.org/10.1007/978-3-030-21927-7\\_8](https://doi.org/10.1007/978-3-030-21927-7_8)
- [5] Guo, J., Xian, B., Wang, F., & Zhang, X. (2013). Development of a three degree-of-freedom testbed for an unmanned helicopter and attitude control design. In Proceedings of the 32nd Chinese Control Conference (pp. 733–738). IEEE.
- [6] Kondratenko, Y., Klymenko, L., Kondratenko, V., Kondratenko, G., & Shvets, E. (2013). Slip displacement sensors for intelligent robots: Solutions and models. In Proceedings of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS) (pp. 861–866). IEEE. <https://doi.org/10.1109/IDAACS.2013.6663022>
- [7] Derkach, M., Matiuk, D., & Skarga-Bandurova, I. (2020). Obstacle avoidance algorithm for small autonomous mobile robot equipped with ultrasonic sensors. In Proceedings of the 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT) (pp. 236–241). IEEE. <https://doi.org/10.1109/DESSERT50317.2020.9125027>
- [8] Tkachenko, A. N., Brovinskaya, N. M., & Kondratenko, Y. P. (1983). Evolutionary adaptation of control processes in robots operating in non-stationary environments. *Mechanism and Machine Theory*, 18(4), 275–278. [https://doi.org/10.1016/0094-114X\(83\)90104-8](https://doi.org/10.1016/0094-114X(83)90104-8)
- [9] Kondratenko, Y., Khademi, G., Azimi, V., Ebeigbe, D., Abdelhady, M., Fakoorian, S. A., Barto, T., Roshaneshat, A., Atamanyuk, I., & Simon, D. (2016). Robotics and prosthetics at Cleveland State University: Modern information, communication, and modeling technologies. In *Communications in Computer and Information Science: Proceedings of the 12th International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications* (Vol. 783, pp. 133–155). Springer. [https://doi.org/10.1007/978-3-319-45916-5\\_9](https://doi.org/10.1007/978-3-319-45916-5_9)



- [10] Patel, U., Hatay, E., D'Arcy, M., Zand, G., & Fazli, P. (2017). A collaborative autonomous mobile service robot. AAAI Fall Symposium on Artificial Intelligence for Human-Robot Interaction (AI-HRI).
- [11] Li, Z., & Huang, Z. (2013). Design of a type of cleaning robot with ultrasonic. *Journal of Theoretical and Applied Information Technology*, 47(1), 1218–1222.
- [12] Kondratenko, Y. P. (2015). Robotics, automation and information systems: Future perspectives and correlation with culture, sport and life science. In A. M. Gil-Lafuente, C. Zopounidis, J. M. Merigó, & J. D. G. Guerrero (Eds.), *Decision making and knowledge decision support systems* (Vol. 675, pp. 43–56). Springer. [https://doi.org/10.1007/978-3-319-03907-7\\_6](https://doi.org/10.1007/978-3-319-03907-7_6)
- [13] Taranov, M. O., & Kondratenko, Y. (2018). Models of robot's wheel-mover behavior on ferromagnetic surfaces. *International Journal of Computing*, 17(1), 8–14.
- [14] D'Arcy, M., Fazli, P., & Simon, D. (2017). Safe navigation in dynamic, unknown, continuous, and cluttered environments. In *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE. <https://doi.org/10.1109/SSRR.2017.8088139>
- [15] Taranov, M., Wolf, C., Rudolph, J., & Kondratenko, Y. (2017). Simulation of robot's wheel-mover on ferromagnetic surfaces. In *Proceedings of the 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)* (pp. 283–288). IEEE. <https://doi.org/10.1109/IDAACS.2017.8095085>
- [16] Driankov, D., & Saffiotti, A. (Eds.). (2013). *Fuzzy logic techniques for autonomous vehicle navigation*. Physica-Verlag.
- [17] Kondratenko, Y. P., Rudolph, J., Kozlov, O. V., Zaporozhets, Y. M., & Gerasin, O. S. (2017). Neuro-fuzzy observers of clamping force for magnetically operated movers of mobile robots. *Technical Electrodynamics*, 5, 53–61. <https://doi.org/10.15407/techned2017.05.053>
- [18] Gerasin, O. S., Topalov, A. M., Taranov, M. O., Kozlov, O. V., & Kondratenko, Y. P. (2020). Remote IoT-based control system of the mobile caterpillar robot. In *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer (ICTERI 2020)* (Vol. 2740, pp. 129–136). CEUR-WS.
- [19] Inoue, K., Kaizu, Y., Igarashi, S., & Imou, K. (2019). The development of autonomous navigation and obstacle avoidance for a robotic mower using machine vision technique. *\*IFAC-PapersOnLine*, 52\*(30), 173–177. <https://doi.org/10.1016/j.ifacol.2019.12.515>
- [20] Zhang, Y., Zhao, J., & Sun, J. (2020). Robot path planning method based on deep reinforcement learning. In *Proceedings of the 2020 IEEE 3rd International Conference on Computer and Communication Engineering Technology (CCET)* (pp. 49–53). IEEE. <https://doi.org/10.1109/CCET50901.2020.9213107>
- [21] Qijie, Z., Yue, Z., & Shihui, L. (2020). A path planning algorithm based on RRT and SARSA( $\lambda$ ) in unknown and complex conditions. In *Proceedings of the 2020 Chinese Control and Decision Conference (CCDC)* (pp. 2035–2040). IEEE. <https://doi.org/10.1109/CCDC49329.2020.9164702>
- [22] Koza, J. R., Bennett, F. H., Andre, D., & Keane, M. A. (1996). Automated design of both the topology and sizing of analog electrical circuits using genetic programming. In J. S. Gero & F. Sudweeks (Eds.), *Artificial intelligence in design '96* (pp. 151–170). Springer. [https://doi.org/10.1007/978-94-009-0279-4\\_9](https://doi.org/10.1007/978-94-009-0279-4_9)
- [23] Hu, J., Niu, H., Carrasco, J., Lennox, B., & Arvin, F. (2020). Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 69(12), 14413–14423. <https://doi.org/10.1109/TVT.2020.3034800>
- [24] Sokoliuk, A., Kondratenko, G., Sidenko, I., Kondratenko, Y., Khomchenko, A., & Atamanyuk, I. (2020). Machine learning algorithms for binary classification of liver disease. In *Proceedings of the 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T)* (pp. 417–421). IEEE. <https://doi.org/10.1109/PICST51311.2020.9467908>
- [25] Sheremet, A., Kondratenko, Y., Sidenko, I., & Kondratenko, G. (2021). Diagnosis of lung disease based on medical images using artificial neural networks. In *Proceedings of the 2021 IEEE 3rd Conference on Electrical and Computer Engineering (UKRCON)* (pp. 561–566). IEEE. <https://doi.org/10.1109/UKRCON53503.2021.9575552>
- [26] Kondratenko, Y., Sidenko, I., Kondratenko, G., Petrovych, V., Taranov, M., & Sova, I. (2020). Artificial neural networks for recognition of brain tumors on MRI images. In *Communications in Computer and Information Science: Proceedings of the 16th International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications (ICTERI 2020)* (Vol. 1308, pp. 119–140). Springer. [https://doi.org/10.1007/978-3-030-77592-6\\_6](https://doi.org/10.1007/978-3-030-77592-6_6)
- [27] Sova, I., Sidenko, I., & Kondratenko, Y. (2020). Machine learning technology for neoplasm segmentation on brain MRI scans. In *\*Proceedings of the 2020 PhD Symposium at ICT in Education, Research, and Industrial Applications (ICTERI-PhD 2020)\** (Vol. 2791, pp. 50–59). CEUR-WS.
- [28] Kassahun, Y., Yu, B., Tibebu, A. T., Stoyanov, D., Giannarou, S., Metzen, J. H., & Poorten, E. V. (2016). Surgical robotics beyond enhanced dexterity instrumentation: A survey of machine learning techniques and their role in intelligent and autonomous surgical actions. *International Journal of Computer Assisted Radiology and Surgery*, 11(4), 553–568. <https://doi.org/10.1007/s11548-015-1305-z>
- [29] Striuk, O., & Kondratenko, Y. (2021). Generative adversarial neural networks and deep learning: Successful cases and advanced approaches. *International Journal of Computing*, 20(3), 339–349. <https://doi.org/10.47839/ijc.20.3.2279>

- [30] Mohammadi, A., Meniaïlov, A., Bazilevych, K., Yakovlev, S., & Chumachenko, D. (2021). Comparative study of linear regression and SIR models of COVID-19 propagation in Ukraine before vaccination. *Radioelectronic and Computer Systems*, 3, 5–14. <https://doi.org/10.32620/reks.2021.3.01>
- [31] Striuk, O., Kondratenko, Y., Sidenko, I., & Vorobyova, A. (2020). Generative adversarial neural network for creating photorealistic images. In *Proceedings of the 2nd IEEE International Conference on Advanced Trends in Information Theory (ATIT)* (pp. 368–371). IEEE. <https://doi.org/10.1109/ATIT50783.2020.9349323>
- [32] Ren, C., Kim, D.-K., & Jeong, D. (2020). A survey of deep learning in agriculture: Techniques and their applications. *Journal of Information Processing Systems*, 16(5), 1015–1033. <https://doi.org/10.3745/JIPS.04.0194>
- [33] Kumar, M., Kumar, A., & Palaparthi, V. S. (2021). Soil sensors-based prediction system for plant diseases using exploratory data analysis and machine learning. *IEEE Sensors Journal*, 21(16), 17455–17468. <https://doi.org/10.1109/JSEN.2021.3084319>
- [34] Atamanyuk, I., Kondratenko, Y., & Sirenko, N. (2018). Management system for agricultural enterprise on the basis of its economic state forecasting. In C. Berger-Vachon, A. M. Gil Lafuente, J. Kacprzyk, Y. Kondratenko, J. Merigó, & C. F. Morabito (Eds.), *Complex systems: Solutions and challenges in economics, management and engineering* (Vol. 125, pp. 453–470). Springer. [https://doi.org/10.1007/978-3-319-69989-9\\_27](https://doi.org/10.1007/978-3-319-69989-9_27)
- [35] Atamanyuk, I., Kondratenko, Y., Poltorak, A., Sirenko, N., Shebanin, V., Baryshevska, I., & Atamaniuk, V. (2019). Forecasting of cereal crop harvest on the basis of an extrapolation canonical model of a vector random sequence. In *Proceedings of the 15th International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications. Volume II: Workshops (ICTERI 2019)* (Vol. 2393, pp. 302–315). CEUR-WS.
- [36] Atamanyuk, I. P., Kondratenko, Y. P., & Sirenko, N. N. (2016). Forecasting economic indices of agricultural enterprises based on vector polynomial canonical expansion of random sequences. In *Proceedings of the 12th International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications (ICTERI 2016)* (Vol. 1614, pp. 458–468). CEUR-WS.
- [37] Wemers, B., & Kondratenko, Y. (2018). Alternative fuzzy approaches for efficiently solving the capacitated vehicle routing problem in conditions of uncertain demands. In C. Berger-Vachon, A. M. Gil Lafuente, J. Kacprzyk, Y. Kondratenko, J. Merigó, & C. F. Morabito (Eds.), *Complex systems: Solutions and challenges in economics, management and engineering* (Vol. 125, pp. 521–543). Springer. [https://doi.org/10.1007/978-3-319-69989-9\\_31](https://doi.org/10.1007/978-3-319-69989-9_31)
- [38] Kondratenko, G. V., Kondratenko, Y. P., & Romanov, D. O. (2006). Fuzzy models for capacitive vehicle routing problem in uncertainty. In *Proceedings of the 17th International DAAAM Symposium Intelligent Manufacturing and Automation: Focus on Mechatronics & Robotics* (pp. 205–206). DAAAM International.
- [39] Zinchenko, V., Kondratenko, G., Sidenko, I., & Kondratenko, Y. (2020). Computer vision in control and optimization of road traffic. In *Proceedings of the 2020 IEEE 3rd International Conference on Data Stream Mining and Processing (DSMP)* (pp. 249–254). IEEE. <https://doi.org/10.1109/DSMP47368.2020.9204193>
- [40] Jingyao, W., Manas, R. P., & Nallappan, G. (2022). Machine learning-based human-robot interaction in ITS. *Information Processing & Management*, 59(2), 102750. <https://doi.org/10.1016/j.ipm.2021.102750>
- [41] Leizerovych, R., Kondratenko, G., Sidenko, I., & Kondratenko, Y. (2020). IoT-complex for monitoring and analysis of motor highway condition using artificial neural networks. In *Proceedings of the 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)* (pp. 207–212). IEEE. <https://doi.org/10.1109/DESSERT50317.2020.9125035>
- [42] Kondratenko, Y. P., & Kondratenko, N. Y. (2016). Reduced library of the soft computing analytic models for arithmetic operations with asymmetrical fuzzy numbers. In A. Casey (Ed.), *Soft computing: Developments, methods and applications* (pp. 1–38). NOVA Science Publishers.
- [43] Gozhyj, A., Nechakhin, V., & Kalinina, I. (2020). Solar power control system based on machine learning methods. In *Proceedings of the 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT)* (pp. 24–27). IEEE. <https://doi.org/10.1109/CSIT49958.2020.9322053>
- [44] Chornovol, O., Kondratenko, G., Sidenko, I., & Kondratenko, Y. (2020). Intelligent forecasting system for NPP's energy production. In *Proceedings of the 2020 IEEE 3rd International Conference on Data Stream Mining and Processing (DSMP)* (pp. 102–107). IEEE. <https://doi.org/10.1109/DSMP47368.2020.9204194>
- [45] Borysenko, V., Kondratenko, G., Sidenko, I., & Kondratenko, Y. (2020). Intelligent forecasting in multi-criteria decision-making. In *\*Proceedings of the 3rd International Workshop on Computer Modeling and Intelligent Systems (CMIS-2020)\** (Vol. 2608, pp. 966–979). CEUR-WS.
- [46] Lavrynenko, S., Kondratenko, G., Sidenko, I., & Kondratenko, Y. (2020). Fuzzy logic approach for evaluating the effectiveness of investment projects. In *Proceedings of the 2020 IEEE 15th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)* (pp. 297–300). IEEE. <https://doi.org/10.1109/CSIT49958.2020.9321905>
- [47] Bidyuk, P. I., Gozhyj, A., Kalinina, I., Vysotska, V., Vasilev, M., & Malets, M. (2020). Forecasting nonlinear nonstationary processes in machine learning task. In *Proceedings of the 2020 IEEE 3rd International Conference on Data Stream Mining and Processing (DSMP)* (pp. 28–32). IEEE. <https://doi.org/10.1109/DSMP47368.2020.9204182>

- [48] Osborne, M. A., Roberts, S. J., Rogers, A., Ramchurn, S. D., & Jennings, N. R. (2008). Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In Proceedings of the 2008 International Conference on Information Processing in Sensor Networks (IPSN 2008) (pp. 109–120). IEEE. <https://doi.org/10.1109/IPSN.2008.47>
- [49] Atamanyuk, I., Shebanin, V., Kondratenko, Y., Havrysh, V., Lykhach, V., & Kramarenko, S. (2020). Identification of the optimal parameters for forecasting the state of technical objects based on the canonical random sequence decomposition. In Proceedings of the 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT) (pp. 259–264). IEEE. <https://doi.org/10.1109/DESSERT50317.2020.9125032>
- [50] Li, X., Shang, W., & Cong, S. (2020). Model-based reinforcement learning for robot control. In Proceedings of the 2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM) (pp. 300–305). IEEE. <https://doi.org/10.1109/ICARM49381.2020.9195299>
- [51] Alsamhi, S. H., Ma, O., & Ansari, S. (2020). Convergence of machine learning and robotics communication in collaborative assembly: Mobility, connectivity and future perspectives. *Journal of Intelligent & Robotic Systems*, 98(3), 541–566. <https://doi.org/10.1007/s10846-019-01079-x>
- [52] Samadi, G. M., & Jond, H. B. (2021). Speed control for leader-follower robot formation using fuzzy system and supervised machine learning. *Sensors*, 21(10), 3433. <https://doi.org/10.3390/s21103433>
- [53] Krishnan, S., Wan, Z., Bharadwaj, K., Whatmough, P., Faust, A., Neuman, S., Wei, G.-Y., Brooks, D., & Reddi, V. J. (2021). Machine learning-based automated design space exploration for autonomous aerial robots. *arXiv*. <https://arxiv.org/abs/2102.02988v1>
- [54] El-Shamouty, M., Kleeberger, K., Lämmler, A., & Huber, M. F. (2019). Simulation-driven machine learning for robotics and automation. *Technisches Messen*, 86(12), 673–684. <https://doi.org/10.1515/teme-2019-0068>
- [55] Rajawat, A. S., Rawat, R., Barhanpurkar, K., Shaw, R. N., & Ghosh, A. (2021). Robotic process automation with increasing productivity and improving product quality using artificial intelligence and machine learning. In R. N. Shaw, A. Ghosh, & V. E. Balas (Eds.), *Artificial intelligence and future generation robotics* (pp. 1–13). De Gruyter. <https://doi.org/10.1515/9783110691320-001>
- [56] Yaseerz, A., & Chen, H. (2021). Machine learning based layer roughness modeling in robotic additive manufacturing. *Journal of Manufacturing Processes*, 70, 543–552. <https://doi.org/10.1016/j.jmpro.2021.09.010>
- [57] Wang, X. V., Pinter, J. S., Liu, Z., & Wang, L. (2021). A machine learning-based image processing approach for robotic assembly system. *Procedia CIRP*, 104, 906–911. <https://doi.org/10.1016/j.procir.2021.11.152>
- [58] Mayr, A., Kißkalt, D., Lomakin, A., Graichen, K., & Franke, J. (2021). Towards an intelligent linear winding process through sensor integration and machine learning techniques. *Procedia CIRP*, 96, 80–85. <https://doi.org/10.1016/j.procir.2021.01.071>
- [59] Al-Mousawi, A. J. (2020). Magnetic explosives detection system (MEDS) based on wireless sensor network and machine learning. *Measurement*, 151, 107112. <https://doi.org/10.1016/j.measurement.2019.107112>
- [60] Martins, P., Sá, F., Morgado, F., & Cunha, C. (2020). Using machine learning for cognitive Robotic Process Automation (RPA). In Proceedings of the 2020 15th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1–6). IEEE. <https://doi.org/10.23919/CISTI49556.2020.9141040>
- [61] Segreto, T., & Teti, R. (2019). Machine learning for in-process end-point detection in robot-assisted polishing using multiple sensor monitoring. *The International Journal of Advanced Manufacturing Technology*, 103(9), 4173–4187. <https://doi.org/10.1007/s00170-019-03822-y>
- [62] Klingspor, V., Morik, K., & Rieger, A. (1996). Learning concepts from sensor data of a mobile robot. *Machine Learning*, 23(2), 305–332. <https://doi.org/10.1007/BF00117446>
- [63] Zheng, Y., Song, Q., Liu, J., Song, Q., & Yue, Q. (2020). Research on motion pattern recognition of exoskeleton robot based on multimodal machine learning model. *Neural Computing and Applications*, 32(13), 1869–1877. <https://doi.org/10.1007/s00521-019-04525-x>
- [64] Radouan, A. M. (2021). Deep learning for robotics. *Journal of Data Analysis and Information Processing*, 9(3), 63–76. <https://doi.org/10.4236/jdaip.2021.93005>
- [65] Teng, X., & Lijun, T. (2021). Adoption of machine learning algorithm-based intelligent basketball training robot in athlete injury prevention. *Frontiers in Neurorobotics*, 14, 117. <https://doi.org/10.3389/fnbot.2020.597690>
- [66] Shih, B., Shah, D., Li, J., Thuruthel, T. G., Park, Y.-L., Iida, F., Bao, Z., Kramer-Bottiglio, R., & Tolley, M. T. (2020). Electronic skins and machine learning for intelligent soft robots. *Science Robotics*, 5(41), eaaz9239. <https://doi.org/10.1126/scirobotics.aaz9239>
- [67] Ibrahim, A., Younes, H., Alameh, A., & Valle, M. (2020). Near sensors computation based on embedded machine learning for electronic skin. *Procedia Manufacturing*, 52, 295–300. <https://doi.org/10.1016/j.promfg.2020.11.050>
- [68] Keser, S., & Hayber, Ş. E. (2021). Fiber optic tactile sensor for surface roughness recognition by machine learning algorithms. *Sensors and Actuators A: Physical*, 332, 113071. <https://doi.org/10.1016/j.sna.2021.113071>
- [69] Gonzalez, R., Fiacchini, M., & Iagnemma, K. (2018). Slippage prediction for off-road mobile robots via machine learning regression and proprioceptive sensing. *Robotics and Autonomous Systems*, 105, 85–93. <https://doi.org/10.1016/j.robot.2018.03.007>

- [70] Wei, P., & Wang, B. (2020). Multi-sensor detection and control network technology based on parallel computing model in robot target detection and recognition. *Computer Communications*, 159, 215–221. <https://doi.org/10.1016/j.comcom.2020.05.017>
- [71] Martinez-Hernandez, U., Rubio-Solis, A., & Prescott, T. J. (2020). Learning from sensory predictions for autonomous and adaptive exploration of object shape with a tactile robot. *Neurocomputing*, 382, 127–139. <https://doi.org/10.1016/j.neucom.2019.11.066>
- [72] Scholl, C., Tobola, A., Ludwig, K., Zanca, D., & Eskofier, B. M. (2021). A smart capacitive sensor skin with embedded data quality indication for enhanced safety in human–robot interaction. *Sensors*, 21(21), 7210. <https://doi.org/10.3390/s21217210>
- [73] Joshi, S., Kumra, S., & Sahin, F. (2020). Robotic grasping using deep reinforcement learning. In *Proceedings of the 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)* (pp. 1461–1466). IEEE. <https://doi.org/10.1109/CASE48305.2020.9216712>
- [74] Bilal, D. K., Unel, M., Tunc, L. T., & Gonul, B. (2022). Development of a vision based pose estimation system for robotic machining and improving its accuracy using LSTM neural networks and sparse regression. *\*Robotics and Computer-Integrated Manufacturing*, 74\*, 102262. <https://doi.org/10.1016/j.rcim.2021.102262>
- [75] Mishra, S., & Jabin, S. (2021). Recent trends in pedestrian detection for robotic vision using deep learning techniques. In R. N. Shaw, A. Ghosh, V. E. Balas, & M. Bianchini (Eds.), *Artificial intelligence for future generation robotics* (pp. 137–157). Elsevier. <https://doi.org/10.1016/B978-0-323-85498-6.00008-3>
- [76] Long, J., Mou, J., Zhang, L., Zhang, S., & Li, C. (2021). Attitude data-based deep hybrid learning architecture for intelligent fault diagnosis of multi-joint industrial robots. *Journal of Manufacturing Systems*, 61, 736–745. <https://doi.org/10.1016/j.jmsy.2021.01.007>
- [77] Subha, T. D., Subash, T. D., Claudia Jane, K. S., Devadharshini, D., & Francis, D. I. (2020). Autonomous under water vehicle based on extreme learning machine for sensor fault diagnostics. *Materials Today: Proceedings*, 24, 2394–2402. <https://doi.org/10.1016/j.matpr.2020.03.772>
- [78] Severo de Souza, P. S., Rubin, F. P., Hohemberger, R., Ferreto, T. C., Lorenzon, A. F., Luizelli, M. C., & Rossi, F. D. (2020). Detecting abnormal sensors via machine learning: An IoT farming WSN-based architecture case study. *Measurement*, 164, 108042. <https://doi.org/10.1016/j.measurement.2020.108042>
- [79] Kamizono, K., Ikeda, K., Kitajima, H., Yasuda, S., & Tanaka, T. (2021). FDC based on neural network with harmonic sensor to prevent error of robot. *IEEE Transactions on Semiconductor Manufacturing*, 34(3), 291–295. <https://doi.org/10.1109/TSM.2021.3087485>
- [80] Bidyuk, P., Kalinina, I., & Gozhyj, A. (2021). An approach to identifying and filling data gaps in machine learning procedures. In S. Babichev & V. Lytvynenko (Eds.), *Lecture notes in computational intelligence and decision making* (Vol. 77, pp. 164–176). Springer. [https://doi.org/10.1007/978-3-030-82014-5\\_11](https://doi.org/10.1007/978-3-030-82014-5_11)
- [81] Kondratenko, Y., & Kondratenko, N. (2018). Real-time fuzzy data processing based on a computational library of analytic models. *Data*, 3(4), 59. <https://doi.org/10.3390/data3040059>
- [82] Kondratenko, Y., & Gordienko, E. (2012). Implementation of the neural networks for adaptive control system on FPGA. In *Proceedings of the 23rd International DAAAM Symposium "Intelligent Manufacturing & Automation"* (Vol. 23, pp. 389–392). DAAAM International.
- [83] Choi, W., Heo, J., & Ahn, C. (2021). Development of road surface detection algorithm using CycleGAN-augmented dataset. *Sensors*, 21(24), 7769. <https://doi.org/10.3390/s21247769>
- [84] Kondratenko, Y. P., Kuntsevich, V. M., Chikrii, A. A., & Gubarev, V. F. (Eds.). (2021). *Advanced control systems: Theory and applications*. River Publishers.
- [85] Kuntsevich, V. M., Gubarev, V. F., Kondratenko, Y. P., Lebedev, D., & Lysenko, V. (Eds.). (2018). *Control systems: Theory and applications*. River Publishers.
- [86] Gerasin, O., Kondratenko, Y., & Topalov, A. (2018). Dependable robot's slip displacement sensors based on capacitive registration elements. In *Proceedings of the IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)* (pp. 378–383). IEEE. <https://doi.org/10.1109/DESSERT.2018.8409160>
- [87] Kondratenko, Y., Gerasin, O., & Topalov, A. (2016). A simulation model for robot's slip displacement sensors. *International Journal of Computing*, 15(4), 224–236.
- [88] Kondratenko, Y. P., Gerasin, O. S., & Topalov, A. M. (2015). Modern sensing systems of intelligent robots based on multi-component slip displacement sensors. In *Proceedings of the 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)* (pp. 902–907). IEEE. <https://doi.org/10.1109/IDAACS.2015.7340830>
- [89] Kondratenko, Y. P., & Kondratenko, V. Y. (2015). Advanced trends in design of slip displacement sensors for intelligent robots. In Y. Kondratenko & R. Duro (Eds.), *Advances in intelligent robotics and collaboration automation* (pp. 167–191). River Publishers.
- [90] Zaporozhets, Y. M., Kondratenko, Y. P., & Shyshkin, O. S. (2012). Mathematical model of slip displacement sensor with registration of transversal constituents of magnetic field of sensing element. *Technical Electrodynamics*, 4, 67–72.
- [91] Kondratenko, Y., Shvets, E., & Shyshkin, O. (2007). Modern sensor systems of intelligent robots based on the slip displacement signal detection. In *Proceedings of the 18th International DAAAM Symposium "Intelligent Manufacturing & Automation"* (pp. 381–382). DAAAM International.

- [92] Kondratenko, Y. P. (1993). Measurement methods for slip displacement signal registration. In Proceedings of the Second International Symposium on Measurement Technology and Intelligent Instruments (Vol. 2101, pp. 1451–1461). SPIE. <https://doi.org/10.1117/12.147293>
- [93] Massalim, Y., Kappasov, Z., Vargas Terres, V., Oliveira, A. S., & Varol, H. A. (2021). Robust detection of absence of slip in robot hands and feet. *Sensors*, 21(21), 7789. <https://doi.org/10.3390/s21237789>
- [94] Kondratenko, Y., Gerasin, O., Kozlov, O., Topalov, A., & Kilimanov, B. (2021). Inspection mobile robot's control system with remote IoT-based data transmission. *Journal of Mobile Multimedia*, 17(4), 499–522. <https://doi.org/10.13052/jmm1550-4646.1741>
- [95] Kondratenko, Y., Zaporozhets, Y., Rudolph, J., Gerasin, O., Topalov, A., & Kozlov, O. (2018). Modeling of clamping magnets interaction with ferromagnetic surface for wheel mobile robots. *International Journal of Computing*, 17(1), 33–46.
- [96] Kondratenko, Y. Y., Zaporozhets, Y., Rudolph, J., Gerasin, O., Topalov, A., & Kozlov, O. (2017). Features of clamping electromagnets using in wheel mobile robots and modeling of their interaction with ferromagnetic plate. In Proceedings of the 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS) (pp. 453–458). IEEE. <https://doi.org/10.1109/IDAACS.2017.8095089>
- [97] Taranov, M., Rudolph, J., Wolf, C., Kondratenko, Y., & Gerasin, O. (2017). Advanced approaches to reduce number of actors in a magnetically-operated wheel-mover of a mobile robot. In Proceedings of the 2017 13th International Conference Perspective Technologies and Methods in MEMS Design (MEMSTECH) (pp. 96–100). IEEE. <https://doi.org/10.1109/MEMSTECH.2017.7937541>
- [98] Kondratenko, Y. P., Kozlov, O. V., Gerasin, O. S., & Zaporozhets, Y. M. (2016). Synthesis and research of neuro-fuzzy observer of clamping force for mobile robot automatic control system. In Proceedings of the 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP) (pp. 90–95). IEEE. <https://doi.org/10.1109/DSMP.2016.7583524>
- [99] Kondratenko, Y., Sichevskiy, S., Kondratenko, G., & Sidenko, I. (2021). Manipulator's control system with application of the machine learning. In Proceedings of the 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS) (pp. 363–368). IEEE. <https://doi.org/10.1109/IDAACS53288.2021.9660999>
- [100] Ueda, M., Iwata, K., & Shingu, H. (1972). Tactile sensors for an industrial robot to detect a slip. In Proceedings of the 2nd International Symposium on Industrial Robots (pp. 63–76).
- [101] Ueda, M., & Iwata, K. (1973). Adaptive grasping operation of an industrial robot. In Proceedings of the 3rd International Symposium on Industrial Robots (pp. 301–310).
- [102] Tiwana, M. I., Shashank, A., Redmond, S. J., & Lovell, N. H. (2011). Characterization of a capacitive tactile shear sensor for application in robotic and upper limb prostheses. *Sensors and Actuators A: Physical*, 165(2), 164–172. <https://doi.org/10.1016/j.sna.2010.08.019>
- [103] Kondratenko, Y. P., Kondratenko, V. Y., & Shvets, E. A. (2016). Intelligent slip displacement sensors in robotics. In S. Y. Yurish (Ed.), *Sensors, transducers, signal conditioning and wireless sensors networks* (Vol. 3, pp. 37–66). IFSA Publishing.
- [104] Sheng, Q., Xu, G. Y., & Liu, G. (2014). Design of PZT micro-displacement acquisition system. *Sensors & Transducers*, 182(11), 119–124.
- [105] Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338–353. [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
- [106] Mamdani, E. H. (1974). Application of fuzzy algorithms for control of a simple dynamic plant. *Proceedings of the Institution of Electrical Engineers*, 121(12), 1585–1588. <https://doi.org/10.1049/piee.1974.0328>
- [107] Kondratenko, Y. P., & Simon, D. (2018). Structural and parametric optimization of fuzzy control and decision making systems. In L. A. Zadeh, R. R. Yager, S. N. Shahbazova, M. Z. Reformat, & V. Kreinovich (Eds.), *Recent developments and the new direction in soft-computing foundations and applications* (Vol. 361, pp. 273–289). Springer. [https://doi.org/10.1007/978-3-319-75408-6\\_22](https://doi.org/10.1007/978-3-319-75408-6_22)
- [108] Kondratenko, Y. P., Klymenko, L. P., & Al Zu'bi, E. Y. M. (2013). Structural optimization of fuzzy systems' rules base and aggregation models. *Kybernetes*, 42(5), 831–843. <https://doi.org/10.1108/K-03-2013-0055>
- [109] Kondratenko, Y. P., Altameem, T. A., & Al Zubi, E. Y. M. (2010). The optimisation of digital controllers for fuzzy systems design. *Advances in Modelling and Analysis A*, 47(1), 19–29.
- [110] Kondratenko, Y. P., & Al Zubi, E. Y. M. (2009). The optimisation approach for increasing efficiency of digital fuzzy controllers. In Proceedings of the Annals of DAAAM for 2009 & Proceedings of the 20th International DAAAM Symposium "Intelligent Manufacturing and Automation" (pp. 1589–1591). DAAAM International.
- [111] Kondratenko, Y. P., & Kozlov, A. V. (2019). Parametric optimization of fuzzy control systems based on hybrid particle swarm algorithms with elite strategy. *Journal of Automation and Information Sciences*, 51(4), 25–45. <https://doi.org/10.1615/JAutomatInfScien.v51.i4.30>
- [112] Pedrycz, W., Li, K., & Reformat, M. (2015). Evolutionary reduction of fuzzy rule-based models. In D. E. Tamir, N. D. Rische, & A. Kandel (Eds.), *Fifty years of fuzzy logic and its applications* (Vol. 326, pp. 459–481). Springer. [https://doi.org/10.1007/978-3-319-19683-1\\_23](https://doi.org/10.1007/978-3-319-19683-1_23)
- [113] Christensen, L., Fischer, N., Kroffke, S., Lemburg, J., & Ahlers, R. (2011). Cost-effective autonomous robots for ballast water tank inspection. *Journal of Ship Production and Design*, 27(3), 127–136. <https://doi.org/10.5957/jspd.2011.27.3.127>

- [114] Souto, D., Faiña, A., Lypez-Peca, F., & Duro, R. J. (2013). Lappa: A new type of robot for underwater non-magnetic and complex hull cleaning. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (pp. 3394–3399). IEEE. <https://doi.org/10.1109/ICRA.2013.6631057>
- [115] Ross, B., Bares, J., & Fromme, C. (2003). A semi-autonomous robot for stripping paint from large vessels. *The International Journal of Robotics Research*, 22(7-8), 617–626. <https://doi.org/10.1177/02783649030227015>
- [116] Kondratenko, Y., Kozlov, O., & Gerasin, O. (2019). Neuroevolutionary approach to control of complex multicoordinate interrelated plants. *International Journal of Computing*, 18(4), 502–514. <https://doi.org/10.47839/ijc.18.4.1699>
- [117] Gerasin, O., Kozlov, O., Kondratenko, G., Rudolph, J., & Kondratenko, Y. (2019). Neural controller for mobile multipurpose caterpillar robot. In Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS) (pp. 222–227). IEEE. <https://doi.org/10.1109/IDAACS.2019.8924365>
- [118] Michael, S. (2021). Metrological characterization and comparison of D415, D455, L515 RealSense devices in the close range. *Sensors*, 21(22), 7770. <https://doi.org/10.3390/s21227770>
- [119] Palar, P. S., Vargas Terres, V., & Oliveira, A. S. (2020). Human–robot interface for embedding sliding adjustable autonomy methods. *Sensors*, 20(21), 5960. <https://doi.org/10.3390/s20215960>
- [120] Atamanyuk, I. P., Kondratenko, V. Y., Kozlov, O. V., & Kondratenko, Y. P. (2012). The algorithm of optimal polynomial extrapolation of random processes. In Lecture notes in business information processing: Proceedings of the International Conference Modeling and Simulation in Engineering, Economics and Management (Vol. 115, pp. 78–87). Springer. [https://doi.org/10.1007/978-3-642-30433-0\\_9](https://doi.org/10.1007/978-3-642-30433-0_9)
- [121] Ryguła, A. (2021). Influence of trajectory and dynamics of vehicle motion on signal patterns in the WIM system. *Sensors*, 21(23), 7895. <https://doi.org/10.3390/s21237895>
- [122] Khan, F., Ahmad, S., Gürüler, H., Cetin, G., Whangbo, T., & Kim, C.-G. (2021). An efficient and reliable algorithm for wireless sensor network. *Sensors*, 21(24), 8355. <https://doi.org/10.3390/s21248355>
- [123] Yu, H., Chen, C., Lu, N., Lu, N., & Wang, C. (2021). Deep auto-encoder and deep forest-assisted failure prognosis for dynamic predictive maintenance scheduling. *Sensors*, 21(24), 8373. <https://doi.org/10.3390/s21248373>
- [124] Atamanyuk, I., Kondratenko, Y., Shebanin, V., & Mirgorod, V. (2015). Method of polynomial predictive control of fail-safe operation of technical systems. In Proceedings of the XIIIth International Conference "The Experience of Designing and Application of CAD Systems in Microelectronics" (CADSM) (pp. 248–251). IEEE. <https://doi.org/10.1109/CADSM.2015.7230833>
- [125] Atamanyuk, I., Shebanin, V., Kondratenko, Y., Volosyuk, Y., Sheptylevskiy, O., & Atamaniuk, V. (2019). Predictive control of electrical equipment reliability on the basis of the non-linear canonical model of a vector random sequence. In Proceedings of the IEEE International Conference on Modern Electrical and Energy Systems (MEES) (pp. 130–133). IEEE. <https://doi.org/10.1109/MEES.2019.8896493>
- [126] Everitt, B. S. (2006). *The Cambridge dictionary of statistics* (3rd ed.). Cambridge University Press.
- [127] Nagrath, I. J., Shripal, P. P., & Chand, A. (1995). Development and implementation of intelligent control strategy for robotic manipulator. In Proceedings of the IEEE/IAS International Conference on Industrial Automation and Control (pp. 215–220). IEEE. <https://doi.org/10.1109/IACC.1995.466448>
- [128] Alagöz, Y., Karabayır, O., & Mustaoğlu, A. F. (2020). Target classification using YOLOv2 in land-based marine surveillance radar. In Proceedings of the 28th Signal Processing and Communications Applications Conference (SIU) (pp. 1–4). IEEE. <https://doi.org/10.1109/SIU49456.2020.9302479>
- [129] Zhang, H., Zhang, L., Li, P., & Gu, D. (2018). Yarn-dyed fabric defect detection with YOLOV2 based on deep convolution neural networks. In Proceedings of the 7th Data Driven Control and Learning Systems Conference (DDCLS) (pp. 170–174). IEEE. <https://doi.org/10.1109/DDCLS.2018.8516097>
- [130] Wang, M., Liu, M., Zhang, F., Lei, G., Guo, J., & Wang, L. (2018). Fast classification and detection of fish images with YOLOv2. In Proceedings of the OCEANS - MTS/IEEE Kobe Techno-Oceans Conference (OTO) (pp. 1–4). IEEE. <https://doi.org/10.1109/OCEANSKOB.2018.8559274>
- [131] Li, L., Chou, W., Zhou, W., & Lou, M. (2016). Design patterns and extensibility of REST API for networking applications. *IEEE Transactions on Network and Service Management*, 13(1), 154–167. <https://doi.org/10.1109/TNSM.2016.2516946>
- [132] Rivera, S., Iannillo, A. K., Lagraa, S., Joly, C., & State, R. (2020). ROS-FM: Fast monitoring for the Robotic Operating System (ROS). In Proceedings of the 25th International Conference on Engineering of Complex Computer Systems (ICECCS) (pp. 187–196). IEEE. <https://doi.org/10.1109/ICECCS51672.2020.00027>
- [133] Gömer, M., Haschke, R., Ritter, H., & Zhang, J. (2019). MoveIt! Task constructor for task-level motion planning. In Proceedings of the International Conference on Robotics and Automation (ICRA) (pp. 190–196). IEEE. <https://doi.org/10.1109/ICRA.2019.8793510>
- [134] Deng, H., Xiong, J., & Xia, Z. (2017). Mobile manipulation task simulation using ROS with MoveIt. In Proceedings of the IEEE International Conference on Real-time Computing and Robotics (RCAR) (pp. 612–616). IEEE. <https://doi.org/10.1109/RCAR.2017.8311916>

- [135] Youakim, D., Ridaoui, P., Palomeras, N., Spadafora, F., Ribas, D., & Muzzupappa, M. (2017). MoveIt!: Autonomous underwater free-floating manipulation. *IEEE Robotics & Automation Magazine*, 24(3), 41–51. <https://doi.org/10.1109/MRA.2016.2636359>
- [136] Salameen, L., Estatieh, A., Darbisi, S., Tutunji, T. A., & Rawashdeh, N. A. (2020). Interfacing computing platforms for dynamic control and identification of an industrial KUKA robot arm. In *Proceedings of the 21st International Conference on Research and Education in Mechatronics (REM)* (pp. 1–5). IEEE. <https://doi.org/10.1109/REM49740.2020.9313883>
- [137] Gughani, S., Lu, X., & Panda, D. K. (2017). Swift-X: Accelerating OpenStack Swift with RDMA for building an efficient HPC cloud. In *\*Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)\** (pp. 238–247). IEEE. <https://doi.org/10.1109/CCGRID.2017.45>