Visual Intelligence

REVIEW

Open Access

Modular design automation of the morphologies, controllers, and vision systems for intelligent robots: a survey

Wenji Li[1,2], Zhaojun Wang[1,2], Ruitao Mai[1,2], Pengxiang Ren[1,2], Qinchang Zhang[1,2], Yutao Zhou[1,2], Ning Xu[1,2], JiaFan Zhuang[1,2], Bin Xin[3], Liang Gao[4], Zhifeng Hao[5] and Zhun Fan[1,2*]

Abstract

Design automation is a core technology in industrial design software and an important branch of knowledge-worker automation. For example, electronic design automation (EDA) has played an important role in both academia and industry. Design automation for intelligent robots refers to the construction of unified modular graph models for the morphologies (body), controllers (brain), and vision systems (eye) of intelligent robots under digital twin architectures, which effectively supports the automation of the morphology, controller, and vision system design processes of intelligent robots by taking advantage of the powerful capabilities of genetic programming, evolutionary computation, deep learning, reinforcement learning, and causal reasoning in model representation, optimization, perception, decision making, and reasoning. Compared with traditional design methods, MOdular DEsigN Automation (MODENA) methods can significantly improve the design efficiency and performance of robots, effectively avoiding the repetitive trial-and-error processes of traditional design methods, and promoting automatic discovery of innovative designs. Thus, it is of considerable research significance to study MODENA methods for intelligent robots. To this end, this paper provides a systematic and comprehensive overview of applying MODENA in intelligent robots, analyzes the current problems and challenges in the field, and provides an outlook for future research. First, the design automation for the robot morphologies and controllers is reviewed, individually, with automated design of control strategies for swarm robots also discussed, which has emerged as a prominent research focus recently. Next, the integrated design automation of both the morphologies and controllers for robotic systems is presented. Then, the design automation of the vision systems of intelligent robots is summarized when vision systems have become one of the most important modules for intelligent robotic systems. Then, the future research trends of integrated "Body-Brain-Eye" design automation for intelligent robots are discussed. Finally, the common key technologies, research challenges and opportunities in MODENA for intelligent robots are summarized.

Keywords: Modular design automation, Intelligent robots, Bond graph, Evolutionary computation, Neural architecture search

1 Introduction

Robots are widely used in industrial manufacturing, agricultural production, services, and defense to help people perform repetitive, heavy, or dangerous tasks [1]. However, in the case of complex and dynamic tasks and environments, robots without intelligence are unable to respond to changes in a correct and timely manner. Therefore, em-

*Correspondence: zfan@stu.edu.cn
[1] International Cooperation Base of Evolutionary Intelligence and Robotics, Shantou, Guangdong, China
[2] Department of Electronic Information Engineering, Shantou University, Shantou 515063, China
Full list of author information is available at the end of the article

Springer

powering robots with intelligence constitutes an important research trend [2, 3]. Intelligent robots combine artificial intelligence (AI) technology with robotics to produce an autonomous system with intelligence. These systems can learn and respond to dynamic requirements and environmental changes via machine learning, image recognition, target detection, and other AI techniques, rather than simply executing pre-defined commands. The main modules that affect how a robot functions as an intelligent machine include the morphology, the controller, and the vision perception system, which are analogous to the human body, brain, and eyes, respectively. Therefore, in the design automation of intelligent robotic systems, our work aims at developing an automated design methodology for the "Body-Brain-Eye" of intelligent robots.

With the emergence of advanced technologies such as deep learning, evolutionary computing, machine learning, intelligent control, and robotics, the study of design automation for intelligent robots has received significant attention from scholars [4, 5], which is also considered to be an important branch of knowledge-worker automation [6]. In this paper, we systematically provide a detailed explanation of the main concept of modular design automation. In general, modular design automation (MODENA) refers to an approach that decomposes the overall design process of an intelligent robot system into multiple relatively simple and independent functional modules. Each module can be modeled as a unified graph model, which facilitates the optimization of the design. This enables the automatic design and combination of modules. In particular, MODENA for intelligent robots refers to the decomposition of the morphology (body) [7], controller (brain) [8], and vision system (eyes) [9] of an intelligent robot into some independent and interpretable graphical modular units in a digital twin architecture. Then, with the help of artificial intelligence technologies such as genetic programming [10], evolutionary computation [11], deep learning [12], reinforcement learning [13], and causal reasoning [14], these modular units are combined automatically, and the evolution of combination rules is performed. Through this approach, the design process of intelligent robots can be automated. During the design automation process, it is notable that the system that is automatically discovered can be constructed into a new modular unit and added to the module library. This new modular unit can then be utilized in a closed-loop design automation process, allowing for systematical and continuous improvement in the performance of the intelligent robot system. Compared with traditional design methods, the MODENA method can significantly improve the design efficiency and performances of intelligent robots, by promoting the generation of innovative designs not limited by the experiences and intuitions of human designers, and the repetitive trial-and-error processes and laborious routine tasks to be conducted by traditional design methods.

The proposed MODENA approach (see Fig. 1) has received increasing academic attention in recent decades. It applies a constrained multi-objective genetic programming method to automatically generate and evolve the topologies and parameters of graph models (e.g., bond graph, finite state machine, gene regulatory network, deep neural network, and Bayesian network). In this way, the design rules of intelligent robots can be constructed to generate robots with high performance. To efficiently solve the multi-objective programming problem, two key techniques, i.e., constrained multi-objective evolutionary algorithms and genetic programming methods, are simultaneously applied to optimize the topology and parameters of an arbitrary graph structure. Specifically, the constrained multi-objective evolutionary algorithm can efficiently solve multiple conflicting objectives with various types of constraints and a large number of discrete or continuous variables. Genetic programming is used to search for optimization of the topologies and internal parameters of graph models, which can obtain models with innovative optimized structures that perform well in specific aspects. To effectively represent the target object with an appropriate graph model according to its characteristics, we applied different types of graph models for various intelligent robot sub-systems, namely, the morphology, controller and vision systems. Specifically, for the morphology and controller sub-systems, bond graphs are used to unify the modeling of multi-domain physical systems and controller systems, which can conduct comprehensive analysis and modeling of dynamic characteristics. For controllers of swarm robot systems, finite state machines and gene regulatory networks are commonly applied. In particular, finite state machines can abstract robot behaviors into several states, allowing the moving robot to switch among different states. The gene regulatory network is a structural model that integrates the interactions among individuals and their environments, enabling the behavior control of each agent in swarm robots. In vision systems, deep neural networks and Bayesian networks are widely utilized. Deep neural networks are used to learn internal relationships and representation levels of data, enabling robots to achieve human-level analysis abilities on various forms of data, such as text, images and sounds. Bayesian networks, on the other hand, utilize a probabilistic graph model to describe causal relationships of uncertainty among variables, which can process environmental information received by vision systems.

For example, Hod Lipson [15] employed evolutionary computation to design robotic systems automatically in a computer and then created the corresponding prototypes using 3D printing, thereby realizing for the first time the concept of using a machine to design and build machines. That work was published in *Nature* in 2000. Subsequently, Lipson published a series of papers about design automation in *Nature* and *Science* [16–18]. There, he presented

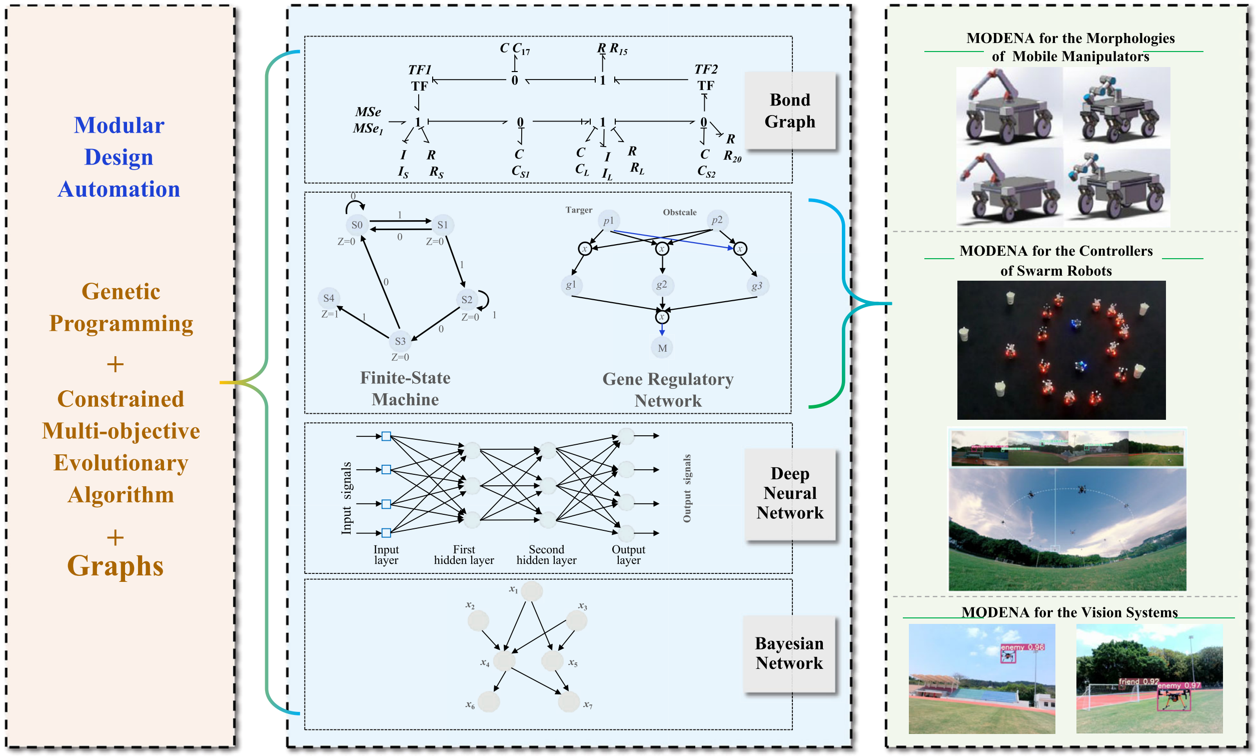


Figure 1 Key components in modular design automation

a more general research question: Can we automatically design a mechatronic or robotic system that can satisfy pre-defined design specifications using Lego-like building blocks? At about the same time, Erik Goodman (the founding director of Beacon center for the study of evolution in action) and his team made breakthrough research in the field of mechatronic design automation (MDA) by employing bond graph (BG) and genetic programming (GP) to automate the design process of general mechatronic systems [19]. BG is a graphical modeling tool that can unify the modeling of multi-domain physical systems in a mechatronic system. GP is a powerful tool in the field of evolutionary computation that can simultaneously optimize the topology and parameters of an arbitrary graph structure. Several circuits and mechanical systems [20–23] have been designed automatically using the bond graph and genetic programming (BGGP) approach, and the combined automatic design of controllers and controlled objects in continuous systems has also been achieved in [24]. In 2007, Clarence D. Silva and his team [25] extended the BGGP approach to allow it to treat nonlinear systems, and proposed the concept of mechatronic design quotients to address design problems involving multiple objectives. In 2012, Zhun Fan and his team [26] proposed an extension of BGGP, called hBGGP with the capability of dealing with both continuous and discrete dynamics as well as designing both the plant and the controller concurrently. The MODENA approach has also been effectively applied to swarm robots. In 2018, Garattoni utilized finite state machines to govern a swarm of robots with complex cognitive capabilities that can perform tasks successfully without knowing the exact execution sequence [27].

To summarize, existing design automation approaches usually pre-define a library of basic modules via a graphical modeling tool. Then, they employ optimization or metaheuristic methods, e.g., evolutionary computation, to search for optimal solutions. When designing mechatronic systems, the modeling language can be a bond graph [28, 29]. In the design of a vision system, the representation can be deep neural networks [30, 31]. When designing the behaviors of swarm robots, the modeling language includes finite state machines and gene regulatory networks [32–34]. These modeling languages are modular and parametric and can be uniformly represented by graphical models. In this paper, systematic and comprehensive reviews of the current state-of-the-art design automation approaches to intelligent robot bodies, controllers, and vision systems are presented. The current problems and challenges of this emerging research field are analyzed, and future research directions are discussed. We purport to attract the attention of the relevant scholars and promote the development of industrial software for design automation of intelligent robots.

The remainder of this paper is organized as follows. Section 2 provides an overview of the design automation for the morphologies of intelligent robots. The design automation for the controllers of intelligent robots is reviewed in Sect. 3. In Sect. 4, the integrated design automation for the morphologies and controllers is presented. Design automation for the vision systems of intelligent robots is summarized in Sect. 5. Section 6 discusses the research and development trends of the integrated design automation of "Body-Brain-Eye" for intelligent robots. Section 7 summarizes and discusses several key technologies, current problems, and challenges involved in the MODENA for intelligent robots. Finally, conclusions are drawn in Sect. 8.

2 Design automation for the morphologies of intelligent robots

MODENA for the morphologies of intelligent robots refers to the systematic use of intelligent design optimization methods to design the robot morphologies, i.e. the plants or mechanical infrastructures. The current research on the design automation for intelligent robot morphologies is primarily divided into two categories: 1) Fixing the morphological topology and optimizing the geometric parameters of the morphology [35–39]. 2) Establishing a library of parametric modules for the morphologies of intelligent robots [40–42], and then simultaneously optimizing the topologies and geometric parameters of the morphologies, by reconfiguring the parameterizable modules.

2.1 Parametric optimization of the morphologies

The optimization of intelligent robots' designs presents a challenging problem which is usually a constrained multi-objective problem with mixed discrete and continuous variables that exhibit non-differentiation, discontinuity, and nonlinearity. The evaluation of some objectives also requires time-consuming simulations. Consequently, evolutionary algorithms are popular choices in practical engineering applications. For example, West et al. [43] utilized a genetic algorithm to optimize the output error system to identify problems for a seven-degree-of-freedom manipulator. The algorithm optimized the parameters of joints to generate a high-performance manipulator. Similarly, Xiao et al. [44] applied NSGA-II to optimize the weight and manipulability of the manipulator, resulting in a lighter and more maneuverable manipulator than the original UR5 structure. Hassan et al. [45] used NSGA-II to optimize a robotic gripper, achieving an optimal gripping force while also revealing significant relationships among objective functions and variable values from Pareto-optimal solutions. In addition, Fan et al. [46] proposed a push and pull search framework [47] combined with a multi-objective evolutionary algorithm based on decomposition to optimize a six-degree-of-freedom teaching manipulator. Their approach resulted in designs that outperformed those of human engineers and some popular constrained multi-objective evolutionary algorithms. Additionally, reinforcement learning has been employed to optimize the parameters of morphologies. As an example, Zhang et al. [48] proposed an algorithm that utilizes reinforcement learning to automate optimal robot hand design, demonstrating its effectiveness in tasks such as grasping boxes, cylinders, and spheres.

2.2 Integrated design automation for parameters and topologies of morphologies

Modular robots [49–52] embody the principles of integrated design automation, which incorporates the optimization of parameters and topologies to create diverse morphologies. Modular graph models for the morphologies of intelligent robots are composed of either homogeneous or heterogeneous modules, each of which involves a variety of actuators and sensors [53, 54], which allows intelligent robots to achieve self-assembly, self-reconfiguration and self-repair. For example, Lipson et al. [15] were not only the first to use modules from a pre-defined library of modules to automatically assemble electromechanical systems that meet pre-defined functional requirements but were also the first to apply evolutionary algorithms to design robotic systems on the computer. Kelly et al. [55] applied a stochastic optimization algorithm to autonomously assemble a model for planar distributed assembly, which achieved innovative designs. Inspired by the large and complex nests built by social insects, Werfel et al. [56] established a distributed system for automating construction, which built some particular desired structures according to a high-level design provided by users. Inspired by the principles of biological evolution [57], Dai et al. proposed the metamorphic theory [58], which allows the topologies of morphologies to be reconfigured and metamorphosed [59] and to evolve dynamically [60] according to actual needs, thus flexibly adapting to changing working environments and functional requirements. On this basis, a variety of robots have been developed, such as a hybrid continuum robot based on pneumatic muscles [61], a crawling robot [62], and a quadruped robot based on the metamorphic mechanism [63, 64].

With the development of topology optimization design methods, modular robots are increasingly applying such methods to achieve innovative designs of morphologies [66, 67]. Compared with traditional topology optimization design methods (e.g., the level set method [68], the evolutionary structural optimization method [69], and the moving morphable component method [70]), isogeometric topology optimization (ITO) [71] is a modern structural optimization technique that leverages isogeometric

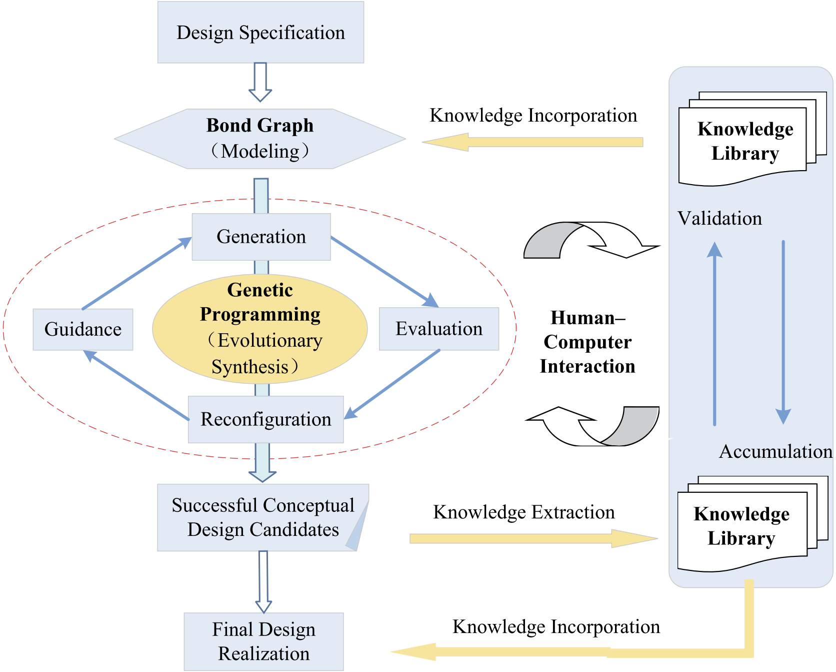


Figure 2 The framework of evolutionary synthesis of mechatronic systems [65]

analysis. Specifically, ITO seamlessly integrates computer-aided design, computer-aided engineering, and structural topology optimization, laying a theoretical foundation for the integration of design, analysis, and optimization of the morphologies for intelligent robots [72]. In recent years, ITO has been extensively studied and has driven the development of a new generation of digital design. For example, Gao et al. [73–75] studied the ITO method to design new materials and structures with special properties, such as auxetic metamaterials [76] and ultra-lightweight architected materials [77]. To improve the stability and accuracy of the optimization process and broaden the application scenarios of topology optimization, Seo et al. [78] proposed a new ITO, which can eliminate the design space dependency. Wang et al. [79] integrated isogeometric analysis with the level set method and proposed a high-precision ITO that satisfies geometric constraints. ITO enables the integration of digital design and analysis, thus significantly shortening the development cycle of the morphologies of intelligent robots and reducing research and development costs.

BGGP combines the capability of bond graphs (BG) to represent the mixed-domain physics of generic mechatronic systems in a unified way, and of genetic programming (GP) to explore in an open-topology design space automatically and optimize both the topologies and parameters of design candidates represented by bond graphs. For example, Fan et al. [19, 29, 81] proposed an automatic design method for mechatronic systems combining bond graphs and genetic programming, which has already been applied to the design of electrical and mechatronic systems, such as analog filters [81], electric filters [19] and the driver system of a printer [29]. Meanwhile, Wang et al. [24] proposed a knowledge-based evolutionary design framework for mechatronic systems by combining the BGGP method with human knowledge, as shown in Fig. 2. In the BGGP method, BG is used to model multi-domain systems and GP is employed to search the open-end design spaces automatically. Figure 3 illustrates the mapping from genotype to phenotype in the BGGP method. Compared with other methods, the BGGP method has a distinct advantage of being able to search in a topologically open-ended design space that is represented uniformly by bond graphs. As a special kind of mechatronic system, robotic systems can also utilize the BGGP approach to the design automation of their morphologies. Because modular robotic morphologies involve many physical subsystems, they need a unified expression to model and analyze their performance. BG, as a modeling language that can describe all physical sub-systems (and continuous controllers) uniformly, can be utilized to model and analyze the dynamics of the designed mechatronic systems effectively and efficiently [82, 83].

In conclusion, many achievements have been made in design automation for the parameters and topologies of the robot morphologies. In particular, self-assembly [84],

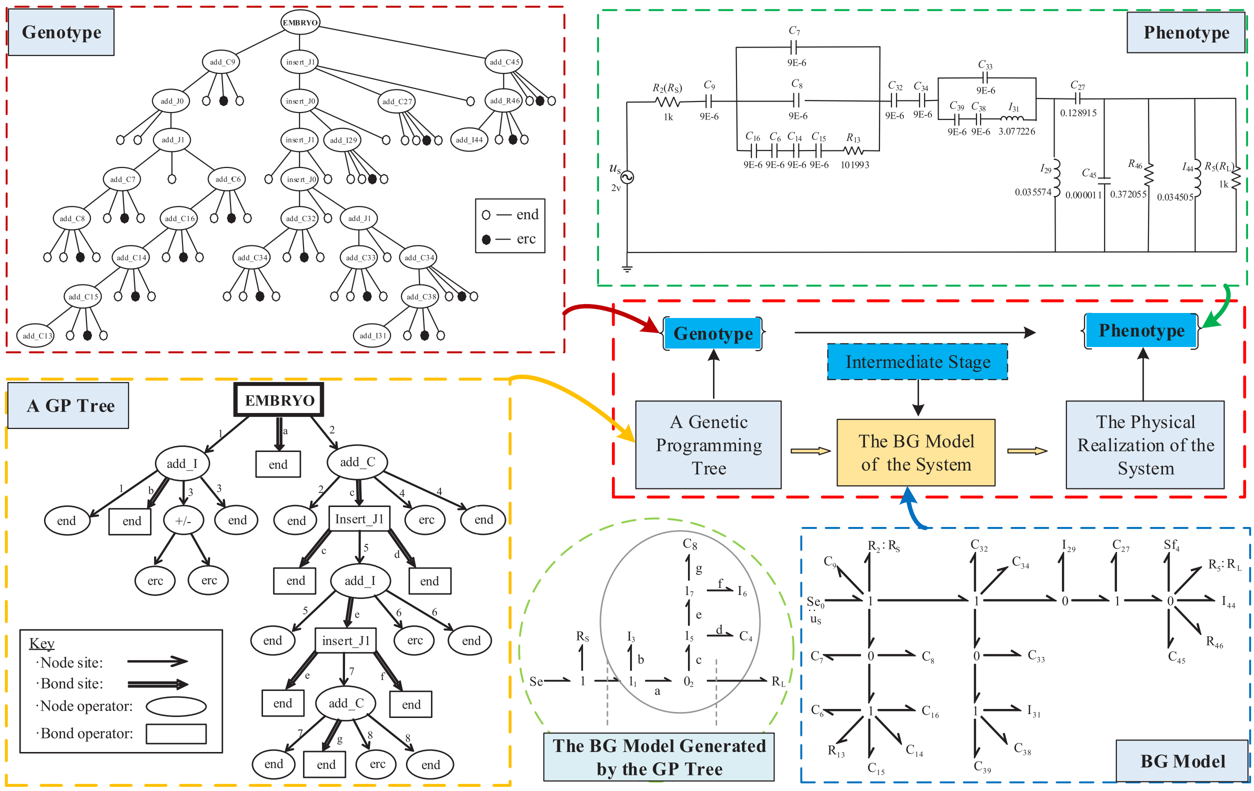


Figure 3 An example of genotype-phenotype mapping [80]

self-reconfiguration [85] and self-repair [86, 87] characteristics of modular robots demonstrate the superiority of applying design automation for parameters and topologies of the morphologies. It is noted that the controller is also an important part of an intelligent robot, and the next section will detail the design automation for the controllers of intelligent robots.

2.3 Summary

In summary, design automation for the morphologies of intelligent robots has been widely applied, which can simultaneously optimize the geometric parameters and topologies of robots. Here, we present a concise overview of the various methods reviewed, highlighting the connections and differences among them from multiple perspectives, as displayed in Fig. 4.

The research on the design automation for the morphologies of intelligent robots is mainly divided into three categories: (1) Optimizing geometric parameters while keeping a fixed morphological topology [43–46, 48]. These methods usually use multi-objective evolutionary algorithms [43–46] or reinforcement learning methods [48] to optimize the geometric parameters to meet task-specific requirements and obtain an optimal design. Since the topology is fixed, it is difficult to adapt to complex tasks.

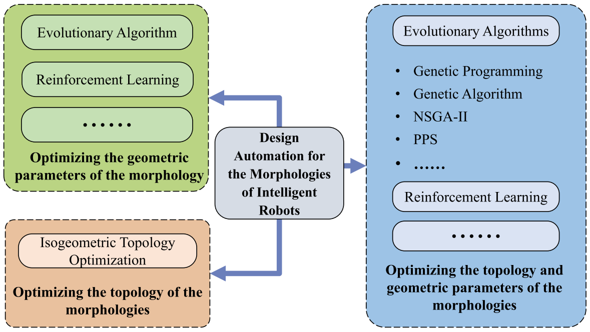

Evolutionary Algorithm
Reinforcement Learning
······
Optimizing the geometric parameters of the morphology
Isogeometric Topology Optimization
Optimizing the topology of the morphologies
Design Automation for the Morphologies of Intelligent Robots
Evolutionary Algorithms
• Genetic Programming
• Genetic Algorithm
• NSGA-II
• PPS
• ......
Reinforcement Learning
······
Optimizing the topology and geometric parameters of the morphologies

Figure 4 A summary of design automation for the morphologies of intelligent robots

(2) Topology optimization methods. These methods are represented by isogeometric topology optimization [73–75, 78, 79]. After setting the design space of the topology structure, optimization objectives and constraints, these methods can automatically perform topology optimization design of the robot system's components based on the implementation of computer aided engineering (CAE) analysis. These topology optimization methods can not only shorten the design cycle but also improve the

design quality. However, the current work is mainly focused on the topology design of the components of intelligent robots. (3) Simultaneous optimization of topologies and geometric parameters of robot morphologies [15, 19, 24, 55, 56, 81]. These methods usually decompose the morphologies of intelligent robots into a series of independent modular units, and then achieve assembly automation and parameter design by using evolutionary computation or reinforcement learning techniques. However, these approaches rarely perform CAE analysis of the assembled morphologies, which cannot perform testing using computer simulations and provide valuable insights into the performance of robot morphologies during the early development phase. To summarize, although a large number of in-depth studies have been conducted on design automation for the morphologies of intelligent robots, further research is still required on how to conduct efficient design automation methods to meet the requirements of dynamic and complex tasks and environments.

3 Design automation for the controllers of intelligent robots

3.1 Design automation for the controllers of individual robots

In an intelligent robotic system, the controller often plays a key role [88, 89]. Many studies [90] have conducted in-depth research on the design automation for the controllers of intelligent robots. For example, Zhong et al. [91] proposed a novel kinematic calibration method based on an improved whale swarm algorithm to optimize the controller design of a biped robot to enable the robot to walk continuously and smoothly on complex ground. Due to the complexity of the walking dynamics of the biped robot, Gao et al. [92] applied a pre-trained neural network to design an optimal gait control model. Simulation results showed that the control model could effectively improve the maximum walking speed and terrain adaptability in a short time. In addition, hydraulic actuators are frequently employed in biped robot controllers. Nevertheless, due to the nonlinearity of hydraulic systems, their dynamic performance of the systems under control requires further improvement [93]. To this end, Dong et al. [94] proposed an improved drone squadron optimization-based approach to optimize the design of the hydraulic controller. The comprehensive experimental results indicated that the optimized hydraulic controller had better stability and higher accuracy.

In addition, proportional-integral-derivative (PID) controllers have been widely utilized in intelligent robots due to their advantages of simple design, easy implementation, fast response, and small steady-state error. Many studies [95–98] have conducted in-depth research on the design optimization of PID controllers. For example, Sharma et al. [99] applied the cuckoo search algorithm to optimize the parameters of the fractional-order fuzzy PID controller for a two-link planar rigid robotic manipulator. Experimental results demonstrated that the optimized PID controller outperformed the other controllers in terms of trajectory tracking, model uncertainty, disturbance rejection, and noise suppression. For the trajectory tracking of autonomous mobile robots, Ali et al. [100] employed an artificial bee colony to optimize the parameters of a PID controller, which obtained two high-performance PID controllers (speed controller and azimuth controller). Taherkhorsandi et al. [101] proposed an adaptive and robust controller that combines PID with sliding control to better control the motion of a biped robot. They utilized a multi-objective genetic algorithm to optimize the controller, resulting in successful control of a biped robot walking on a slope in the lateral plane. In general, PID controllers have difficulty in achieving optimal control of complex and nonlinear control systems [102]. To this end, Sun et al. [103] established a set of component units and performance units, and designed an optimal controller using the differential evolution algorithm. On this basis, Xin et al. [104] proposed a general design automation method for controllers to simultaneously optimize the structures and parameters of the controllers. Their approach combines basic controller components and related parameters to automatically create an optimal control model tailored to specific requirements.

In addition to the design automation methods mentioned above for PID controllers, many studies have employed neural networks as controllers for intelligent robots [105–107]. For example, Gallagher et al. [108] developed an approach in which they evolved neural networks in simulation to control the locomotion in an artificial insect, and successfully transferred the controller to a real hexapod robot. Nolfi et al. [109] applied an evolutionary algorithm to design and optimize a neural controller, which makes a bipedal robot equipped with actuators and sensors move according to concentration differences. In Paul et al.'s study [110], an evolutionary algorithm was used to optimize the design of a closed loop recurrent neural network controller, which achieved stable and bipedal movements on a 5-link biped robot in a physics-based simulation environment. In addition, Rahmani et al. [111] proposed a novel adaptive neural network integral sliding-mode controller that utilized a bat algorithm to control a biped robot, and proved its stability using the Lyapunov theory.

3.2 Design automation for the controllers of swarm robots

Traditional control methods were initially designed to control the motions of individual robotic systems. However, when the scale of intelligent robotic systems is enlarged with numerous individual robots involved, traditional control approaches may face many challenges.

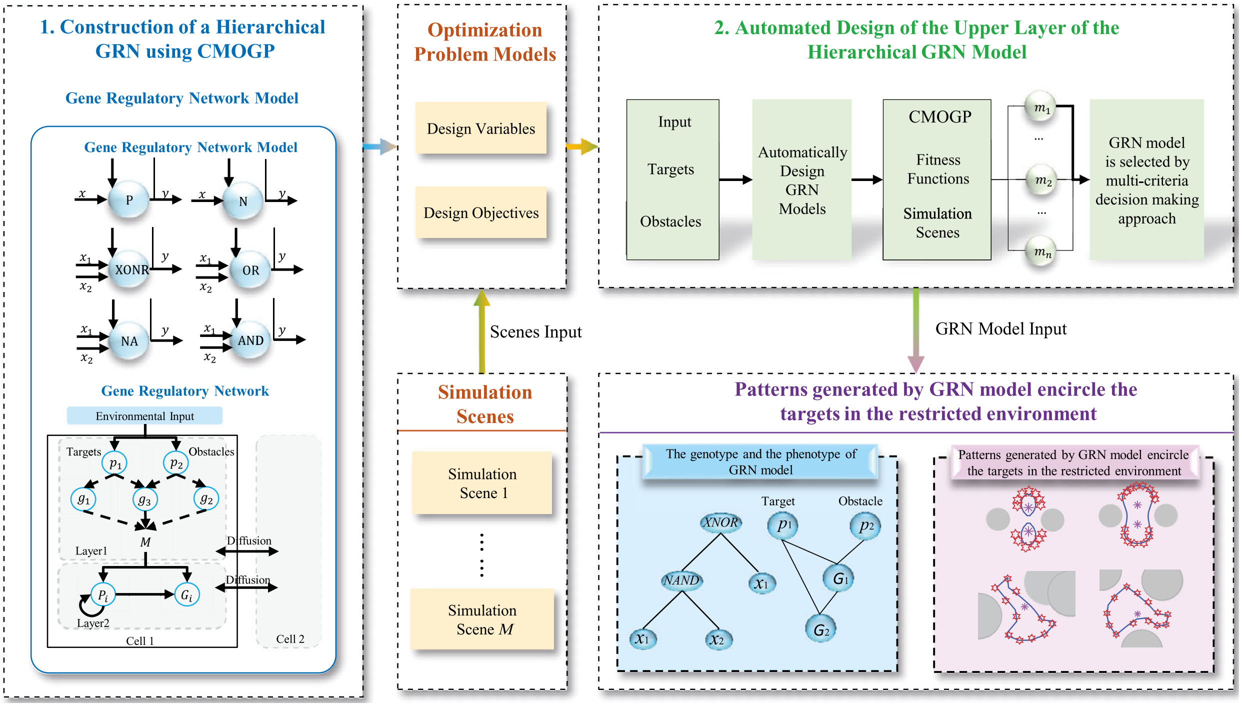

1. Construction of a Hierarchical GRN using CMOGP
Gene Regulatory Network Model
Gene Regulatory Network Model
P
N
XONR
OR
NA
AND
x
y
x_1
x_2
Gene Regulatory Network
Environmental Input
Targets
Obstacles
p_1
p_2
g_1
g_3
g_2
M
Layer1
Diffusion
Diffusion
P_i
G_i
Layer2
Cell 1
Cell 2
Optimization Problem Models
Design Variables
Design Objectives
Scenes Input
Simulation Scenes
Simulation Scene 1
Simulation Scene M
2. Automated Design of the Upper Layer of the Hierarchical GRN Model
Input
Targets
Obstacles
Automatically Design GRN Models
CMOGP
Fitness Functions
Simulation Scenes
m_1
m_2
m_n
GRN model is selected by multi-criteria decision making approach
GRN Model Input
Patterns generated by GRN model encircle the targets in the restricted environment
The genotype and the phenotype of GRN model
Target
Obstacle
XNOR
NAND
p_1
p_2
x_1
x_2
G_1
G_2
Patterns generated by GRN model encircle the targets in the restricted environment

Figure 5 Diagram of the automated design framework [33] for entrapping pattern generation

These challenges include insufficient fault tolerance, meaning that the failure of a few individuals may lead to the failure of the whole system, a significant increase in computational overhead, making it difficult to respond to unexpected occurrences timely, and other issues. The design automation of controllers for swarm robots provides a viable solution to the above difficulties. To this end, some studies have extracted the basic unit of swarm behavior by exploring the mapping between swarm behavior and individual behavior [112–116]. Then, an evolutionary computation-based swarm behavior control framework suitable for dynamic and complex task environments is automatically designed. For example, Francesca et al. [117] abstracted some individual behavior into several states (such as random motion and static state) and then applied an optimization algorithm (named F-Race) to automatically design controllers based on a probability finite state machine. In the following year, Francesca et al. [118] improved the design of control software for robot swarms and proposed two automated design methods (Vanilla and EvoStick). The experimental results demonstrated that the proposed design automation methods outperformed human designers in specific experimental scenarios. Although the works [117, 118] successfully addressed relatively simple or constrained problems, their limitations quickly emerged as the problem complexity increased [119]. In particular, a complex task is made of several subtasks that may require cooperation and have mutual dependencies and time constraints [120]. To this end, Fan et al. [33] constructed a library of logical relationships of information exchange between agents by learning from the method of information exchange between cells in organisms. They then applied genetic programming to automatically design the optimal swarm behavior control model so that swarm robots can entrap targets in different patterns according to different environments (as shown in Fig. 5). Furthermore, Wu et al. [121] refined individual simple behavioral rules with universal applicability (such as exploration, moving to the target, and avoiding obstacles) through an in-depth analysis of the flocking task. They then optimized these individual behavior rules by combining behavioral trees and the proposed heterogeneous–homogeneous co-evolution method to automatically design swarm behavior control strategies. Currently, these studies [33, 121] are mainly in laboratory environments or simulation environments, and few studies are deployed in practical application environments. To this end, Vásárhelyi et al. [122] applied CMA-ES to optimize the design of the swarm control mechanism by considering the presence of machine failures, communication delays, and airflow disturbances in actual flight, which achieved a successful flocking flight in the field with 30 unmanned aerial vehicles (UAVs).

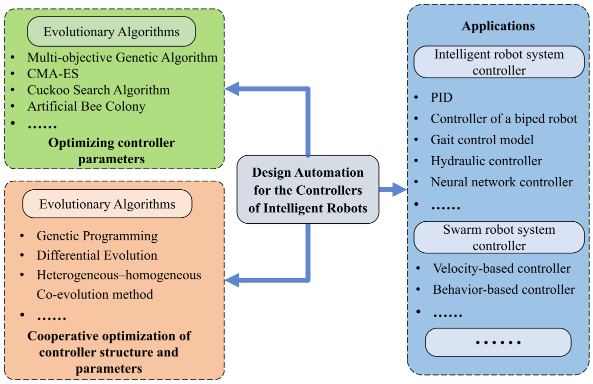

Evolutionary Algorithms
Multi-objective Genetic Algorithm
CMA-ES
Cuckoo Search Algorithm
Artificial Bee Colony
......
Optimizing controller parameters
Evolutionary Algorithms
Genetic Programming
Differential Evolution
Heterogeneous–homogeneous Co-evolution method
......
Cooperative optimization of controller structure and parameters
Design Automation for the Controllers of Intelligent Robots
Applications
Intelligent robot system controller
PID
Controller of a biped robot
Gait control model
Hydraulic controller
Neural network controller
......
Swarm robot system controller
Velocity-based controller
Behavior-based controller
......
......

Figure 6 A summary of design automation for the controllers of intelligent robots

3.3 Summary

To summarize, research on the design automation of the controllers is a key procedure to achieve the design automation of the entire intelligent robots. In this regard, we have summarized the characteristics and applicability of various design automation methods for controllers in two different aspects: the applied techniques and target objects, such as single robot controller and swarm robot controller, as shown in Fig. 6.

The research on the design automation for controllers consists of two main aspects: (1) Optimizing the geometric parameters of the controller with a fixed controller topology. For example, evolutionary algorithms, such as MOGA [101, 108–110], CMA-ES [122] and hybrid evolutionary algorithms (such as the improved whale swarm algorithm [91], cuckoo search algorithm [99], artificial bee colony [100], and bat algorithm [111]), are applied to optimize controller parameters [91, 94, 99–101, 108–111, 122]. (2) Simultaneous optimization of the topologies and geometric parameters of the controller [33, 92, 103, 104, 117, 118, 121]. These methods usually pre-build various modular control units and then apply evolutionary algorithms to automatically assemble and parameterize these units, resulting in the automatic design of the optimal controller topology and parameters.

From the perspective of the scale of controlled objects, the design automation of controllers can be divided into two categories: (1) Design automation for the controllers of single robotic systems [91, 94, 99–101, 108–111]. (2) Design automation for controllers of swarm robotic systems [33, 103, 104, 117, 118, 121, 122]. Compared to the design of a single robot controller, designing a swarm robot controller is more complex. The main reason is that the mapping mechanism from swarm behavior control to individual behavior control is not clear. Designing behavior control rules for each robot in the swarm robot to generate intelligent swarm behavior at the system level is an important research direction in the future.

4 Integrated design automation for the morphologies and controllers of intelligent robots

In recent years, researchers have introduced the idea of biological evolution into integrated design automation for morphologies and controllers of intelligent robots [123–126], which can automatically identify the optimal designs of intelligent robots according to fitness functions determined by given tasks or environments. Based on these ideas, some studies [127–129] have proposed an underlying system architecture called the triangle of life, which consists of three stages: morphogenesis, infancy, and mature life. This system allows for a population of robotic organisms that evolve and adapt to the given environment. Additionally, evolutionary computation, as a biologically-inspired algorithm, has been used in numerous studies for integrated design automation of morphologies and controllers of intelligent robots [53, 130]. Modular robots can integrate the morphologies and controllers into a whole and simplify the search space, improving the efficiency of evolutionary computation [51]. Thus, the design automation of modular robots based on evolutionary computing has become an important research method for integrated design automation for the morphologies and controllers of intelligent robots. For example, Marbach et al. [131] utilized genetic programming to integrate configuration and control of locomoting homogenous modular robots, breaking through the limitations of human designers' experience and intuitions in manual design methods. It is worth noting that crossover and mutation in the evolutionary process may cause mismatches between robot morphologies and controllers of the offspring. To alleviate this problem, Agrim Gupta et al. [132] designed a deep evolutionary reinforcement learning framework, which learned challenging motor tasks in complex environments by evolving different surrogate models. The study confirmed that environmental complexity can promote the evolutionary design of robots, helping offspring robots learn new skills. Furthermore, the study confirmed that the robot structure is related to the learning efficiency of the controller. An excellent structure can promote the effective learning of the offspring robots.

Recently, neural network-based approaches have been widely applied in integrated design automation for the morphologies and controllers of intelligent robots [133, 134]. A RoboGrammar system inspired by arthropods was proposed by Zhao et al. [135]. The proposed system could efficiently generate hundreds of thousands of robotic structures composed of the given components. Then, high-performance robots were found by applying graph heuristic search and model predictive control (MPC), achieving concurrent optimization of robot morphologies and controllers. By extending the single-objective graph heuristic search procedure based on the RoboGrammar system, Xu et al. [136] proposed a new multi-objective co-design algorithm for obtaining Pareto-optimal robot topologies and controllers. Aslan Miriyev and Technology and Mirko Kovač [137] created a symbiotic human–robot ecosystem (physical artificial intelligence) through the integrated evolution of the organism, control, morphology, action execution, and perception. The ecosystem decides and adapts in real-time for navigation, locomotion, and manipulation by processing combinations of signals simultaneously sent from multiple sensors in their "body" to their "brain".

In addition, genetic programming can also be utilized for efficient integrated design automation of the morphologies and controllers of electromechanical systems. For example, Wang et al. [138] proposed a "body-brain" design automation method that integrates GP and bond graphs to automate the integrated design of a quarter-car suspension control system's morphologies and controllers. Compared with traditional methods, this method can help designers to achieve more creative and flexible designs. In addition, Dupuis et al. [26] proposed a design automation method called HBGGP, which merges hybrid bond graph (HBG) and genetic programming (GP) into the evolutionary design of topologies and parameters of a hybrid dynamical system. In the proposed method, HBG is utilized to represent dynamic systems involving both continuous and discrete system dynamics, and GP is used to explore the open-ended design space of HBGs to optimize the morphologies and parameters of DC-DC converters. Thereafter, they investigated the evolutionary design of controllers for hybrid mechatronic systems [139] and employed a finite state automaton (FSA) to represent discrete controllers. A case study of a two-tank system demonstrated that the proposed evolutionary approach can lead to a successful design of an FSA controller for the hybrid mechatronic system.

To summarize, the integrated design automation of the morphologies and controllers of intelligent robots is an important trend in future research. Separate consideration of the design automation of the morphologies and the controllers would lead to sub-optimal solutions and unsatisfactory overall performance. Here, we summarize the characteristics and applications of various methods from the perspective of research directions and applied optimization techniques of integrated "body-brain" design automation for intelligent robots, as illustrated in Fig. 7.

The current research directions for morphologies and controllers mainly consist of three aspects: (1) Designing the search space [26, 135, 138]. It is crucial to construct a reasonable search space so that novel solutions can be found. (2) Designing the search strategy [26, 131, 132, 135, 136]. A good search strategy can improve the efficiency and effectiveness of the algorithm. (3) Designing evaluation indicators [131, 135]. The evaluation indicators of comprehensive performance are designed to evaluate the performance of search candidates and guide the algorithm's search.

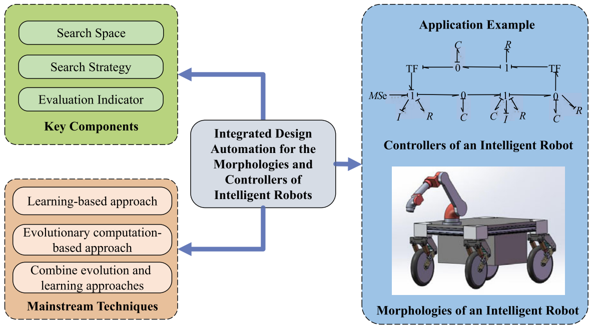

Search Space
Search Strategy
Evaluation Indicator
Key Components
Learning-based approach
Evolutionary computation-based approach
Combine evolution and learning approaches
Mainstream Techniques
Integrated Design Automation for the Morphologies and Controllers of Intelligent Robots
Application Example
C
R
TF
0
1
TF
MSe
1
0
1
0
I
R
C
C
I
R
C
R
Controllers of an Intelligent Robot
Morphologies of an Intelligent Robot

Figure 7 A summary of integrated design automation for the morphologies and controllers of intelligent robots

According to the applied optimization techniques, the integrated design automation for the morphologies and controllers is divided into three main categories: (1) Evolutionary computation-based approaches [131, 136]. These approaches focus on finding the best design solutions for the integrated design of the morphologies and controllers by simulating the evolutionary process in nature. The advantage of these methods is that they allow the design of solutions that are superior to manual ones. However, due to the complex design space and randomness in the search process, the optimal design is not guaranteed. (2) Learning-based approaches [135]. These approaches focus on learning the integrated design strategies for the morphologies and controllers by setting appropriate reward functions and making dynamic decisions with known knowledge to obtain a design solution that maximizes rewards. The method simplifies the design space and improves the search efficiency through a heuristic search method, and is suitable for the integrated design automation for the morphologies and controllers of intelligent robots with complex structures. (3) Combination evolution and learning approaches [26, 132, 137, 138]. These methods mainly apply the evolution-based method to design the morphologies, and then apply the learning-based method to design the controllers, which can effectively reduce the search space and improve search efficiency.

The integrated design automation of the morphologies and controllers of intelligent robots presents a challenge due to the strong coupling relationship between the morphology and controller, as it involves multi-energy domain physical systems. This makes it an important area for further research.

5 Design automation for the vision systems of intelligent robots

The vision systems of intelligent robots can provide rich visual perception information, such as depth information and motion information. This information is often one of the most important components for guiding the intelligent robot's motion-decision-making process [140]. However, in practice, vision systems are often designed manually. In most cases, designers require numerous trial-and-error experiments to obtain an appropriate design scheme for the vision system [141, 142]. Design automation for vision systems can be used to automatically design an optimal or desired vision system design scheme for the robotic vision tasks needed. Therefore, design automation for vision systems represents an indispensable element of design automation for intelligent robots.

Computer vision research provides an essential foundation for the design automation of robotic vision systems, where deep learning has become a crucial research direction in this field. Researchers can obtain desired results by constructing a neural network and using the corresponding image data for training, provided that the neural network architecture is properly designed. However, the design of neural network architecture requires designers to have a full understanding of various computing modules and training methods. In addition, designers have to conduct repeated experiments to adjust network architectures to produce optimal architectures with excellent performance [142–144]. In recent years, neural architecture search (NAS) has gradually emerged as a research hotspot. In a given search space, NAS can automatically identify optimized neural network architectures without manual design. Therefore, NAS provides an important foundation for the design automation of intelligent robotic vision systems.

This section introduces the recent work related to NAS and highlights the shortcomings of existing research. It also identifies the problems that need to be addressed in the future to achieve the design automation of intelligent robotic vision systems.

5.1 Neural architecture search

NAS is primarily composed of three parts: search space design, search strategy, and performance estimation strategy. Depending on the search strategy, NAS can be mainly classified into three categories [145–147]: (1) RL-based NAS, (2) differentiable NAS, and (3) evolutionary NAS.

The RL-based NAS models the search task as a Markov decision process and offers rewards depending on the performance of the generated network after training on a test set. Then, the method trains the RL model according to the reward and adjusts the generated neural network architecture, thereby using the RL to guide the neural network architecture generation. Representative achievements include MetaQNN [148] (proposed by MIT) and NASNet [149, 150] (proposed by Google), both of which search the layers of the neural network. In contrast, BlockQNN [151] (proposed by Shangtang Technology) searches modules of the neural network. Unlike the application of evolutionary algorithms or RL to a discrete and non-differentiable search space, differentiable methods make architecture searches more efficient by using gradient information through the continuous relaxation of the architecture representation [152]. The network architectures designed by differentiable-based NAS have also achieved excellent performances with representative examples, including the differentiable architecture search (DARTS) [152] (proposed by Google Brain) and PDARTS (proposed by Huawei's Noah's Ark Laboratory) [153]. Evolutionary NAS regards the topological structure and superparameter adjustments of the model as an optimization problem and adopts an evolutionary algorithm to optimize the neural network. In 2019, the Uber AI Lab published a review article in *Nature Machine Intelligence* that strongly advocated the evolutionary NAS and anticipated its future development [154]. Representative evolutionary NAS examples include the neuroevolution of augmenting topologies (NEAT) [155], CoDeepNEAT [156], and NSGA-Net algorithms [157].

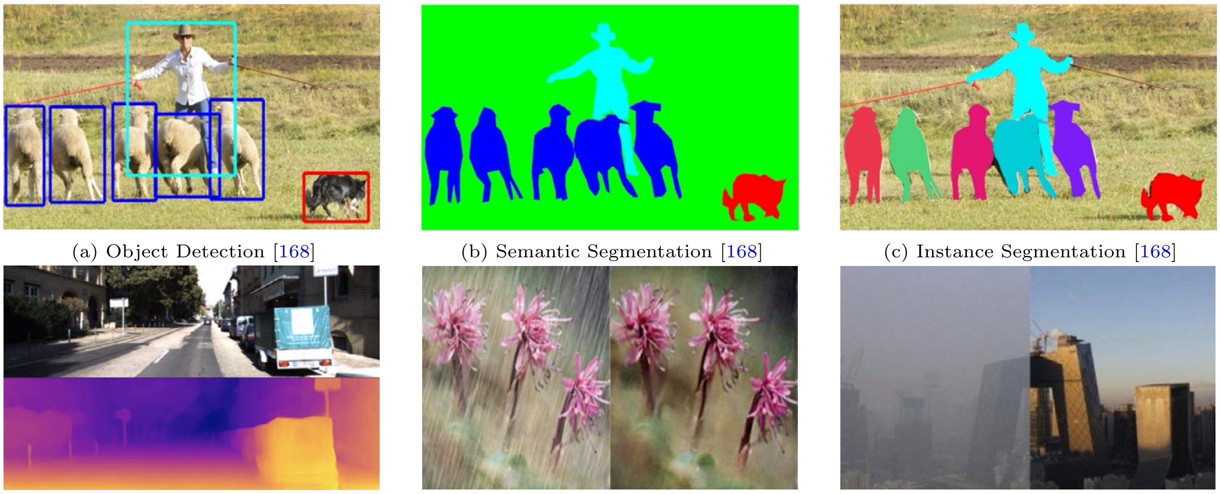
5.2 Design automation for vision systems

In real life, robots assigned to different tasks require different visual capabilities. For example, drones use object detection [158], object tracking [159], motion estimation [160] and depth estimation [161] for autonomous obstacle avoidance. Autonomous cars use 3D object detection [162] to establish the physical positions of obstacles for path planning. Medical robots use image segmentation [163–165] to analyze the information in medical examination reports and thereby help doctors diagnose a patient's condition, and more (see Fig. 8). Different from laboratory studies, robots in practical applications typically are unable to provide sufficient computing resources with the embedded devices offered. Consequently, the development of light-weight models is a promising research area.

Thus, by investigating the vision tasks often encountered in current robot applications, this section introduces design automation for vision systems involved in the vision tasks that robots currently face, including (1) object detection, (2) image segmentation, (3) depth estimation, (4) video analysis, and (5) embedded device application.

5.2.1 Neural architecture search for object detection

Object detection can enable a robot to identify the object of interest in an image and determine its position, allowing the robot to perform tasks such as object picking [170, 171], object tracking [172, 173], and other tasks. The network architecture of object detection is primarily classified into three parts: the backbone, neck, and head. The



(a) Object Detection [168]

(b) Semantic Segmentation [168]

(c) Instance Segmentation [168]

(d) Depth Estimation [169]

(e) Image Deraining [170]

(f) Image Dehazing [171]

Figure 8 Different visual tasks that robots often encountered, which include object detection [166], semantic segmentation [166], instance segmentation [166], depth estimation [167], image deraining [168] and image dehazing [169]

backbone is responsible for extracting image features, the neck is responsible for fusing features, and the head is responsible for classifying and locating objects. Currently, two main methods are available for object detection architectures: (1) searching for the overall network architecture [174] and (2) searching for parts of the network architecture while using other parts of the existing network architecture [175]. Depending on the problem characteristics of the object detection tasks and the characteristics of the network structures, various methods have been introduced for searching object detection network architectures.

Chen et al. [176] proposed the DetNAS algorithm to address the problem of losing object location features when directly using an image classification network as the backbone for object detection. To achieve this, they search the entire network architecture using ShuffleNetV2 as the search space. The algorithm is pre-trained on ImageNet datasets and fine-tuned on object detection task datasets to improve classification and localization capabilities. Meanwhile, DetNAS employs an evolutionary algorithm to search the sub-network. Wang et al. [175] proposed NAS-FCOS, a fast neural architecture search algorithm for object detection, to reduce the computational burden and improve search speed. The algorithm uses an existing image classification network, such as ResNet or MobileNet, as the backbone network and constructs the network according to the feature pyramid network (FPN) and detection head. NAS-FCOS searches only the network structures of the FPN and detection header in different search spaces. The algorithm employs a long short-term memory (LSTM) network as an agent and uses an RL-based search strategy to build a network for the FPN and detection header. Structural-to-modular NAS [177] adopts a two-stage search strategy to search network architectures for object detection. In the first stage, different existing networks are combined based on the structure of the target detection network to identify the combination of network structures that achieved the Pareto optimum in terms of inference speed and accuracy. In the second stage, all network structures in the Pareto solution set are further searched in different modules.

In recent years, NAS for object detection has received increasing attention and achieved very competitive results. However, how to define an optimal search strategy and search space remains a problem for object-detection NAS.

5.2.2 *Neural architecture search for image segmentation*

Image segmentation is a process that involves classifying each pixel in an image, making it a dense prediction task. Robots can utilize image segmentation for various functions, including defect detection and measurement [178] and medical analysis [179, 180]. Currently, image segmentation architecture search methods fall into two categories. The first category involves searching for the module structure under a fixed network architecture, while the second category involves searching for both the network architecture and module structure simultaneously.

Liu et al. [181] proposed Auto-DeepLab, which first applied NAS to image segmentation. Auto-DeepLab uses architecture- and cell-level search methods to explore the overall architecture of the model and cell structure, respectively. It formulates the architecture search problem as a differentiable optimization one and uses the gradient-based method to search the model architecture. To quickly

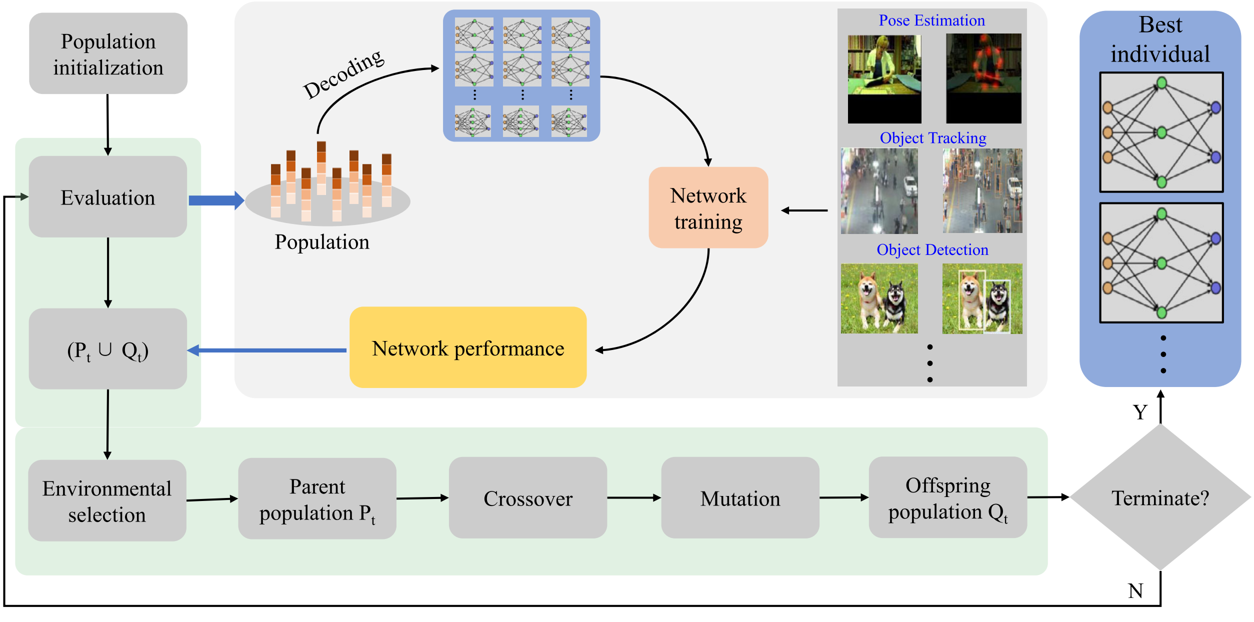


Figure 9 The framework of Genetic U-Net [183]. In the framework, if the dataset is pose estimation [184], object tracking [185] or object detection [186], the framework can automatically generate the corresponding optimal neural network model

search a lightweight semantic segmentation network for mobile device applications, Nekrasov et al. [182] employed the existing network architecture as the encoder and focused on searching the decoder network architecture under the encoder-decoder network architecture. Wei et al. [183] proposed a Genetic U-Net estimation for retinal vessel segmentation, which takes U-shaped encoder-decoder structure as the network architecture and explores the network structure within each cell in the encoder network and decoder network by an evolutionary algorithm. Genetic U-Net uses binary coding to encode the network structure and regards the network performance on the test dataset as the fitness of individuals. Through genetic operations such as selection, crossover and mutation, better offspring individuals are evolved continuously and finally the network structures with the best performance are identified. Experimental results show that Genetic U-Net has higher segmentation accuracy yet fewer parameters than existing algorithms in DRIVE, STARE, CHAS_DB and HRF public datasets. It is worth noting that Genetic U-Net is a rather general framework, which can conveniently switch to different vision tasks and generate optimal models according to the provided training data, as depicted in Fig. 9.

5.2.3 *Neural architecture search for depth estimation*

Depth estimation enables robots to calculate the distance to objects by analyzing images [167, 187]. These estimations are crucial in downstream tasks such as autonomous obstacle avoidance [188, 189] and path planning [190, 191], making them important vision functions for robots. Monocular and binocular depth estimation techniques are the two most commonly used vision systems in robots. Therefore, this section primarily focuses on introducing architecture search algorithms for monocular and binocular depth estimation.

Monocular depth estimation directly predicts the depth map of the input image, which is an intensive prediction task. Huynh et al. [192] proposed LiDNAS to search for lightweight monocular depth estimation networks. Under the preset network architecture, each module structure is searched using an auxiliary tabu search algorithm. During network training, the prediction accuracy and the number of parameters are used to obtain a network model with fewer parameters and higher estimation accuracy. Saikia et al. [193] extended DARTS [152] and applied it to depth estimation tasks by using AutoML technology to efficiently search for optimal network structures. Nekrasov et al. [182] utilized a method to search for a lightweight semantic segmentation network architecture for depth estimation, which resulted in competitive performance compared to manually designed depth estimation networks.

Binocular depth estimation primarily involves identifying matching points in left and right images using stereo matching [194]. A stereo vision system model is then used to estimate the depth map. Therefore, the architecture search in the binocular depth estimation task is one of the search tasks for the stereo matching network model. This network is typically composed of two parts: a feature extraction network and a matching network. Inspired by multi-resolution feature extraction and fusion, Cheng et al. [195] proposed the learning effective architecture stereo algorithm. This algorithm, which is based on a gradient-based search strategy, adopts a two-level hierarchical search strategy to search the network architecture and the internal structures of the constitutive modules simultaneously. To solve the problem of decreased matching accuracy in unseen scenes, Zhang et al. [196] established a reusable architecture growth framework that allows the resulting network to learn to match stereo unseen scenes. Wang et al. [197] introduced an elastic and accurate network for stereo matching (EASNet), which divides the network architecture into four components based on different functions. The search space of each component includes manually designed calculation modules for stereo matching. Experiments show that EASNet achieves superior results in terms of both inference speed and matching accuracy.

To summarize, depth estimation is primarily classified into monocular and binocular depth estimation. Different estimation methods produce different search models. Currently, the depth estimation architecture search is based on single images. However, depth estimations implementing multi-image information can exploit more spatial information. Therefore, multi-image-based depth estimation architecture search is a promising research direction in the field of depth estimation in the future.

5.2.4 *Neural architecture search for video analysis*

Different from rapid development towards image data, NAS on video data is still an under-explored area and only several video tasks are studied, including action recognition, super resolution and pose estimation. Existing methods mainly focus on introducing successful experiences from image data and further exploit spatio-temporal cues and motion information in video data.

For action recognition, Peng et al. [198] first proposed a NAS method for 3D models to achieve design automation. Specifically, it uses the pseudo 3D operator to process spatial and temporal features in the search space. To further exploit spatio-temporal relationships, Piergiovanni et al. [199] proposed EvaNet by introducing an inflated temporal gaussian mixture (iTGM) to the search space, which enables the model to catch the spatial and temporal interactions among feature flows. In addition, Ryoo et al. [200] established AssembleNet to consider object motion information in design automation. In particular, it first builds a two-stream model as directed graphs and then uses evolutionary algorithms to establish connections between different blocks on RGB and optical flow input at different temporal resolutions. This can better exploit appearance and motion information from videos. Unlike the previous methods, Wang et al. [201] considered introducing an attention mechanism and proposed AttentionNAS, which builds a spatio-temporal attention cell search space and enables generated models to catch long-distance dependencies in video data. Additionally, Piergiovanni et al. [202] focused on improving the computation efficiency of video models and proposed TinyVideoNet, which introduces model running time into the reward loss function and guides the search strategy to generate a desired model with low computing latency. For video super resolution, Liu et al. [203] proposed EVSRNet to achieve high fidelity results and efficient computation. Specifically, it uses the residual block as the basic building block, and then similarly introduces the fidelity of results and computation cost of candidate models into the reward loss function. After that, a gradient descent method is performed to search the optical number and size of the residual blocks. Consequently, the generated model can produce more accurate details while keeping lower computation costs and fewer model parameters. For video pose estimation, Xu et al. [204] proposed ViPNAS to utilize pose relationships between adjacent frames. Particularly, it established the search space by considering the correlation information between adjacent frames and then performed feature fusion on the heatmaps of the previous and current frames via a series of optional operations. Thus, the model can automatically learn the best fusion operation and the best stage to fuse.

5.2.5 *Neural architecture search for embedded devices*

Although current NAS methods can generate high-precision models, these models are often not applicable in real-world intelligent robots due to unacceptable computing latency. This is because real-world robots are usually built on embedded devices, which can only provide limited memory and computing resources. However, current NAS methods do not account for these important factors. Therefore, designing a suitable NAS method according to the characteristics and requirements of embedded devices is an urgent problem to be solved.

On embedded devices, computing latency and memory consumption of models are two key factors. To optimize the computing latency, Cai et al. [205] developed ProxylessNAS to model the computing latency of models as a continuous function and optimized it as a regularization loss to find a model with low latency. Similarly, Wu et al. [206] established DANS, which uses the latency of each block to estimate the latency of the entire model and introduces a latency reward loss to guide the search strategy. López et al. [207] introduced E-DNAS to use a multi-objective differentiable loss function combining classification accuracy and minimum latency on the feature map. Luo et al. [208] proposed LightNAS with a two-step procedure, which first applies a large-scale and one-time search for models that satisfy the latency constraints and then iteratively selects the candidate with the best accuracy. To optimize the memory consumption, Cassimon et al. [209] proposed introducing two soft constraints (cache and performance) and two hard constraints (memory cost and latency) into the reward loss function, which can guide the search strategy to find a model that meets resource requirements. In addition, Wan et al. [210] developed DMaskingNAS with an efficient masking mechanism for feature reuse and effective shape propagation, drastically expanding the search space by supporting searches over spatial and channel dimensions.

In addition to building models from scratch, another direction to consider is how to automatically compress an existing large model. He et al. [211] first proposed automated model compression (AMC) to achieve automated model pruning by using reinforcement learning. Specifically, AMC models the pruning rate and parameter-related information of each layer as the action space and state space, respectively. Then it uses DDPG [212] to train the agent to automatically determine the pruning rate of each layer. Motivated by AMC, Gupta et al. [213] developed PuRL to provide rewards at each pruning step, achieving sparsity and accuracy comparable to state-of-the-art (SOTA) methods with a shorter training cycle. Yu et al. [214] proposed introducing topological information into the model compression procedure, finding the optimal compression ratio while ensuring model accuracy instead of relying solely on the local importance of parameters. To consider the relationship between convolutional filters and channels, Wang et al. [215] established MCTS-RL to prune unnecessary filters before channel pruning, effectively reducing the search space and making channel pruning ratio searching easier. In addition to network pruning, tensor decomposition [216], data quantization [217] and knowledge distillation [218] are other effective techniques for model compression. We do not discuss them here because they rely on hand-crafted design and expert experience and are unrelated to the topic of design automation. Interested readers can refer to [219–222] for further investigation.

5.3 Summary

To summarize, neural architecture search (NAS) has been widely applied in design automation for vision systems, which can automatically search for neural networks and offer improved performances in various vision tasks. In this section, we provide a brief overview from different angles to illustrate the connection and difference between the methods reviewed in this section, as displayed in Fig. 10.

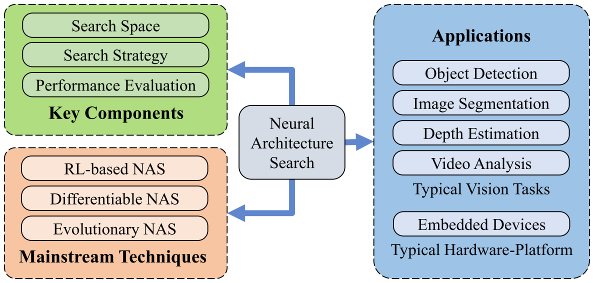


Figure 10 A summary of design automation for vision systems with NAS

Existing works in NAS mainly focus on three key components: (1) the search space [176, 183, 196, 201, 204, 210, 211, 215], which contains all network architecture candidates to be chosen, (2) the search strategy [175, 177, 181, 182, 192, 193, 195, 198–200], guiding how to select a good candidate that meets a specific requirement from the search space, and (3) the performance evaluation [197, 202, 203, 205–209, 213, 214], which generates a performance matrix of a candidate and provides guidance information for the search strategy.

From the view of applied techniques, existing works primarily lie in three categories: RL-based NAS [148–151], differentiable NAS [152, 153] and evolutionary NAS [155–157]. Although RL-based NAS methods can achieve superior performance, they often require thousands of GPUs performing several days even on a median-scale dataset. Differentiable NAS methods are usually more efficient than RL-based methods. However, they often find ill-conditioned architectures due to improper gradient-based optimization. Because evolutionary NAS methods are insensitive to local minima and do not require gradient information, they have shown promising characteristics in solving complex non-convex optimization problems [223], even when the objective function’s mathematical form is unknown [224].

Regarding applications, existing studies typically either incorporate specific prior information into the construction of a NAS method or tackle some special issues within a particular visual task. Taking binocular depth estimation as an example, existing works [195–197] are proposed to preset the network architecture as a stereo matching network and search for the internal structures. For embedded devices, since memory cost and computation latency are highly considered in practical applications, existing works [205–209] evaluate these two factors during searching and encode them in the rewarding functions. In this way, the proposed NAS method can automatically generate a reasonable network with low memory cost and latency.

6 Integrated design automation for the “body-brain-eye” of intelligent robots

At present, most studies separately design the morphologies, controllers and vision systems of intelligent robots. However, strong couplings exist between the designs of the morphologies, controllers, and vision systems [225]. Therefore, it is necessary to consider the integrated design relationship of morphologies, controllers, and vision systems of intelligent robots. These strong coupling relationships are also reflected in nature. According to the law of "survival of the fittest" in biological evolution, many creatures have evolved a large diversity of eye structures and corresponding body morphologies. For example, the morphologies of birds and primates are very different, and their eye locations on the face are also different. It is believed that their brains' mechanisms of processing visual information are also quite different. It is notable that through the cooperation of biological populations, the perception of individual organisms can be further improved [226]. If studies in intelligent robots can automate the design of morphologies, controllers and vision systems, such as in biological evolution, intelligent robots with significantly improved performance may be developed.

Qiao et al. [227–229] took the lead in introducing a "hand-eye-brain" system of intelligent robots that imitates the mechanism, structure and function of the human brain, nervous system, and body motor system. In their proposed method, the role of the "hand" is the motion control of the intelligent robots. Inspired by the "muscle-tendon-bone" organization, Qiao et al. [230] established a control framework based on synergistic activation of muscles and an "attractive region in environment" theory [231, 232]. This framework enabled high-precision flexible operation under low-precision morphologies and low-precision sensors. The role of "eyes" is to construct the visual cognitive system of intelligent robots. Inspired by the brain-inspired visual cognition and memory mechanism of the hippocampus, Qiao et al. [233–235] established a new visual recognition framework, ensuring that intelligent robots can achieve higher recognition accuracy and faster recognition speed. The role of the "brain" is the decision-making of intelligent robots. Inspired by the brain's nervous system, Qiao et al. [236, 237] introduced a brain-inspired motor decision model based on emotion regulation modulation. This model implemented high-level decision-making with an "accuracy-efficiency-speed" balance. Compared with the traditional robot design method, the proposed "hand-eye-brain" system of intelligent robots realizes human-like manipulation with high precision, flexibility, and robustness.

Inspired by Qiao et al.'s "Hand-Eye-Brain" system of intelligent robots, this paper proposes an integrated "Body-Brain-Eye" design automation for intelligent robots, as illustrated in Fig. 11. Specifically, this paper proposes the integrated MODENA framework for automatically designing the morphologies, controllers, and vision systems of intelligent robots, inspired by the evolution of

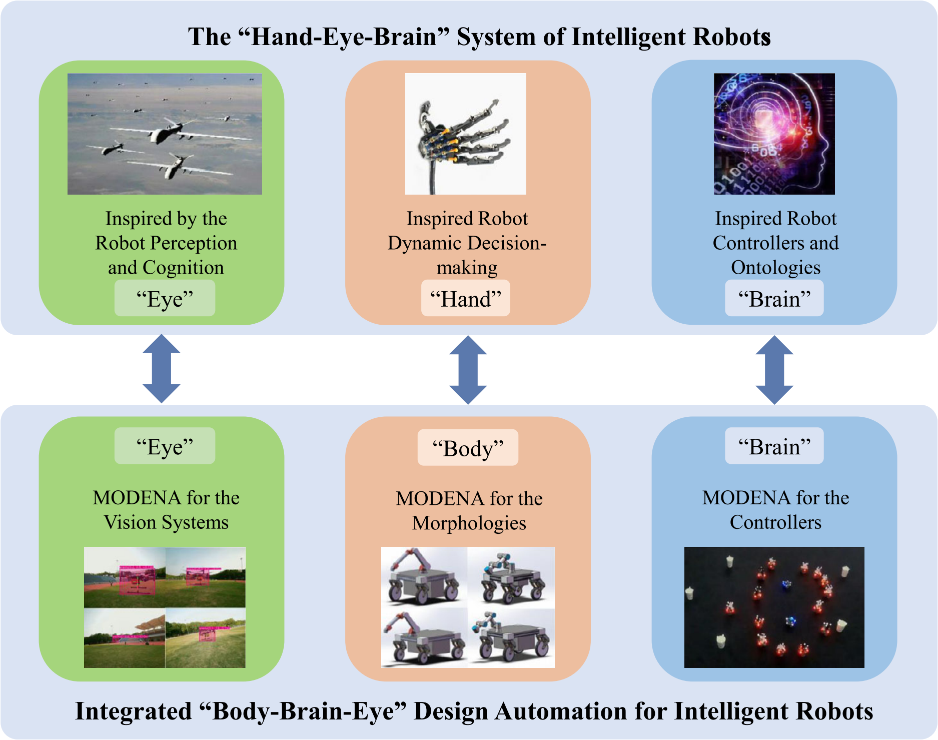

The "Hand-Eye-Brain" System of Intelligent Robots
Inspired by the Robot Perception and Cognition
"Eye"
Inspired Robot Dynamic Decision-making
"Hand"
Inspired Robot Controllers and Ontologies
"Brain"
"Eye"
MODENA for the Vision Systems
"Body"
MODENA for the Morphologies
"Brain"
MODENA for the Controllers
Integrated "Body-Brain-Eye" Design Automation for Intelligent Robots

Figure 11 "Hand-Eye-Brain" system of intelligent robots *vs* integrated "Body-Brain-Eye" design automation for intelligent robots

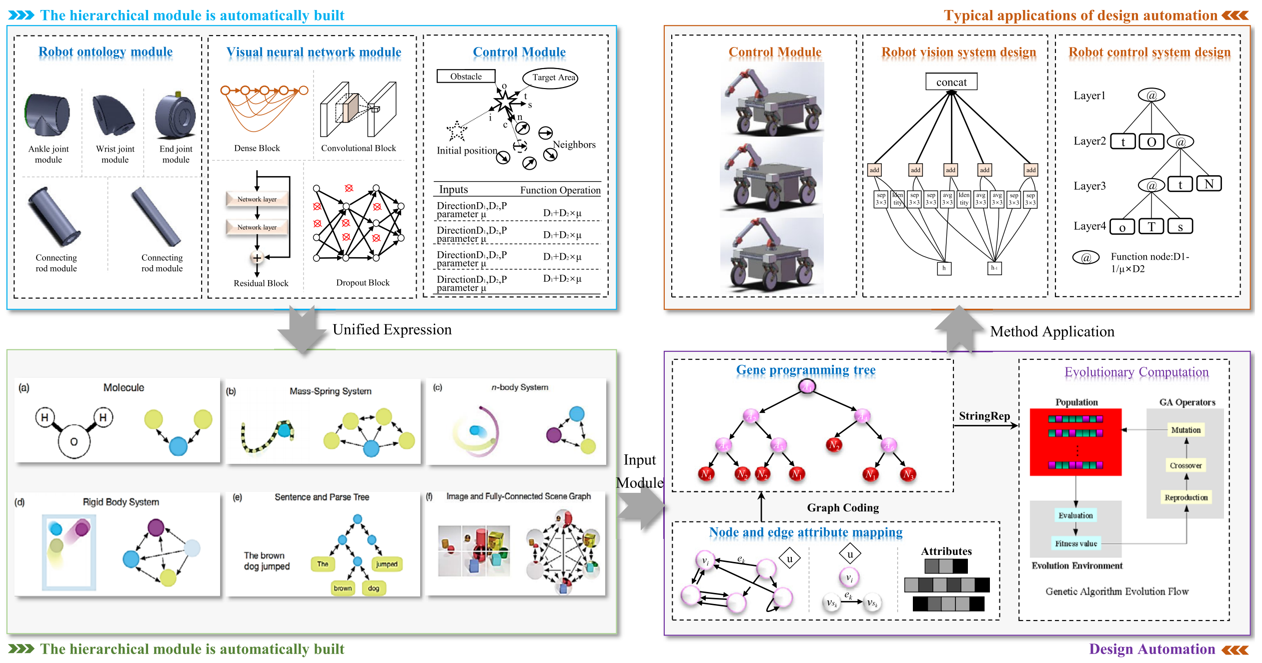


Figure 12 The proposed MODENA for the morphologies, controllers, and vision systems of intelligent robots

biological forms as displayed in Fig. 12. By constructing a modular graph model for the morphologies, controllers, and vision systems of intelligent robots under digital twin architectures and by applying powerful capabilities of genetic programming, evolutionary computation, deep learning, reinforcement learning, and causal reasoning in optimization, decision-making, and reasoning, the MODENA framework can achieve the purpose of obtaining innovative and optimal designs of intelligent robots.

In the process of applying MODENA to design the morphologies, controllers, and vision systems of intelligent robots, the construction of a modular graph model is a fundamental task. These modules are selected or designed according to the application scopes, operating characteristics, and functionalities of the designed robotic systems to meet pre-defined design specifications. In the mechanical field, the modular graph model contains running modules, link modules, joint modules, and end-effector modules, which are used to build the morphologies of intelligent robots [238]. For the image processing part, the modular graph models may contain convolutional layers, pooling layers, and fully connected layers, among others, which are components of a deep neural network architecture that can be used to construct the vision systems of intelligent robots [239]. In the control field, the modular graph model contains main control units, actuators, detecting units, among others, which build the controller of intelligent robots [240]. For the control of swarm robots, the modular graph model contains basic network motifs that can be employed to automatically construct gene regulatory network (GRN) models. A multi-objective genetic programming method can be applied to optimize the structure and parameters of the GRN-based model in parallel so that the behavior of swarm robots can be controlled [33].

7 Problems and prospects

7.1 Existing problems

In this section, we will explore the various aspects that should be considered in the integrated design automation for the “Body-Brain-Eye” of intelligent robots. These include modeling, optimization, knowledge extraction, environment perception, swarm robots, and generalization in unseen scenarios.

(1) Unified Modeling for the “Body-Brain-Eye” of Intelligent Robots

Since intelligent robots are typically multi-energy domain physical systems [241, 242], we need to build a unified graph model to facilitate the design automation process. However, for different categories of intelligent robot modules, we still need to use different modeling tools. For example, we use a geometric model or a bond graph for morphologies, a finite state machine or a model predictive controller for controllers, a gene regulatory network for swarm control, and a deep neural network model for vision systems. Although all these models can be abstracted to a graph model, they are still different modeling languages. Different parts of the graph models need to be decoded separately to obtain the complete intelligent robot. On the other hand, various modules within the intelligent robots are usually coupled with each other, and it is still challenging to express this coupling relationship through a unified graph model.

(2) Efficient Methods for Solving Robot Optimization Problems

In the integrated "Body-Brain-Eye" design automation process of intelligent robots, various types of decision variables (e.g., continuous variables, discrete variables) [243] are included, along with various types of optimization objectives and constraints with different difficulty types [46]. The calculation of objectives or constraints is usually time-consuming [244], and in most cases, external simulators need to be called, making it a computationally expensive optimization problem. Therefore, efficiently solving these constrained multi-objective optimization problems with mixed decision variables and expensive fitness evaluation is a challenging task.

(3) Knowledge Extraction during the Design Process

The optimization of integrated "Body-Brain-Eye" systems of intelligent robots in various experimental scenarios generates a vast amount of data, including intermediate data that contain crucial design knowledge as well as optimization-related knowledge [245]. To extract knowledge and rules from the data with good interpretability, genetic programming methods are effective. However, their accuracy is limited. Deep learning methods, on the other hand, offer high model accuracy, but their black-box characteristics pose a problem for model interpretability. A crucial challenge in creating an iterative optimization system with feedback is to identify causal relationships within and between modules of an intelligent robot to gain innovative design knowledge automatically.

(4) Multi-modal Information Fusion for Environment Perception

The working environments faced by intelligent robots are often complex and varied. Therefore, intelligent robots should have the capability to learn actively and continuously optimize their systems during operations, make efficient and accurate judgments, and respond quickly and appropriately in complex and dynamic working environments. Hence, solving the problem of combining multimodal architecture search and active vision technology to endow robots with the ability to integrate multi-sensor information and actively optimize their hardware system in real-time is essential.

(5) Design Automation for Swarm Robots

The control of swarm robots is witnessing rapid progress in applications, which can be divided into two categories: centralized control and decentralized control. Centralized control is a natural and widely accepted approach, but it faces many challenges when the size of the swarm increases to a certain level. For example, a large system with centralized control has insufficient fault tolerance. Failures of just a few individuals may lead to the failure of the whole system's functionality. Computational costs may also increase dramatically, making it difficult to react to unexpected factors timely. As a result, decentralized control has received increasing attention recently and has gradually become a new mainstream. The key idea here is to design a proper (and in most cases a common) control scheme for each robot in the swarm so that the swarm as a whole can accomplish the specified tasks. It is obviously a challenge to do so, especially when the size of the swarm is large. To address this challenge, design automation approaches play an increasingly important role [246], where MODENA can also contribute greatly [33]. To design and manage such a complex UAV swarm system, the key challenge is to define a rigorous engineering approach to program each robot so that the UAV swarm behaves in a desired manner. How to distill the basic units of the swarm behavior strategy and thus carry out the research on the design automation of unmanned swarm behavior strategy is another emerging issue.

(6) Poor Generalization in Unseen Scenarios

Existing methods mostly automatically generate a model and utilize this fixed model for practical applications. However, this paradigm usually results in unsatisfactory performance, because the model is unable to generalize well to unseen testing scenarios that are different from the training one. Typically, this domain gap between training and testing scenarios is common in real-world applications since the environment is changing all the time, especially for vision systems. Therefore, how to design a NAS method to search for a robust vision model that can perform consistently among different scenes is an urgent issue to be addressed.

7.2 Future directions

Although significant progress has been made in modular design automation over the past two decades, several important issues still need to be addressed, and new application areas are emerging. The following subsections will discuss potential future research directions from two perspectives: theoretical studies and practical applications.

7.2.1 Theoretical studies

(1) Multi-view Unified Modeling of Intelligent Robots

Building a unified model of the morphology, controller and vision system of an intelligent robot is an effective approach to facilitate the design of automation processes. Currently, morphology, controller and vision systems are usually represented by different modeling tools and composed of various modules [241, 242]. For example, when designing a mechanical system, the modeling language might be a bond graph. When designing an unmanned swarm controller, the modeling language may be a finite state machine or a gene regulatory network. When designing a vision system, the modeling language may be a deep neural network. Different modeling languages have different application scopes and characteristics, and it is challenging to capture the coupling relationships among the modules represented by them. Therefore, constructing a multi-view unified modeling tool that can represent the morphology, controller and vision systems effectively and efficiently is an essential direction for the design automation of intelligent robots.

(2) Surrogate-assisted Constrained Multi-objective Optimization for Intelligent Robots

The optimization of intelligent robots often requires the simultaneous consideration of multiple conflicting design objectives and a large number of constraints. In addition, the calculations of objectives and constraints are usually time-consuming and often require the invocation of external simulation software. Therefore, the optimization problem of an intelligent robot can be defined as an expensive constrained multi-objective optimization problem [46, 247]. In the research of MODENA for intelligent robots, constrained multi-objective evolutionary algorithms are gradually becoming a popular approach to solve the above multi-objective optimization problems. In the study of constrained multi-objective evolutionary algorithms, the conventional view is that each infeasible region is equally important. Only the constraints represented by infeasible regions close to the unconstrained Pareto front affect the true Pareto front. Therefore, how to take advantage of features like this to deal with the contradiction among convergence, diversity and feasibility has become a major consideration in designing constrained multi-objective optimization algorithms. In terms of surrogate models, considering an adaptive surrogate model approach by combining global and local surrogate models for optimization objectives and constraints to establish novel constrained multi-objective evolutionary algorithms is another direction worthy of in-depth investigation in the future.

(3) Knowledge Extraction in Design Automation

The knowledge extracted in the design automation process of intelligent robots involves both explicit knowledge and implicit knowledge. Explicit knowledge is also called human knowledge, which can often be directly understood by human experts and has very good interpretability. On the other hand, implicit knowledge is usually not directly understandable by humans, but can be stored and inferred by machines. Thus, it is also called machine knowledge, which has the potential to be understood by humans one day in the future. Symbolic regression, a method based on genetic programming, is usually used for explicit knowledge mining. This method can automatically mine the explicit knowledge contained in the data by manually defining a set of functions and terminals using prior knowledge of the problem domain. Causal reasoning, an emerging research field, can also be used to obtain explainable knowledge. This can, in turn, guide the search for expensive constrained multi-objective evolutionary algorithms and the adjustment of the problem formulation of the intelligent robot optimization problems.

(4) MODENA for Intelligent Robots Based on Digital Twins

MODENA for intelligent robots necessitates numerous simulations and experiments in both virtual and real-world environments. These efforts can be significantly expedited through the application of emerging digital twin technology. This technology creates a unique type of metaspace that replicates the physical laws of space with exceptional precision and is a subset of the Metaverse. Traditional methods of designing intelligent robots typically involve a laborious and time-consuming trial-and-error process. Conversely, implementing the digital twin approach enables the faithful mapping of the robot from real space to virtual space in four dimensions: geometry, contact dynamics, behavior, and rules. Moreover, it allows for the arbitrary adjustment of the morphology, controller, and vision systems, generating practically unlimited design candidates whose optimization is efficiently supported through the integration of the powerful capabilities of machine learning and evolutionary computing. Additionally, machine learning can be used to mine knowledge and rules from data generated during the design process, which can then be utilized in future design activities. Therefore, exploring how to fully harness the power of digital twin technology for MODENA is another crucial area of study.

(5) Domain Adaptation and Generalization in Design Automation

Since the poor generalization ability of designed models to unseen scenarios is an urgent issue, especially for vision systems, model robustness becomes an important factor when designing a NAS method. A promising solution is to introduce domain adaptation [248] and generalization [249] evaluation in the design procedure, which focuses on transferring learned knowledge in training scenarios to unseen testing ones. Specifically, we can introduce an additional evaluation matrix for generalization ability for candidate model selection. In this way, the search strategy can choose a model with a particular trade-off that can achieve good performance and be robust to noisy and varied environments. There are a few works [250–252] that concentrate on this appealing direction.

(6) Active and Continual Learning in Design Automation

In addition to designing a robust model, another solution to tackle the poor generalization issue is to perform online learning in testing scenarios, which can allow the model to quickly adjust its parameters and adapt to unknown environments. To achieve effective online learning, there are two key problems to be solved. First, the model needs to figure out what to learn in a given environment. To address this, active vision and learning [253, 254] can guide models to explore valuable targets and learn superior decision-making behaviors, as studied in different applications, including robot exploration [255–257], unmanned aerial vehicle (UAV) swarm localization, and other tasks [258–260]. Second, the model needs to overcome the catastrophic forgetting issue during online learning. Specifically, when the model learns new knowledge in a new scene, the previously learned knowledge will be dramatically forgotten, leading to a severe overfitting issue and making the model harder to generalize to another unseen scene. To tackle this issue, continual learning [261] has been proposed to guide models to continually learn over time by accommodating new knowledge while retaining previously learned experiences. Several works [262–264] have tried to introduce continual learning in NAS.

To summarize, it is important and desirable for a model to automatically optimize itself and adapt to varied and unseen scenarios, achieving higher levels of intelligence. This remains as an open and attractive problem in the design automation of intelligent robotic vision systems.

7.2.2 Practical applications

In this section, we present some exemplary scenarios to illustrate the potential benefits of applying MODENA. For instance, power plants serve as the cornerstone of the power system, and their operational health plays a crucial role in ensuring the system's safety. The intricate layout of pipelines in power plants makes manual inspection challenging. Moreover, manual inspection is vulnerable to problems such as missed inspections, false inspections, and concerns about the personal safety of inspectors, which can be influenced by various factors, such as labor intensity and weather conditions. With the advent of heterogeneous unmanned swarm technology, the integration of flying inspection robots, ground inspection robots, and pipeline leak-detecting and repairing robots has become technically feasible, offering significant advantages over manual inspections. This integration may also become a hot research theme in the future, as illustrated in Fig. 13. Therefore, we suggest considering the following prospects for future research in this paper.

(1) Design Automation for the Morphologies of Unmanned Swarm Systems

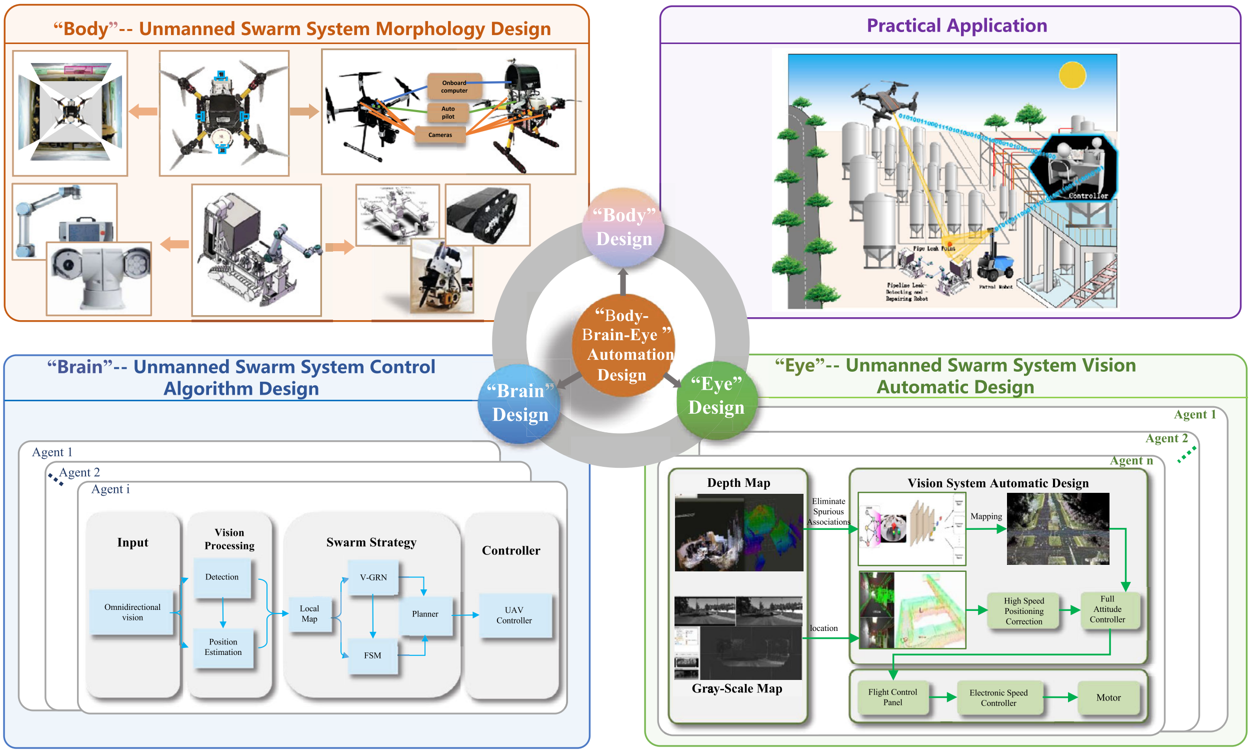


Figure 13 The integrated "Body-Brain-Eye" design automation for heterogeneous unmanned swarm system of inspection and repair robots used in the power plant

Unmanned swarm systems need to accomplish multiple tasks, and the relationship between their overall performance and component properties is extremely complex, which makes the morphological structure design of unmanned swarm systems complicated. The MODENA method provides a new idea for the morphological structure design of unmanned swarm systems. For example, in the power plant environment, the unmanned swarm contains flying inspection robots, ground inspection robots and pipeline leak-detecting and repairing robots. Their morphologies differ greatly, with different application scopes, operation characteristics, functions and performances. Designing corresponding morphological models based on the application scopes, operational characteristics, functions, and performances of various robots within unmanned swarm systems to achieve superior overall performance is a challenge and a focus of future research.

(2) Design Automation for the Environmental Perception and Cognitive Systems of Unmanned Swarm Systems

In complex environments with high dynamics, uncertainty and resource constraints, unmanned swarm systems need to achieve distributed sensing and cognition of the environment through multi-modal interaction techniques. For example, in the power plant environment, aerial inspection robots equipped with vision sensors, ground inspection robots equipped with high-precision LIDAR and vision sensors, and ground repair robots equipped with infrared vision sensors work together to achieve rapid and precise localization of power plant faults and timely repair through data obtained from different sensors. It is crucial to design a proper model that can process heterogeneous sensor data to achieve efficient perception and cognition of complex environments. Therefore, research on the design automation of the visual perception model is an important direction for distributed environment perception and cognition.

(3) Design Automation for the Controllers of Unmanned Swarm Systems

Because the environment faced by an unmanned swarm system is uncertain or unpredictable, it is difficult to design algorithms that can control swarm behaviors based on accurate models. In swarm control, the biggest challenge is to design a proper control scheme for each robot so that the swarm as a whole can generate collective behavior that can accomplish the pre-defined task for the swarm. Since each robot in the swarm follows the same behavioral model, designing controllers using traditional methods faces great difficulty. Design automation methods can play a significant role here since they can generate and explore a large number of potential candidates with the help of digital twin technology. Design automation techniques can also identify the optimal ones that satisfy the specified task requirements more efficiently by using metaheuristic methods, such as evolutionary computation. Therefore, the design automation of behavioral control strategies for UAV swarms based on evolutionary algorithms is another important research direction.

8 Conclusion

In this paper, we present a comprehensive survey of MODENA for designing the morphologies, controllers and vision systems of intelligent robots. Given the increasing complexity of working environments and the diversification of tasks, there is a growing need for MODENA to design the "Body-Brain-Eye" of intelligent robots. In the MODENA approach, the robot system's morphology, controller, and vision systems can all be expressed as a graphical model. By automatically exploring the design space of the graphical model, a set of design candidates of intelligent robots that satisfy pre-defined functions and requirements can be obtained. The key components in MODENA include surrogate-assisted constrained multi-objective evolutionary algorithms (CMOEAs), topological search algorithms such as genetic programming, neural architecture search, and techniques for knowledge extraction during the design process, among others. MODENA is a core technology that can significantly improve the design efficiency and performance of robots, and it will become an increasingly important research theme in the future for designing either individual or swarm robots, just as EDA has played an important role in both academia and industry.

Acknowledgements
The authors would like to express their sincere gratitude to the International Cooperation Base of Evolutionary Intelligence and Robotics of Guangdong Province, China, and the Key Laboratory of Intelligent Manufacturing Technology (Shantou University), Ministry of Education, China, for their invaluable support and assistance in this study.

Funding
This research was supported in part by National Key R&D Program of China (No. 2021ZD0111501), National Natural Science Foundation of China (No. 62176147), Science and Technology Planning Project of Guangdong Province of China (Nos. 2021A0505030072 and 2022A1515110660), Science and Technology Special Funds Project of Guangdong Province of China (Nos. STKJ2021176 and STKJ2021019), and STU Scientific Research Foundation for Talents (Nos. NTF21001 and NTF22030).

Abbreviations
AI, Artificial Intelligence; AMC, Automated Model Compression; BG, Bond Graph; BGGP, Bond Graph and Genetic Programming; CAE, Computer Aided Engineering; CMOEAs, Constrained Multi-objective Evolutionary Algorithms; DARTS, Differentiable Architecture Search; EASNet, Elastic and Accurate Network for Stereo Matching; EDA, Electronic Design Automation; FPN, Feature Pyramid Network; FSA, Finite State Automaton; GP, Genetic Programming; GRN, Gene Regulatory Network; HBG, Hybrid Bond Graph; iTGM, Inflated Temporal Gaussian Mixture; ITO, Isogeometric Topology Optimization; LSTM, Long Short-term Memory; MDA, Mechatronic Design Automation; MODENA, Modular Design Automation; MPC, Model Predictive Control; NAS, Neural Architecture Search; NEAT, the Neuroevolution of Augmenting Topologies; PID, Proportional-Integral-Derivative; SOTA, State-of-the-art; UAVs, Unmanned Aerial Vehicles.

Availability of data and materials
The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Declarations

Competing interests

The authors declare no competing interests.

Author contributions

All authors contributed to the study's conception and design. The idea for the article was performed by ZH and ZF. The literature search and data analysis were performed by RM, PR, QZ, YZ and NX. The first draft of the manuscript was written by WL, JFZ and ZW. ZF, BX and LG revised the manuscript. In addition, all authors commented on previous versions of the manuscript, and read and approved the final manuscript.

Author details

[1]International Cooperation Base of Evolutionary Intelligence and Robotics, Shantou, Guangdong, China. [2]Department of Electronic Information Engineering, Shantou University, Shantou 515063, China. [3]School of Automation, Beijing Institute of Technology, Beijing, 100081, China. [4]School of Mechanical Science & Engineering, HUST, Huazhong University of Science and Technology, Wuhan, 430074, China. [5]College of Science, Shantou University, Shantou, 515063, China.

Received: 1 November 2022 Revised: 29 December 2022
Accepted: 1 March 2023 Published online: 08 May 2023

References

1. Reynolds, M. F., Cortese, A. J., Liu, Q., Zheng, Z., Wang, W., Norris, S. L., et al. (2022). Microscopic robots with onboard digital control. *Science Robotics*, *7*(70), eabq2296.
2. Billard, A., & Kragic, D. (2019). Trends and challenges in robot manipulation. *Science*, *364*(6446), eaat8414.
3. Macenski, S., Foote, T., Gerkey, B., Lalancette, C., & Woodall, W. (2022). Robot operating system 2: design, architecture, and uses in the wild. *Science Robotics*, *7*(66), eabm6074.
4. Honarpardaz, M., Ölvander, J., & Tarkian, M. (2019). Fast finger design automation for industrial robots. *Robotics and Autonomous Systems*, *113*, 120–131.
5. Lipson, H. (2005). Evolutionary robotics and open-ended design automation. In Y. Bar-Cohen (Ed.), *Biomimetics—biologically inspired technologies* (pp. 147–174). Boca Raton: CRC Press.
6. Hoebert, T., Lepuschitz, W., Vincze, M., & Merdan, M. (2021). Knowledge-driven framework for industrial robotic systems. *Journal of Intelligent Manufacturing*, *34*(2), 771–788.
7. Ramos, F., Vázquez, A. S., Fernández, R., & Olivares-Alarcos, A. (2018). Ontology based design, control and programming of modular robots. *Integrated Computer-Aided Engineering*, *25*(2), 173–192.
8. Short, M., & Burn, K. (2011). A generic controller architecture for intelligent robotic systems. *Robotics and Computer-Integrated Manufacturing*, *27*(2), 292–305.
9. Armangué Quintana, X. (2003). *Modelling stereoscopic vision systems for robotic applications*. PhD thesis, Universitat de Girona.
10. Diveev, A., & Sofronova, E. (2019). Automation of synthesized optimal control problem solution for mobile robot by genetic programming. In *Proceedings of SAI intelligent systems conference* (pp. 1054–1072). Berlin: Springer.
11. Alattas, R. J., Patel, S., & Sobh, T. M. (2019). Evolutionary modular robotics: survey and analysis. *Journal of Intelligent & Robotic Systems*, *95*(3), 815–828.
12. Pierson, H. A., & Gashler, M. S. (2017). Deep learning in robotics: a review of recent research. *Advanced Robotics*, *31*(16), 821–835.
13. Singh, B., Kumar, R., & Singh, V. P. (2022). Reinforcement learning in robotic applications: a comprehensive survey. *Artificial Intelligence Review*, *55*(2), 945–990.
14. Hellström, T. (2021). The relevance of causation in robotics: a review, categorization, and analysis. *Paladyn, Journal of Behavioral Robotics*, *12*(1), 238–255.
15. Lipson, H., & Pollack, J. B. (2000). Automatic design and manufacture of robotic lifeforms. *Nature*, *406*(6799), 974–978.
16. Kwiatkowski, R., & Lipson, H. (2019). Task-agnostic self-modeling machines. *Science Robotics*, *4*(26), eaau9354.
17. Schmidt, M., & Lipson, H. (2009). Distilling free-form natural laws from experimental data. *Science*, *324*(5923), 81–85.
18. Zykov, V., Mytilinaios, E., Adams, B., & Lipson, H. (2005). Self-reproducing machines. *Nature*, *435*(7039), 163–164.
19. Fan, Z., Seo, K., Hu, J., Goodman, E. D., & Rosenberg, R. C. (2004). A novel evolutionary engineering design approach for mixed-domain systems. *Engineering Optimization*, *36*(2), 127–147.
20. Xu, P., Wei, Z., Guo, Z., Jia, L., Han, G., Si, C., Ning, J., & Yang, F. (2021). A real-time circuit phase delay correction system for MEMS vibratory gyroscopes. *Micromachines*, *12*(5), Article No. 506.
21. Krylov, G., Kawa, J., & Friedman, E. G. (2021). Design automation of superconductive digital circuits: a review. *IEEE Nanotechnology Magazine*, *15*(6), 54–67.
22. Gongora, A. E., Xu, B., Perry, W., Okoye, C., Riley, P., Reyes, K. G., Morgan, E. F., & Brown, K. A. (2020). A Bayesian experimental autonomous researcher for mechanical design. *Science Advances*, *6*(15), eaaz1708.
23. Sneineh, A. A., & Salah, W. A. (2019). Design and implementation of an automatically aligned solar tracking system. *International Journal of Power Electronics and Drive Systems*, *10*(4), 2055.
24. Wang, J., Fan, Z., Terpenny, J. P., & Goodman, E. D. (2005). Knowledge interaction with genetic programming in mechatronic systems design using bond graphs. *IEEE Transactions on Systems, Man and Cybernetics. Part C, Applications and Reviews*, *35*(2), 172–182.
25. Behbahani, S., & de Silva, C. W. (2008). System-based and concurrent design of a smart mechatronic system using the concept of mechatronic design quotient (MDQ). *IEEE/ASME Transactions on Mechatronics*, *13*(1), 14–21.
26. Dupuis, J.-F., Fan, Z., & Goodman, E. D. (2012). Evolutionary design of both topologies and parameters of a hybrid dynamical system. *IEEE Transactions on Evolutionary Computation*, *16*(3), 391–405. https://doi.org/10.1109/TEVC.2011.2159724.
27. Garattoni, L., & Birattari, M. (2018). Autonomous task sequencing in a robot swarm. *Science Robotics*, *3*(20), eaat0430.
28. Fan, Z. (2010). *Mechatronic design automation: an emerging research and recent advances*. New York: Nova Science Publishers.
29. Fan, Z., Wang, J., & Goodman, E. (2004). Exploring open-ended design space of mechatronic systems. *International Journal of Advanced Robotic Systems*, *1*(4), 295–302.
30. Lindsay, G. W. (2021). Convolutional neural networks as a model of the visual system: past, present, and future. *Journal of Cognitive Neuroscience*, *33*(10), 2017–2031.
31. Qian, Y., Chen, Z., & Wang, S. (2021). Audio-visual deep neural network for robust person verification. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, *29*, 1079–1092.
32. Cai, Y., Li, H., Fan, Z., Hong, J., Xu, P., Cheng, H., Zhu, X., Hu, B., & Hao, Z. (2022). *VGSwarm: a vision-based gene regulation network for UAVs swarm behavior emergence*. arXiv preprint arXiv:2206.08669.
33. Fan, Z., Wang, Z., Zhu, X., Hu, B., Zou, A., & Bao, D. (2019). *An automatic design framework of swarm pattern formation based on multi-objective genetic programming*. arXiv preprint arXiv:1910.14627.
34. Li, J., & Tan, Y. (2019). A probabilistic finite state machine based strategy for multi-target search using swarm robotics. *Applied Soft Computing*, *77*, 467–483.
35. Xu, G., Ding, H., & Feng, Z. (2019). Optimal design of hydraulic excavator shovel attachment based on multiobjective evolutionary algorithm. *IEEE/ASME Transactions on Mechatronics*, *24*(2), 808–819.
36. Hsiao, J. C., Shivam, K., Chou, C. L., & Kam, T. Y. (2020). Shape design optimization of a robot arm using a surrogate-based evolutionary approach. *Applied Sciences*, *10*(7), 2223.
37. Datta, R., & Deb, K. (2011). Multi-objective design and analysis of robot gripper configurations using an evolutionary-classical approach. In *Proceedings of the 13th annual conference on genetic and evolutionary computation* (pp. 1843–1850). New York: ACM.
38. Datta, R., Pradhan, S., & Bhattacharya, B. (2015). Analysis and design optimization of a robotic gripper using multiobjective genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *46*(1), 16–26.
39. Rezazadeh, S., & Hurst, J. W. (2014). On the optimal selection of motors and transmissions for electromechanical and robotic systems. In *2014 IEEE/RSJ international conference on intelligent robots and systems* (pp. 4605–4611). Los Alamitos: IEEE.
40. Murata, S., Yoshida, E., Kamimura, A., Kurokawa, H., Tomita, K., & Kokaji, S. (2002). M-TRAN: self-reconfigurable modular robotic system. *IEEE/ASME Transactions on Mechatronics*, *7*(4), 431–441.

41. Shen, W.-M., Salemi, B., & Will, P. (2002). Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots. *IEEE Transactions on Robotics and Automation*, *18*(5), 700–712.
42. Brandt, D., Christensen, D. J., & Lund, H. H. (2007). ATRON robots: versatility from self-reconfigurable modules. In *2007 international conference on mechatronics and automation* (pp. 26–32). Los Alamitos: IEEE.
43. West, C., Montazeri, A., Monk, S. D., & Taylor, C. J. (2016). A genetic algorithm approach for parameter optimization of a 7DOF robotic manipulator. *IFAC-PapersOnLine*, *49*(12), 1261–1266.
44. Xiao, Y., Fan, Z., Li, W., Chen, S., Zhao, L., & Xie, H. (2016). A manipulator design optimization based on constrained multi-objective evolutionary algorithms. In *2016 international conference on industrial informatics-computing technology, intelligent technology, industrial information integration (ICIICII)* (pp. 199–205). Los Alamitos: IEEE.
45. Hassan, A., & Abomoharam, M. (2017). Modeling and design optimization of a robot gripper mechanism. *Robotics and Computer-Integrated Manufacturing*, *46*, 94–103.
46. Fan, Z., You, Y., Cai, X., Zheng, H., Zhu, G., Li, W., Garg, A., Deb, K., & Goodman, E. (2019). Analysis and multi-objective optimization of a kind of teaching manipulator. *Swarm and Evolutionary Computation*, *50*, 100554.
47. Fan, Z., Li, W., Cai, X., Li, H., Wei, C., Zhang, Q., Deb, K., & Goodman, E. (2019). Push and pull search for solving constrained multi-objective optimization problems. *Swarm and Evolutionary Computation*, *44*, 665–679.
48. Zhang, Z., Zheng, Y., Hu, Z., Liu, L., Zhao, X., Li, X., & Pan, J. (2021). A computational framework for robot hand design via reinforcement learning. In *2021 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 7216–7222). Los Alamitos: IEEE.
49. Hornby, G. S., Lipson, H., & Pollack, J. B. (2003). Generative representations for the automated design of modular physical robots. *IEEE Transactions on Robotics and Automation*, *19*(4), 703–719.
50. Faíña, A., Bellas, F., Souto, D., & Duro, R. J. (2011). Towards an evolutionary design of modular robots for industry. In *International work-conference on the interplay between natural and artificial computation* (pp. 50–59). Berlin: Springer.
51. Faíña, A., Bellas, F., López-Peña, F., & Duro, R. J. (2013). EDHMoR: evolutionary designer of heterogeneous modular robots. *Engineering Applications of Artificial Intelligence*, *26*(10), 2408–2423.
52. Veenstra, F., Faina, A., Risi, S., & Stoy, K. (2017). Evolution and morphogenesis of simulated modular robots: a comparison between a direct and generative encoding. In *European conference on the applications of evolutionary computation* (pp. 870–885). Berlin: Springer.
53. Silva, F., Duarte, M., Correia, L., Oliveira, S. M., & Christensen, A. L. (2016). Open issues in evolutionary robotics. *Evolutionary Computation*, *24*(2), 205–236.
54. Dong, Y., Wang, L., Xia, N., Yang, Z., Zhang, C., Pan, C., Jin, D., Zhang, J., Majidi, C., & Zhang, L. (2022). Untethered small-scale magnetic soft robot with programmable magnetization and integrated multifunctional modules. *Science Advances*, *8*(25), eabn8932.
55. Kelly, J., & Zhang, H. (2006). Combinatorial optimization of sensing for rule-based planar distributed assembly. In *2006 IEEE/RSJ international conference on intelligent robots and systems* (pp. 3728–3734). Los Alamitos: IEEE.
56. Werfel, J. (2006). *Anthills built to order: automating construction with artificial swarms*. PhD thesis, Harvard University.
57. Kang, X., Feng, H., Dai, J. S., & Yu, H. (2020). High-order based revelation of bifurcation of novel Schatz-inspired metamorphic mechanisms using screw theory. *Mechanism and Machine Theory*, *152*, 103931.
58. Dai, J. S., & Rees Jones, J. (1999). Mobility in metamorphic mechanisms of foldable/erectable kinds. *Journal of Mechanical Design*, *121*(3), 375–382.
59. Chai, X., & Dai, J. S. (2019). Three novel symmetric Waldron–Bricard metamorphic and reconfigurable mechanisms and their isomerization. *Journal of Mechanisms and Robotics*, *11*(5), 051011.
60. Zhang, L., Wang, D., & Dai, J. S. (2008). Biological modeling and evolution based synthesis of metamorphic mechanisms. *Journal of Mechanical Design*, *130*(7), 072303.
61. Sun, C., Chen, L., Liu, J., Dai, J. S., & Kang, R. (2020). A hybrid continuum robot based on pneumatic muscles with embedded elastic rods. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *234*(1), 318–328.
62. Meng, L., Kang, R., Gan, D., Chen, G., Chen, L., Branson, D. T., & Dai, J. S. (2020). A mechanically intelligent crawling robot driven by shape memory alloy and compliant bistable mechanism. *Journal of Mechanisms and Robotics*, *12*(6), 061005.
63. Tang, Z., Wang, K., Spyrakos-Papastavridis, E., & Dai, J. S. (2022). Origaker: a novel multi-mimicry quadruped robot based on a metamorphic mechanism. *Journal of Mechanisms and Robotics*, *14*(6), 060907.
64. Wang, R., Song, Y., & Dai, J. S. (2021). Reconfigurability of the origami-inspired integrated 8R kinematotropic metamorphic mechanism and its evolved 6R and 4R mechanisms. *Mechanism and Machine Theory*, *161*, 104245.
65. Fan, Z., Zhu, G., & Li, W. (2020). Mechatronic design automation: a short review. In W. Banzhaf, B. H. C. Cheng, K. Deb, et al. (Eds.), *Evolution in action: past, present and future* (pp. 453–466). Berlin: Springer.
66. Caasenbrood, B., Pogromsky, A., & Nijmeijer, H. (2020). A computational design framework for pressure-driven soft robots through nonlinear topology optimization. In *2020 3rd IEEE international conference on soft robotics (RoboSoft)* (pp. 633–638). Los Alamitos: IEEE.
67. Zhao, Z.-L., Zhou, S., Feng, X.-Q., & Xie, Y. M. (2020). Morphological optimization of scorpion telson. *Journal of the Mechanics and Physics of Solids*, *135*, 103773.
68. Ottaviano, E., Husty, M., & Ceccarelli, M. (2006). Level-set method for workspace analysis of serial manipulators. In J. Lenarcic & B. Roth (Eds.), *Advances in robot kinematics, mechanisms and motion* (pp. 307–314). Berlin: Springer.
69. Ye, D., Sun, S., Chen, J., & Luo, M. (2014). The lightweight design of the humanoid robot frameworks based on evolutionary structural optimization. In *2014 IEEE international conference on robotics and biomimetics (ROBIO 2014)* (pp. 2286–2291). Los Alamitos: IEEE.
70. Lei, X., Liu, C., Du, Z., Zhang, W., & Guo, X. (2019). Machine learning-driven real-time topology optimization under moving morphable component-based framework. *Journal of Applied Mechanics*, *86*(1), 011004.
71. Gao, J., Wang, L., Luo, Z., & Gao, L. (2021). IgaTop: an implementation of topology optimization for structures using IGA in Matlab. *Structural and Multidisciplinary Optimization*, *64*(3), 1669–1700.
72. Gao, J., Xiao, M., Zhang, Y., & Gao, L. (2020). A comprehensive review of isogeometric topology optimization: methods, applications and prospects. *Chinese Journal of Mechanical Engineering*, *33*(6), 24–37.
73. Gao, J., Gao, L., Luo, Z., & Li, P. (2019). Isogeometric topology optimization for continuum structures using density distribution function. *International Journal for Numerical Methods in Engineering*, *119*(10), 991–1017.
74. Wang, Y., Xiao, M., Xia, Z., Li, P., & Gao, L. (2022). From computer-aided design (CAD) toward human-aided design (HAD): an isogeometric topology optimization approach. *Engineering*, *22*(3), 94–105.
75. Gao, J., Xiao, M., Yan, Z., Gao, L., & Li, H. (2022). Robust isogeometric topology optimization for piezoelectric actuators with uniform manufacturability. *Frontiers of Mechanical Engineering*, *17*(2), 205–224.
76. Gao, J., Xue, H., Gao, L., & Luo, Z. (2019). Topology optimization for auxetic metamaterials based on isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, *352*, 211–236.
77. Xu, J., Gao, L., Xiao, M., Gao, J., & Li, H. (2020). Isogeometric topology optimization for rational design of ultra-lightweight architected materials. *International Journal of Mechanical Sciences*, *166*, 105103.
78. Seo, Y.-D., Kim, H.-J., & Youn, S.-K. (2010). Isogeometric topology optimization using trimmed spline surfaces. *Computer Methods in Applied Mechanics and Engineering*, *199*(49–52), 3270–3296.
79. Wang, Y., & Benson, D. J. (2016). Isogeometric analysis for parameterized LSM-based structural topology optimization. *Computational Mechanics*, *57*(1), 19–35.
80. Zhun, F., Jie, Z. G., Ji, L. W., Gen, Y. Y., Ming, L. X., Han, L. P., & Bin, X. (2021). Applications of evolutionary computation in the design automation of complex mechatronic system: a survey. *Acta Automatica Sinica*, *47*(7), 1495–1515.
81. Seo, K., Fan, Z., Hu, J., Goodman, E. D., & Rosenberg, R. C. (2003). Toward a unified and automated design methodology for multi-domain dynamic systems using bond graphs and genetic programming. *Mechatronics*, *13*(8–9), 851–885.
82. Wu, Z., Campbell, M. I., & Fernández, B. R. (2008). Bond graph based automated modeling for computer-aided design of dynamic systems. *Journal of Mechanical Design*, *130*(4), 041102.

83. Li, J., Wang, L., & Yan, B. (2021). Modeling and dynamic analysis of the dynamic stabilization unit based on bond graph. *Archive of Applied Mechanics*, *91*(6), 2681–2695.
84. Tolley, M. T., Hiller, J. D., & Lipson, H. (2011). Evolutionary design and assembly planning for stochastic modular robots. In S. Doncieux, N. Bredèche, & J.-B. Mouret (Eds.), *New horizons in evolutionary robotics* (pp. 211–225). Berlin: Springer.
85. Yim, M., Shen, W.-M., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., & Chirikjian, G. S. (2007). Modular self-reconfigurable robot systems [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, *14*(1), 43–52.
86. White, P., Zykov, V., Bongard, J. C., & Lipson, H. (2005). Three dimensional stochastic reconfiguration of modular robots. In S. Thrun, G. S. Sukhatme, & S. Schaal (Eds.), *Robotics: science and systems I* (pp. 161–168). Cambridge: The MIT Press.
87. Østergaard, E. H., Kassow, K., Beck, R., & Lund, H. H. (2006). Design of the ATRON lattice-based self-reconfigurable robot. *Autonomous Robots*, *21*(2), 165–183.
88. Miki, T., Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., & Hutter, M. (2022). Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, *7*(62), eabk2822.
89. Abadía, I., Naveros, F., Ros, E., Carrillo, R. R., & Luque, N. R. (2021). A cerebellar-based solution to the nondeterministic time delay problem in robotic control. *Science Robotics*, *6*(58), eabf2756.
90. Chen, T., He, Z., & Ciocarlie, M. (2021). Co-designing hardware and control for robot hands. *Science Robotics*, *6*(54), eabg2133.
91. Zhong, H., Hu, C., Li, X., Gao, L., Zeng, B., & Dong, H. (2019). Kinematic calibration method for a two-segment hydraulic leg based on an improved whale swarm algorithm. *Robotics and Computer-Integrated Manufacturing*, *59*, 361–372.
92. Zhong, H., Xie, S., Li, X., Gao, L., & Lu, S. (2022). Online gait generation method based on neural network for humanoid robot fast walking on uneven terrain. *International Journal of Control, Automation, and Systems*, *20*(3), 941–955.
93. Dong, H., Li, X., Shen, P., Gao, L., & Zhong, H. (2021). Interval type-2 fuzzy logic PID controller based on differential evolution with better and nearest option for hydraulic serial elastic actuator. *International Journal of Control, Automation, and Systems*, *19*(2), 1113–1132.
94. Dong, H., Gao, L., Shen, P., Li, X., Lu, Y., & Dai, W. (2019). An interval type-2 fuzzy logic controller design method for hydraulic actuators of a human-like robot by using improved drone squadron optimization. *International Journal of Advanced Robotic Systems*, *16*(6). https://doi.org/10.1177/1729881419891553.
95. Hai, X., Wang, Z., Feng, Q., Ren, Y., Xu, B., Cui, J., & Duan, H. (2019). Mobile robot ADRC with an automatic parameter tuning mechanism via modified pigeon-inspired optimization. *IEEE/ASME Transactions on Mechatronics*, *24*(6), 2616–2626.
96. Cáceres Flórez, C. A., Rosário, J. M., & Amaya, D. (2020). Control structure for a car-like robot using artificial neural networks and genetic algorithms. *Neural Computing & Applications*, *32*(20), 15771–15784.
97. Chin, C. S., & Lin, W. P. (2018). Robust genetic algorithm and fuzzy inference mechanism embedded in a sliding-mode controller for an uncertain underwater robot. *IEEE/ASME Transactions on Mechatronics*, *23*(2), 655–666.
98. Feng, H., Yin, C.-B., Weng, W., Ma, W., Zhou, J., Jia, W., & Zhang, Z. (2018). Robotic excavator trajectory control using an improved GA based PID controller. *Mechanical Systems and Signal Processing*, *105*, 153–168.
99. Sharma, R., Rana, K. P. S., & Kumar, V. (2014). Performance analysis of fractional order fuzzy PID controllers applied to a robotic manipulator. *Expert Systems with Applications*, *41*(9), 4274–4289.
100. Ali, R. S., Aldair, A. A., & Almousawi, A. K. (2014). Design an optimal PID controller using artificial bee colony and genetic algorithm for autonomous mobile robot. *International Journal of Computer Applications*, *100*(16), 8–16.
101. Taherkhorsandi, M., Mahmoodabadi, M. J., Talebipour, M., & Castillo-Villar, K. K. (2015). Pareto design of an adaptive robust hybrid of PID and sliding control for a biped robot via genetic algorithm optimization. *Nonlinear Dynamics*, *79*(1), 251–263.
102. Zhenlu, S., Bin, X., & Jie, C. (2015). Optimal design of controllers based on libraries and differential evolution. In *2015 34th Chinese control conference (CCC)* (pp. 5599–5604). Los Alamitos: IEEE.
103. Jiaoyang, Z., Bin, X., & Jie, C. (2017). Evolutionary design of controllers with optimized structure and its application in a Maglev ball control system. In *2017 36th Chinese control conference (CCC)* (pp. 2545–2550). Los Alamitos: IEEE.
104. Xin, B., Wang, Y., Xue, W., Cai, T., Fan, Z., Zhan, J., & Chen, J. (2021). Evolution of controllers under a generalized structure encoding/decoding scheme with application to magnetic levitation system. *IEEE Transactions on Industrial Electronics*, *69*(9), 9655–9666.
105. Zhang, S., Yang, P., Kong, L., Chen, W., Fu, Q., & Peng, K. (2019). Neural networks-based fault tolerant control of a robot via fast terminal sliding mode. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *51*(7), 4091–4101.
106. Grzeszczuk, R., & Terzopoulos, D. (1995). Automated learning of muscle-actuated locomotion through control abstraction. In S. G. Mair & R. Cook (Eds.), *Proceedings of the 22nd annual conference on computer graphics and interactive techniques* (pp. 63–70). New York: ACM.
107. Hornby, G. S., & Pollack, J. B. (2002). Creating high-level components with a generative representation for body-brain evolution. *Artificial Life*, *8*(3), 223–246.
108. Gallagher, J. C., Beer, R. D., Espenschied, K. S., & Quinn, R. D. (1996). Application of evolved locomotion controllers to a hexapod robot. *Robotics and Autonomous Systems*, *19*(1), 95–103.
109. Floreano, D., Husbands, P., & Nolfi, S. (2008). Evolutionary robotics. Technical report, Berlin: Springer.
110. Paul, C., & Bongard, J. C. (2001). The road less travelled: morphology in the optimization of biped robot locomotion. In *Proceedings 2001 IEEE/RSJ international conference on intelligent robots and systems. Expanding the societal role of robotics in the the next millennium (cat. no.01CH37180)* (Vol. 1, pp. 226–232). Los Alamitos: IEEE. https://doi.org/10.1109/IROS.2001.973363.
111. Rahmani, M., Ghanbari, A., & Ettefagh, M. M. (2018). A novel adaptive neural network integral sliding-mode control of a biped robot using bat algorithm. *Journal of Vibration and Control*, *24*(10), 2045–2060.
112. Dorigo, M., Birattari, M., & Brambilla, M. (2014). Swarm robotics. *Scholarpedia*, *9*(1), 1463.
113. Gao, G., Mei, Y., Xin, B., Jia, Y.-H., & Browne, W. N. (2022). Automated coordination strategy design using genetic programming for dynamic multipoint dynamic aggregation. *IEEE Transactions on Cybernetics*, *52*(12), 13521–13535. https://doi.org/10.1109/TCYB.2021.3080044.
114. Kazadi, S. (2009). Model independence in swarm robotics. *International Journal of Intelligent Computing and Cybernetics*, *2*(4), 672–694.
115. Berman, S., Kumar, V., & Nagpal, R. (2011). Design of control policies for spatially inhomogeneous robot swarms with application to commercial pollination. In *2011 IEEE international conference on robotics and automation* (pp. 378–385). Los Alamitos: IEEE.
116. Brambilla, M., Pinciroli, C., Birattari, M., & Dorigo, M. (2012). Property-driven design for swarm robotics. In V. Conitzer & M. Winikoff (Eds.), *Proceedings of the 11th international conference on autonomous agents and multiagent systems* (Vol. 1, pp. 139–146). IFAAMAS.
117. Francesca, G., Brambilla, M., Brutschy, A., Trianni, V., & Birattari, M. (2014). AutoMoDe: a novel approach to the automatic design of control software for robot swarms. *Swarm Intelligence*, *8*(2), 89–112.
118. Francesca, G., Brambilla, M., Brutschy, A., Garattoni, L., Miletitch, R., Podevijn, G., et al. (2015). AutoMoDe-Chocolate: automatic design of control software for robot swarms. *Swarm Intelligence*, *9*(2), 125–152.
119. Dorigo, M., Theraulaz, G., & Trianni, V. (2021). Swarm robotics: past, present, and future [point of view]. *Proceedings of the IEEE*, *109*(7), 1152–1165.
120. Nunes, E., Manner, M., Mitiche, H., & Gini, M. (2017). A taxonomy for task allocation problems with temporal and ordering constraints. *Robotics and Autonomous Systems*, *90*, 55–70.
121. Wu, M., Zhu, X., Ma, L., Wang, J., Bao, W., Li, W., & Fan, Z. (2022). Torch: strategy evolution in swarm robots using heterogeneous–homogeneous coevolution method. *Journal of Industrial Information Integration*, *25*, 100239.
122. Vásárhelyi, G., Virágh, C., Somorjai, G., Nepusz, T., Eiben, A. E., & Vicsek, T. (2018). Optimized flocking of autonomous drones in confined environments. *Science Robotics*, *3*(20), eaat3536.
123. Pfeifer, R., Lungarella, M., & Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science*, *318*(5853), 1088–1093.
124. Shah, D., Yang, B., Kriegman, S., Levin, M., Bongard, J., & Kramer-Bottiglio, R. (2021). Shape changing robots: bioinspiration, simulation, and physical realization. *Advanced Materials*, *33*(19), 2002882.

125. Miras, K., Ferrante, E., & Eiben, A. E. (2020). Environmental influences on evolvable robots. *PLoS ONE*, *15*(5), e0233848.
126. Lan, G., Jelisavcic, M., Roijers, D. M., Haasdijk, E., & Eiben, A. E. (2018). Directed locomotion for modular robots with evolvable morphologies. In *International conference on parallel problem solving from nature* (pp. 476–487). Berlin: Springer.
127. Eiben, A. E., Bredeche, N., Hoogendoorn, M., Stradner, J., Timmis, J., Tyrrell, A., & Winfield, A. (2013). The triangle of life: evolving robots in real-time and real-space. In *European conference on artificial life (ECAL-2013)* (pp. 1056–1063). Cambridge: The MIT Press.
128. Eiben, A. E., & Smith, J. (2015). From evolutionary computation to the evolution of things. *Nature*, *521*(7553), 476–482.
129. Eiben, A. E., Kernbach, S., & Haasdijk, E. (2012). Embodied artificial evolution. *Evolutionary Intelligence*, *5*(4), 261–272.
130. Bongard, J. C. (2013). Evolutionary robotics. *Communications of the ACM*, *56*(8), 74–83.
131. Marbach, D., & Ijspeert, A. J. (2004). Co-evolution of configuration and control for homogenous modular robots. In *Proceedings of the eighth conference on intelligent autonomous systems (IAS8)* (pp. 712–719). IOS Press.
132. Gupta, A., Savarese, S., Ganguli, S., & Fei-Fei, L. (2021). Embodied intelligence via learning and evolution. *Nature Communications*, *12*(1), 1–12.
133. Schaff, C. (2022). *Neural approaches to co-optimization in robotics*. arXiv preprint arXiv:2209.00579.
134. Meeden, L., & Kumar, D. (1998). Trends in evolutionary robotics. In L. C. Jain & T. Fukuda (Eds.), *Soft computing for intelligent robotic systems* (pp. 215–233). Berlin: Springer.
135. Zhao, A., Xu, J., Konaković-Luković, M., Hughes, J., Spielberg, A., Rus, D., & Matusik, W. (2020). Robogrammar: graph grammar for terrain-optimized robot design. *ACM Transactions on Graphics*, *39*(6), 1–16.
136. Xu, J., Spielberg, A., Zhao, A., Rus, D., & Matusik, W. (2021). Multi-objective graph heuristic search for terrestrial robot design. In *2021 IEEE international conference on robotics and automation (ICRA)* (pp. 9863–9869). Los Alamitos: IEEE.
137. Miriyev, A., & Kovač, M. (2020). Skills for physical artificial intelligence. *Nature Machine Intelligence*, *2*(11), 658–660.
138. Wang, J., Fan, Z., Terpenny, J. P., & Goodman, E. D. (2008). Cooperative body–brain coevolutionary synthesis of mechatronic systems. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *22*(3), 219–234.
139. Dupuis, J.-F., Fan, Z., & Goodman, E. (2015). Evolutionary design of discrete controllers for hybrid mechatronic systems. *International Journal of Systems Science*, *46*(2), 303–316.
140. Zhang, H., Liu, L. Z., Xie, H., Jiang, Y., Zhou, J., & Wang, Y. (2022). Deep learning-based robot vision: high-end tools for smart manufacturing. *IEEE Instrumentation & Measurement Magazine*, *25*(2), 27–35.
141. Zhang, X., Huang, Z., Wang, N., Xiang, S., & Pan, C. (2021). You only search once: single shot neural architecture search via direct sparse optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(9), 2891–2904.
142. Zheng, X., Ji, R., Chen, Y., Wang, Q., Zhang, B., Chen, J., Ye, Q., Huang, F., & Tian, Y. (2021). MIGO-NAS: towards fast and generalizable neural architecture search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(9), 2936–2952.
143. Xiong, Y., Liu, H., Gupta, S., Akin, B., Bender, G., Wang, Y., et al. (2021). Mobiledets: searching for object detection architectures for mobile accelerators. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3825–3834). Los Alamitos: IEEE.
144. Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., & Le, Q. V. (2019). MnasNet: platform-aware neural architecture search for mobile. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2820–2828). Los Alamitos: IEEE.
145. Zhou, X., Qin, A. K., Sun, Y., & Tan, K. C. (2021). A survey of advances in evolutionary neural architecture search. In *2021 IEEE congress on evolutionary computation (CEC)* (pp. 950–957). Los Alamitos: IEEE.
146. Baymurzina, D., Golikov, E., & Burtsev, M. (2022). A review of neural architecture search. *Neurocomputing*, *474*, 82–93.
147. Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural architecture search: a survey. *Journal of Machine Learning Research*, *20*(1), 1997–2017.
148. Baker, B., Gupta, O., Naik, N., & Raskar, R. (2016). *Designing neural network architectures using reinforcement learning*. arXiv preprint arXiv:1611.02167.
149. Zoph, B., & Le, Q. V. (2016). *Neural architecture search with reinforcement learning*. arXiv preprint arXiv:1611.01578.
150. Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8697–8710). Los Alamitos: IEEE.
151. Zhong, Z., Yan, J., Wu, W., Shao, J., & Liu, C.-L. (2018). Practical block-wise neural network architecture generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2423–2432). Los Alamitos: IEEE.
152. Liu, H., Simonyan, K., & Yang, Y. (2018). *Darts: differentiable architecture search*. arXiv preprint arXiv:1806.09055.
153. Chen, X., Xie, L., Wu, J., & Tian, Q. (2019). Progressive differentiable architecture search: bridging the depth gap between search and evaluation. In *Proceedings of the IEEE international conference on computer vision* (pp. 1294–1303). Los Alamitos: IEEE.
154. Stanley, K. O., Clune, J., Lehman, J., & Miikkulainen, R. (2019). Designing neural networks through neuroevolution. *Nature Machine Intelligence*, *1*(1), 24–35.
155. Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, *10*(2), 99–127.
156. Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., et al. (2019). Evolving deep neural networks. In R. Kozma, C. Alippi, Y. Choe, et al. (Eds.), *Artificial intelligence in the age of neural networks and brain computing* (pp. 293–312). Amsterdam: Elsevier.
157. Lu, Z., Whalen, I., Boddeti, V., Dhebar, Y., Deb, K., Goodman, E., & Banzhaf, W. (2019). NSGA-NET: neural architecture search using multi-objective genetic algorithm. In *Proceedings of the genetic and evolutionary computation conference* (pp. 419–427). New York: ACM.
158. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788). Los Alamitos: IEEE.
159. Meinhardt, T., Kirillov, A., Leal-Taixe, L., & Feichtenhofer, C. (2022). Trackformer: multi-object tracking with transformers. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8844–8854). Los Alamitos: IEEE.
160. Gilles, T., Sabatini, S., Tsishkou, D., Stanciulescu, B., & Moutarde, F. (2022). GOHOME: graph-oriented heatmap output for future motion estimation. In *2022 international conference on robotics and automation (ICRA)* (pp. 9107–9114). Los Alamitos: IEEE.
161. Ji, R., Li, K., Wang, Y., Sun, X., Guo, F., Guo, X., Wu, Y., Huang, F., & Luo, J. (2019). Semi-supervised adversarial monocular depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(10), 2410–2422.
162. Zhang, H., Liang, Z., Li, C., Zhong, H., Liu, L., Zhao, C., Wang, Y., & Wu, Q. J. (2021). A practical robotic grasping method by using 6-d pose estimation with protective correction. *IEEE Transactions on Industrial Electronics*, *69*(4), 3876–3886.
163. Gupta, A., Sheth, P., & Xie, P. (2022). Neural architecture search for pneumonia diagnosis from chest X-rays. *Scientific Reports*, *12*(1), Article No. 11309.
164. Oyelade, O. N., & Ezugwu, A. E. (2021). A bioinspired neural architecture search based convolutional neural network for breast cancer detection using histopathology images. *Scientific Reports*, *11*(1), Article No. 19940.
165. Chen, Y., Zhang, H., Wang, Y., Yang, Y., Zhou, X., & Wu, Q. M. J. (2021). MAMA Net: multi-scale attention memory autoencoder network for anomaly detection. *IEEE Transactions on Medical Imaging*, *40*(3), 1032–1041. https://doi.org/10.1109/TMI.2020.3045295.
166. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: common objects in context. In *European conference on computer vision* (pp. 740–755). Berlin: Springer.
167. Godard, C., Mac Aodha, O., Firman, M., & Brostow, G. J. (2019). Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE international conference on computer vision* (pp. 3828–3838). Los Alamitos: IEEE.
168. Wang, Y., Song, Y., Ma, C., & Zeng, B. (2020). Rethinking image deraining via rain streaks and vapors. In *European conference on computer vision* (pp. 367–382). Berlin: Springer.
169. Ren, W., Zhou, L., & Chen, J. (2022). Unsupervised single image dehazing with generative adversarial network. *Multimedia Systems*, 1–11.

170. Wan, J., Tang, S., Li, d., Imran, M., Zhang, C., Liu, C., & Pang, Z. (2018). Reconfigurable smart factory for drug packing in healthcare industry 4.0. *IEEE Transactions on Industrial Informatics*, *15*(1), 507–516.
171. Ma, Q., Li, H., & Chirikjian, G. S. (2016). New probabilistic approaches to the AX=XB hand-eye calibration without correspondence. In *2016 IEEE international conference on robotics and automation (ICRA)* (pp. 4365–4371). Los Alamitos: IEEE.
172. Niu, C., Zhu, Q., Wang, Y., Zhou, X., & Shen, W. (2021). Real time counting system of glass bottle based on multi objects tracking. In *2021 China automation congress (CAC)* (pp. 5402–5407). Los Alamitos: IEEE.
173. Yan, B., Peng, H., Wu, K., Wang, D., Fu, J., & Lu, H. (2021). LightTrack: finding lightweight neural networks for object tracking via one-shot architecture search. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 15180–15189). Los Alamitos: IEEE.
174. Viriyasaranon, T., & Choi, J.-H. (2022). Object detectors involving a NAS-gate convolutional module and capsule attention module. *Scientific Reports*, *12*(1), Article No. 3916.
175. Wang, N., Gao, Y., Chen, H., Wang, P., Tian, Z., Shen, C., & Zhang, Y. (2020). NAS-FCOS: fast neural architecture search for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 11943–11951). Los Alamitos: IEEE.
176. Chen, Y., Yang, T., Zhang, X., Meng, G., Xiao, X., & Sun, J. (2019). DetNAS: backbone search for object detection. In H. Wallach, H. Larochelle, A. Beygelzimer, et al. (Eds.), *Advances in neural information processing systems 32* (pp. 6642–6652). Red Hook: Curran Associates.
177. Yao, L., Xu, H., Zhang, W., Liang, X., & Li, Z. (2020). SM-NAS: structural-to-modular neural architecture search for object detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(7), 12661–12668. Menlo Park: AAAI Press.
178. Zhang, H., Wu, L., Chen, Y., Chen, R., Kong, S., Wang, Y., Hu, J., & Wu, J. (2022). Attention-guided multitask convolutional neural network for power line parts detection. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–13. https://doi.org/10.1109/TIM.2022.3162615.
179. Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., & Liang, J. (2018). Unet++: a nested u-net architecture for medical image segmentation. In D. Stoyanov, Z. Taylor, G. Carneiro, et al. (Eds.), *Deep learning in medical image analysis and multimodal learning for clinical decision support* (pp. 3–11). Berlin: Springer.
180. Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J., & Maier-Hein, K. H. (2021). NNU-NET: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, *18*(2), 203–211.
181. Liu, C., Chen, L.-C., Schroff, F., Adam, H., Hua, W., Yuille, A. L., & Fei-Fei, L. (2019). Auto-deeplab: hierarchical neural architecture search for semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 82–92). Los Alamitos: IEEE.
182. Nekrasov, V., Chen, H., Shen, C., & Reid, I. (2019). Fast neural architecture search of compact semantic segmentation models via auxiliary cells. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 9126–9135). Los Alamitos: IEEE.
183. Wei, J., Zhu, G., Fan, Z., Liu, J., Rong, Y., Mo, J., Li, W., & Chen, X. (2021). Genetic U-Net: automatically designed deep networks for retinal vessel segmentation using a genetic algorithm. *IEEE Transactions on Medical Imaging*, *41*(2), 292–307.
184. Zhang, F., Zhu, X., & Ye, M. (2019). Fast human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3517–3526). Los Alamitos: IEEE.
185. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., & Schindler, K. (2016). *MOT16: a benchmark for multi-object tracking*. arXiv preprint arXiv:1603.00831.
186. Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). *Object detection in 20 years: a survey*. arXiv preprint arXiv:1905.05055.
187. Godard, C., Mac Aodha, O., & Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 270–279). Los Alamitos: IEEE.
188. Nagarajan, V. R., & Singh, P. (2021). Obstacle detection and avoidance for mobile robots using monocular vision. In *2021 8th international conference on smart computing and communications (ICSCC)* (pp. 275–279). Los Alamitos: IEEE. https://doi.org/10.1109/ICSCC51209.2021.9528162.
189. Ohya, I., Kosaka, A., & Kak, A. (1998). Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing. *IEEE Transactions on Robotics and Automation*, *14*(6), 969–978.
190. Cao, T., Xiang, Z.-Y., & Liu, J.-L. (2015). Perception in disparity: an efficient navigation framework for autonomous vehicles with stereo cameras. *IEEE Transactions on Intelligent Transportation Systems*, *16*(5), 2935–2948. https://doi.org/10.1109/TITS.2015.2430896.
191. Song, S., Kim, D., & Choi, S. (2021). View path planning via online multiview stereo for 3-D modeling of large-scale structures. *IEEE Transactions on Robotics*, *38*(1), 372–390.
192. Huynh, L., Nguyen, P., Matas, J., Rahtu, E., & Heikkilä, J. (2022). Lightweight monocular depth with a novel neural architecture search method. In *Proceedings of the IEEE winter conference on applications of computer vision* (pp. 3643–3653). Los Alamitos: IEEE.
193. Saikia, T., Marrakchi, Y., Zela, A., Hutter, F., & Brox, T. (2019). Autodispnet: Improving disparity estimation with automl. In *Proceedings of the IEEE international conference on computer vision* (pp. 1812–1823). Los Alamitos: IEEE.
194. Zeng, K., Wang, Y., Mao, J., Liu, C., Peng, W., & Yang, Y. (2021). Deep stereo matching with hysteresis attention and supervised cost volume construction. *IEEE Transactions on Image Processing*, *31*, 812–822.
195. Cheng, X., Zhong, Y., Harandi, M., Dai, Y., Chang, X., Li, H., Drummond, T., & Ge, Z. (2020). Hierarchical neural architecture search for deep stereo matching. In H. Larochelle, M. Ranzato, R. Hadsell, et al. (Eds.), *Advances in Neural Information Processing Systems 33* (pp. 22158–22169). Red Hook: Curran Associates.
196. Zhang, C., Tian, K., Fan, B., Meng, G., Zhang, Z., & Pan, C. (2022). Continual stereo matching of continuous driving scenes with growing architecture. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 18901–18910). Los Alamitos: IEEE.
197. Wang, Q., Shi, S., Zhao, K., & Chu, X. (2022). *EASNet: searching elastic and accurate network architecture for stereo matching*. arXiv preprint arXiv:2207.09796.
198. Peng, W., Hong, X., & Zhao, G. (2019). Video action recognition via neural architecture searching. In *2019 IEEE international conference on image processing (ICIP)* (pp. 11–15). Los Alamitos: IEEE.
199. Piergiovanni, A. J., Angelova, A., & Ryoo, M. S. (2022). Tiny video networks. *Applied AI Letters*, *3*(1), e38.
200. Ryoo, M. S., Piergiovanni, A. J., Tan, M., & Angelova, A. (2019). *AssembleNET: searching for multi-stream neural connectivity in video architectures*. arXiv preprint arXiv:1905.13209.
201. Wang, X., Xiong, X., Neumann, M., Piergiovanni, A. J., Ryoo, M. S., Angelova, A., Kitani, K. M., & Hua, W. (2020). AttentionNAS: spatiotemporal attention cell search for video classification. In *European conference on computer vision* (pp. 449–465). Berlin: Springer.
202. Piergiovanni, A. J., Angelova, A., & Ryoo, M. (2020). Tiny video networks: architecture search for efficient video models. [Paper presentation]. In *ICML workshop on automated machine learning (AutoML)*. http://icml2020.automl.org.
203. Liu, S., Zheng, C., Lu, K., Gao, S., Wang, N., Wang, B., Zhang, D., Zhang, X., & Xu, T. (2021). Evsrnet: efficient video super-resolution with neural architecture search. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2480–2485). Los Alamitos: IEEE.
204. Xu, L., Guan, Y., Jin, S., Liu, W., Qian, C., Luo, P., Ouyang, W., & Wang, X. (2021). Vipnas: efficient video pose estimation via neural architecture search. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 16072–16081). Los Alamitos: IEEE.
205. Cai, H., Zhu, L., & Han, S. (2018). *ProxylessNAS: direct neural architecture search on target task and hardware*. arXiv preprint arXiv:1812.00332.
206. Wu, B., Dai, X., Zhang, P., Wang, Y., Sun, F., Wu, Y., Tian, Y., Vajda, P., Jia, Y., & Keutzer, K. (2019). Fbnet: hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 10734–10742). Los Alamitos: IEEE.
207. López, J. G., Agudo, A., & Moreno-Noguer, F. (2021). E-DNAS: differentiable neural architecture search for embedded systems. In *2020 25th international conference on pattern recognition (ICPR)* (pp. 4704–4711). Los Alamitos: IEEE.
208. Luo, X., Liu, d., Kong, H., Huai, S., Chen, H., & Liu, W. (2022). LightNAS: on lightweight and scalable neural architecture search for embedded platforms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1–14. https://doi.org/10.1109/TCAD.2022.3208187.
209. Cassimon, T., Vanneste, S., Bosmans, S., Mercelis, S., & Hellinckx, P. (2020). Designing resource-constrained neural networks using neural architecture search targeting embedded devices. *Internet of Things*, *12*, 100234.

210. Wan, A., Dai, X., Zhang, P., He, Z., Tian, Y., Xie, S., et al. (2020). Fbnetv2: differentiable neural architecture search for spatial and channel dimensions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 12965–12974). Los Alamitos: IEEE.
211. He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J., & Han, S. (2018). AMC: automl for model compression and acceleration on mobile devices. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 784–800). Berlin: Springer.
212. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). *Continuous control with deep reinforcement learning*. arXiv preprint arXiv:1509.02971.
213. Gupta, M., Aravindan, S., Kalisz, A., Chandrasekhar, V., & Jie, L. (2020). *Learning to prune deep neural networks via reinforcement learning*. arXiv preprint arXiv:2007.04756.
214. Yu, S., Mazaheri, A., & Jannesari, A. (2022). Topology-aware network pruning using multi-stage graph embedding and reinforcement learning. In *International conference on machine learning* (pp. 25656–25667). PMLR.
215. Wang, Z., & Li, C. (2022). Channel pruning via lookahead search guided reinforcement learning. In *Proceedings of the IEEE winter conference on applications of computer vision* (pp. 2029–2040). Los Alamitos: IEEE.
216. Yin, M., Sui, Y., Liao, S., & Yuan, B. (2021). Towards efficient tensor decomposition-based dnn model compression with optimization framework. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 10674–10683). Los Alamitos: IEEE.
217. Rokh, B., Azarpeyvand, A., & Khanteymoori, A. (2022). *A comprehensive survey on model quantization for deep neural networks*. arXiv preprint arXiv:2205.07877.
218. Chen, P., Liu, S., Zhao, H., & Jia, J. (2021). Distilling knowledge via knowledge review. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5008–5017). Los Alamitos: IEEE.
219. Cheng, J., Wang, P., Li, G., Hu, Q., & Lu, H. (2018). Recent advances in efficient computation of deep convolutional neural networks. *Frontiers of Information Technology & Electronic Engineering*, *19*(1), 64–77.
220. Bhalgaonkar, S. A., Munot, M. V., & Anuse, A. D. (2022). Pruning for compression of visual pattern recognition networks: a survey from deep neural networks perspective. In D. Gupta, R. S. Goswami, S. Banerjee, et al. (Eds.), *Pattern recognition and data analysis with applications* (pp. 675–687). Berlin: Springer.
221. Cheng, Y., Wang, D., Zhou, P., & Zhang, T. (2018). Model compression and acceleration for deep neural networks: the principles, progress, and challenges. *IEEE Signal Processing Magazine*, *35*(1), 126–136.
222. Wang, C.-H., Huang, K.-Y., Yao, Y., Chen, J.-C., Shuai, H.-H., & Cheng, W.-H. (2022). Lightweight deep learning: an overview. *IEEE Consumer Electronics Magazine*, 1–12. https://doi.org/10.1109/MCE.2022.3181759.
223. Sun, Y., Yen, G. G., & Yi, Z. (2018). IGD indicator-based evolutionary algorithm for many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, *23*(2), 173–187.
224. Darwish, A., Hassanien, A. E., & Das, S. (2020). A survey of swarm and evolutionary computing approaches for deep learning. *Artificial Intelligence Review*, *53*(3), 1767–1812.
225. Stamenkovic, A., Stapley, P. J., Robins, R., & Hollands, M. A. (2018). Do postural constraints affect eye, head, and arm coordination? *Journal of Neurophysiology*, *120*(4), 2066–2082.
226. Glaeser, G., & Paulus, H. F. (2015). *The evolution of the eye*. Springer.
227. Qiao, H., Ma, C., & Li, R. (2021). *The hand-eye-brain system of intelligent robot: from interdisciplinary perspective of information science and neuroscience*. Berlin: Springer.
228. Qiao, H., Chen, J., & Huang, X. (2022). A survey of brain-inspired intelligent robots: integration of vision, decision, motion control, and musculoskeletal systems. *IEEE Transactions on Cybernetics*, *52*(10), 11267–11280.
229. Huang, X., Wu, W., Qiao, H., & Ji, Y. (2018). Brain-inspired motion learning in recurrent neural network with emotion modulation. *IEEE Transactions on Cognitive and Developmental Systems*, *10*(4), 1153–1164.
230. Li, R., & Qiao, H. (2019). A survey of methods and strategies for high-precision robotic grasping and assembly tasks—some new trends. *IEEE/ASME Transactions on Mechatronics*, *24*(6), 2718–2732.
231. Chen, Z., & Qiao, H. (2020). Realizing compliant insertion task based on attractive-region-in-environment. In *2020 7th international conference on information science and control engineering (ICISCE)* (pp. 1063–1067). Los Alamitos: IEEE.
232. Qiao, H., Ma, C., & Li, R. (2022). The concept of "attractive region in environment (ARIE)" and its application in high-precision tasks with low-precision systems. In *The "hand-eye-brain" system of intelligent robot* (pp. 15–38). Berlin: Springer.
233. Qiao, H., Li, Y., Li, F., Xi, X., & Wu, W. (2015). Biologically inspired model for visual cognition achieving unsupervised episodic and semantic feature learning. *IEEE Transactions on Cybernetics*, *46*(10), 2335–2347.
234. Yin, P., Qiao, H., Wu, W., Qi, L., Li, Y., Zhong, S., & Zhang, B. (2017). A novel biologically inspired visual cognition model: automatic extraction of semantics, formation of integrated concepts, and reselection features for ambiguity. *IEEE Transactions on Cognitive and Developmental Systems*, *10*(2), 420–431.
235. Qiao, H., Ma, C., & Li, R. (2022). Biologically inspired visual model with preliminary cognition and active attention adjustment. In *The "hand-eye-brain" system of intelligent robot* (pp. 131–150). Berlin: Springer.
236. Huang, X., Wu, W., & Qiao, H. (2019). Connecting model-based and model-free control with emotion modulation in learning systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *51*(8), 4624–4638.
237. Huang, X., Wu, W., & Qiao, H. (2020). Computational modeling of emotion-motivated decisions for continuous control of mobile robots. *IEEE Transactions on Cognitive and Developmental Systems*, *13*(1), 31–44.
238. Yu, W., Hua, W., Qi, J., Zhang, H., Zhang, G., Xiao, H., Xu, S., & Ma, G. (2020). Coupled magnetic field-thermal network analysis of modular-spoke-type permanent-magnet machine for electric motorcycle. *IEEE Transactions on Energy Conversion*, *36*(1), 120–130.
239. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90.
240. Nahian, S. A., Truong, D. Q., Chowdhury, P., Das, D., & Ahn, K. K. (2016). Modeling and fault tolerant control of an electro-hydraulic actuator. *International Journal of Precision Engineering and Manufacturing*, *17*(10), 1285–1297.
241. Wei, H., Chen, Y., Tan, J., & Wang, T. (2010). Sambot: a self-assembly modular robot system. *IEEE/ASME Transactions on Mechatronics*, *16*(4), 745–757.
242. Gilpin, K., & Rus, D. (2010). Modular robot systems. *IEEE Robotics & Automation Magazine*, *17*(3), 38–55.
243. Fukuda, T., & Kubota, N. (2003). Computational intelligence for robotic systems. In *12th IEEE international conference on fuzzy systems* (pp. 1495–1508). Los Alamitos: IEEE.
244. Gallala, A., Kumar, A. A., Hichri, B., & Plapper, P. (2022). Digital twin for human–robot interactions by means of industry 4.0 enabling technologies. *Sensors*, *22*(13), 4950.
245. Sutton, S. G., Arnold, V., & Holt, M. (2018). How much automation is too much? Keeping the human relevant in knowledge work. *Journal of Emerging Technologies in Accounting*, *15*(2), 15–25.
246. Dorigo, M., Theraulaz, G., & Trianni, V. (2020). Reflections on the future of swarm robotics. *Science Robotics*, *5*(49), eabe4385.
247. Rodríguez-Molina, A., Mezura-Montes, E., Villarreal-Cervantes, M. G., & Aldape-Pérez, M. (2020). Multi-objective meta-heuristic optimization in intelligent control: a survey on the controller tuning problem. *Applied Soft Computing*, *93*, 106342.
248. Wang, M., & Deng, W. (2018). Deep visual domain adaptation: a survey. *Neurocomputing*, *312*, 135–153.
249. Wang, J., Lan, C., Liu, C., Ouyang, Y., Qin, T., Lu, W., Chen, Y., Zeng, W., & Yu, P. (2022). Generalizing to unseen domains: a survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 1–14. https://doi.org/10.1109/TKDE.2022.3178128.
250. Bai, H., Zhou, F., Hong, L., Ye, N., Chan, S. G., & Li, Z. (2021). NAS-OOD: neural architecture search for out-of-distribution generalization. In *Proceedings of the IEEE international conference on computer vision* (pp. 8320–8329). Los Alamitos: IEEE.
251. Wen, Y.-W., Peng, S.-H., & Ting, C.-K. (2021). Two-stage evolutionary neural architecture search for transfer learning. *IEEE Transactions on Evolutionary Computation*, *25*(5), 928–940.
252. Li, Y., Yang, Z., Wang, Y., & Xu, C. (2020). Adapting neural architectures between domains. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems 33* (pp. 789–798). Red Hook: Curran Associates.
253. Guo, N., Gu, K., Qiao, J., & Liu, H. (2022). Active vision for deep visual learning: a unified pooling framework. *IEEE Transactions on Industrial Informatics*, *18*(10), 6610–6618. https://doi.org/10.1109/TII.2021.3129813.

254. Ito, J., Joana, C., Yamane, Y., Fujita, I., Tamura, H., Maldonado, P. E., & Grün, S. (2022). Latency shortening with enhanced sparseness and responsiveness in V1 during active visual sensing. *Scientific Reports*, *12*(1), 1–17.
255. Xu, Y., Xie, L., Dai, W., Zhang, X., Chen, X., Qi, G.-J., Xiong, H., & Tian, Q. (2021). Partially-connected neural architecture search for reduced computational redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(9), 2953–2970.
256. Lan, X., & Schwager, m. (2016). Rapidly exploring random cycles: persistent estimation of spatiotemporal fields with multiple sensing robots. *IEEE Transactions on Robotics*, *32*(5), 1230–1244.
257. Carrillo, H., Dames, P., Kumar, V., & Castellanos, J. A. (2015). Autonomous robotic exploration using occupancy grid maps and graph slam based on Shannon and Rényi entropy. In *2015 IEEE international conference on robotics and automation (ICRA)* (pp. 487–494). Los Alamitos: IEEE.
258. Meng, Y., Wang, W., Han, H., & Ban, J. (2019). A visual/inertial integrated landing guidance method for UAV landing on the ship. *Aerospace Science and Technology*, *85*, 474–480.
259. Zheng, J., Yang, T., Liu, H., Su, T., & Wan, L. (2020). Accurate detection and localization of unmanned aerial vehicle swarms-enabled mobile edge computing system. *IEEE Transactions on Industrial Informatics*, *17*(7), 5059–5067.
260. Zheng, J., Chen, R., Yang, T., Liu, X., Liu, H., Su, T., & Wan, L. (2021). An efficient strategy for accurate detection and localization of UAV swarms. *IEEE Internet of Things Journal*, *8*(20), 15372–15381.
261. Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: a review. *Neural Networks*, *113*, 54–71.
262. Du, X., Li, Z., Sun, J., Liu, F., & Cao, Y. (2021). Evolutionary NAS in light of model stability for accurate continual learning. In *2021 international joint conference on neural networks (IJCNN)* (pp. 1–8). Los Alamitos: IEEE.
263. Gao, Q., Luo, Z., Klabjan, D., & Zhang, F. (2022). Efficient architecture search for continual learning. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(2), 690–702.
264. Mundt, M., Pliushch, I., & Ramesh, V. (2021). Neural architecture search of deep priors: towards continual learning without catastrophic interference. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3523–3532). Los Alamitos: IEEE.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.