Main Strategies for Autonomous Robotic Controller Design

I. Paterega

Abstract— **This review gives an overall introduction to the artificial evolution mechanism. It presents the main strategies for robotic controller design. It gives a review of the pertinent literature, focusing on approaches that use neural networks, evolutionary computing, and fuzzy logic. Various applications of artificial evolution in robotics are surveyed and classified.**

***Index Terms*— evolutionary algorithms, fuzzy logic, neural networks, robot navigation.**

I. INTRODUCTION

Early robots were nothing more than clever mechanical devices that performed simple pick-and-place operations. Nowadays robots are becoming more and more sophisticated and diversified so as to meet the ever-changing user requirements. The robots are developed to perform more precise industrial operations, such as welding, spray painting, and simple parts assembly.

However, such operations do not really require the robot to have intelligence and behave like human beings since the robots are simply programmed to perform a series of repetitive tasks. If anything interferes with the prespecified task, the robot cannot work properly anymore, since it is not capable of sensing its external environment and figuring out what to do independently.

Modern robots are required to carry out work in unstructured dynamic human environments. In the recent decades, the application of artificial evolution to autonomous mobile robots to enable them to adapt their behaviors to changes of the environments has attracted much attention. As a result, an infant research field called evolutionary robotics has been rapidly developed that is primarily concerned with the use of artificial evolution techniques for the automatic design of adaptive robots. As an innovative and effective solution to autonomous robot controller design, it can derive adaptive robotic controllers capable of elegantly dealing with continuous changes in unstructured environments in real time [1]. It has been shown in [2] that the robot behaviors could be achieved more effectively by using simpler and more robust evolutionary approaches than the traditional decomposition/integration approach.

Iurii I. Paterega is with CAD/CAM Department, Lviv Polytechnic National University, 12, S. Bandera Str., Lviv, 79013, UKRAINE. E-mail: paterega@mail.lviv.ua

Evolutionary robotics aims to develop a suitable control system of the robot through artificial evolution. Evolution and learning are two forms of biological adaptation that operate on different time scales. Evolution is capable of capturing slow environmental changes that might occur through several generations, whereas learning may produce adaptive changes in an individual during its lifetime. Recently, researchers have started using artificial evolution techniques, such as genetic algorithm (GA), fuzzy logic (FA) and learning technique, namely neural network (NN), to study the interaction between evolution and learning [3].

Evolutionary robotics deals with this interaction. In behavior-based robotics, a task is divided into a number of basic behaviors by the designer and each basic behavior is implemented in a separate layer of the robot control system. The control system is built up incrementally layer by layer and each layer is responsible for a single basic behavior. The coordination mechanism of basic behaviors is usually designed through a trial and error process and the behaviors are coordinated by a central mechanism. It is important to note that the number of layers increases with the complexity of the problem and for a very complex task, it may go beyond the capability of the designer to define all the layers, their interrelationships and dependencies. Hence, there is a need for a technique by which the robot is able to acquire new behaviors automatically depending on the situations of changing environment. Evolutionary robotics may provide a feasible solution to the abovementioned problem. In evolutionary robotics, the designer plays a passive role and the basic behaviors emerge automatically through evolution due to the interactions between the robot and its environment.

This review gives an overall introduction of the artificial evolution mechanism. It presents the main strategies for robotic controller design. Various applications of artificial evolution in robotics are surveyed and classified. Furthermore, in this review their specific merits and drawbacks in robotic controller design are discussed, as at present, there is little consensus among researchers as to the most appropriate artificial evolution approach for heterogeneous evolutionary systems.

II. Evolution Mechanisms

A robot is required to have intelligence and autonomous abilities when it works far from an operator and these are a large time delay or working in a world containing uncertainty. The robot collects or receives the necessary information concerning its external environment, and takes action in the environment. Both processes are usually designed by human operators, but ideally, the robot should perform the given task automatically without human assistance. Computational intelligence methods, including neural networks (NNs), fuzzy logic (FLs), evolutionary algorithms (EAs), reinforcement learning, expert systems and others, have been applied to realize intelligence in robotic systems.

To realize an advanced intelligent system, a synthesized algorithm of various techniques such as NN, FL, and EC is required. Each technique plays a specific role in intelligence features. There are no complete techniques for realizing all features of intelligence. Therefore, it is necessary to integrate and combine several techniques to compensate for the disadvantages of each technique. The main characteristics of NN are to classify or recognize patterns, and to adapt itself to dynamic environments by learning, but the mapping structure of NN is a black box and is incomprehensible. On the other hand, FL has been applied to represent human linguistic rules and classify numerical information into symbolic classes. It also has a reasonable structure for inference, which is composed of if-then rules as in human knowledge [4].

However, FL does not fundamentally have a learning mechanism. Fuzzy-neural networks have been developed to overcome these disadvantages. In general, the neural network part is used for learning, while the fuzzy logic part is used for representing knowledge. Learning capabilities such as incremental learning, the back-propagation method, and the delta rule based on error functions are used in essential changes. EC can also tune NN and FL. However, evolution can be defined as a resultant or accidental change, not a necessary change, since EC cannot predict or estimate the effect of the change. To summarize, an intelligent system can quickly adapt to a dynamic environment via NN and FL using the back-propagation method or the delta rule, and furthermore, the structure of an intelligent system can evolve globally via EC according to its objectives.

III. Neural Networks

Many evolutionary approaches have been applied to the field of evolvable robotic controller design in the recent decades [5]-[7]. Some researchers used artificial Neural Networks (NN) as the basic building blocks for the control system due to their smooth search space. NNs can be envisaged as simple nodes connected together by directional interconnects along which signals flow. The nodes perform an input-output mapping that is usually some sort of sigmoid function.

An artificial NN is a collection of neurons connected by weighted links used to transmit signals. Input and output neurons exchange information with the external environment by receiving and broadcasting signals. In essence, a neural network can be regarded as a parallel computational control system since signals in it travel independently on weighted channels and neuron states can be updated in parallel. NN advantages include its learning and adaptation through efficient knowledge acquisition, domain free modeling, robustness to noise, and fault tolerance, etc. [8]. Also neural networks can easily exploit various forms of learning during life-time and this learning process may help and speed up the evolutionary process [9], [10]. Neural networks are resistant to noise that is massively present in robot/environment interactions. This fact also implies that the fitness landscape of neural networks is not very rugged because sharp changes of the network parameters do not normally imply big changes in the fitness level. On the contrary it has been shown that introducing noise in neural networks can have a beneficial effect on the course of the evolutionary process [11]. The primitive’s components manipulated by the evolutionary process should be at the lowest level possible in order to avoid undesirable choices made by the human designer [12]. Synaptic weights and nodes are low level primitive components.

The behaviors that evolutionary robotics is concerned with at present are low-level behaviors, tightly coupled with the environment through simple, precise feedback loops. Neural networks are suitable for this kind of applications so that the predominant class of systems for generating adaptive behaviors adopts neural networks [13]. The same encoding schemes can be used independently of the specific autonomous robot navigation system since different types of functions can be achieved with the same type of network structure by varying the properties and parameters of simple processing used. Other adaptive processes such as supervised and unsupervised learning can also be incorporated into NN to speed up the evolution process.

NNs have been widely used in the evolutionary robotics due to the aforementioned merits. For instance, locomotion-control module based on recurrent neural networks has been studied by Beer and Gallagher [14] for an insect-like agent. Parisi, Nolfi, and Cecconi [15] developed back propagation neural networks for agents collecting food in a simple cellular world. Cliff, Harvey, and Husbands [12] have integrated the incremental evolution into arbitrary recurrent neural networks for robotic controller design. Floreano and Mondada [16] presented an evolution system of a discrete-time recurrent neural network to create an emergent homing behavior.

NN has also been used for Intelligent Autonomous Vehicles (IAV) design. The primary goal of IAV is related to the theory and applications of robotic systems capable of some degree of self-sufficiency. The focus is on the ability to move and be self-sufficient in partially structured environments. IAV have many applications in a large variety of domains, from spatial exploration to handling material, and from military tasks to the handicapped help. The recent developments in autonomy requirements, intelligent components, multi-robot systems, and massively parallel computers have made the IAV very used in particular in planetary explorations, mine industry, and highways [17].

To reach their targets without collisions with possibly encountered obstacles, IAV must have the capability to achieve target localization and obstacle avoidance behaviors. More, current IAV requirements with regard to these behaviors are real-time, autonomy and intelligence. Thus, to acquire these behaviors while answering IAV requirements, IAV must be endowed with recognition, learning, decision-making, and action capabilities.

To achieve this goal, classical approaches rapidly have been replaced by current approaches in particular the Neural Networks (NN) based approaches. Indeed, the aim of NN is to bring the machine behavior near the human one in recognition, learning, decision-making, and action. In [17], a first current NN based navigation approaches in IAV, autonomy, and intelligence have been discussed.

However, neural networks also have certain drawbacks. For instance, a NN cannot explain its results explicitly and its training is usually time-consuming. Furthermore, the learning algorithm may not be able to guarantee the convergence to an optimal solution [8].

IV. Evolutionary Algorithms

There are currently several flavors of evolutionary algorithms (EAs). Genetic Algorithms (GAs) [18] is the most commonly used one where genotypes typically are strings of binary. Genetic Programming (GP) [19] is an offshoot of GAs, where genotypes are normally computer programs. Other flavors such as Evolution Strategies (ES) are also used in evolutionary robotics (ER). Many concerns are shared among these approaches.

As a commonly used EA, GA has also been used in [10], [19] for generating robotic behaviors. Thompson [20] adopts the conventional GA as the training tool to derive the robot controllers in the hardware level. The encouraging experimental results justify the effectiveness of GA as a robust search algorithm even in hardware evolution.

Most applications nowadays use the orthodox GA, however, Species Adaptation GAs (SAGA) suggested by [21], [22] would be more suitable for certain robot evolution applications such as evolvable hardware based robotic evolutions. In SAGA, different structures are encoded with genotypes of different lengths, which offer a search space of open-ended dimensionality. Cyclic Genetic Algorithm (CGA) has also been introduced in [23] to evolve robotic controllers for cyclic behaviors. Also distributed genetic algorithms have been introduced into the evolutionary robotics field recently. For instance, in the spatially distributed GA, for each iteration a robot is randomly selected from a population distributed across a square grid. The robot is bred with one of its fittest neighbors and their offspring replaces one of the least fit neighbors such that the selection pressure keeps successful genes in the population. The distributed GA is usually robust and efficient in evolving capable robots. GA exhibits its advantages in deriving robust robotic behavior in conditions where large numbers of constraints and/or huge amounts of training data are required [24]. Furthermore, GA can be applied to a variety of research communities due to its gene representation. However, GA is computationally expensive [24]. Though GA is now widely used in the ER field, a variety of issues are still open in the GA-based ER.

For instance, the fitness function design is an important issue in GA-based evolution schemes [25]. The fitness function should present measurement of its ability to perform under all of the operating conditions. In fact, all these objectives can be fulfilled by setting an appropriate fitness function so as to derive the desired robotic performance exhibited during autonomous navigation. Therefore, the fitness function design needs to be investigated more carefully to make the robot evolve in a more effective way. Several experiments have also been performed where the robotic controllers were evolved through Genetic Programming (GP) [19], [26].

V. Fuzzy Logic

Fuzzy logic provides a flexible means to model the nonlinear relationship between input information and control output [27]. It incorporates heuristic control knowledge in the form of if-then rules, and is a convenient alternative when the system to be controlled cannot be precisely modeled [28], [29]. They have also shown a good degree of robustness in face of large variability and uncertainty in the parameters.

These characteristics make fuzzy control particularly suited to the needs of autonomous robot navigation [30]. Fuzzy logic has remarkable features that are particularly attractive to the hard problems posed by autonomous robot navigation. It allows us to model uncertainty and imprecision, to build robust controllers based on the heuristic and qualitative models, and to combine symbolic reasoning and numeric computation. Thus, fuzzy logic is an effective tool to represent real world environments. In evolutionary robotics, fuzzy logic has been used to design sensor interpretation systems since it is good at describing uncertain and imprecise information.

All the specific methods have their own strengths and drawbacks. Actually they are deeply interconnected and in many applications some of them have been combined together to derive the desired robotic controller in the most effective and efficient manner. For instance, Fuzzygenetic system [31] is a typical evolution mechanism in evolving adaptive robot controller. Arsene and Zalzala [32] controlled the autonomous robots by using fuzzy logic controllers tuned by GA. Pratihar, Deb, and Ghosh [33] used fuzzy-GA to find obstacle-free paths for a mobile robot. Driscoll and Peters II [34] implemented a robotic evolution platform supporting both GA and NN. Xiao, et al. [35] designed autonomous robotic controller using DNA coded GA for fuzzy logic optimization.

Fuzzy control has shown to be a very useful tool in the field of autonomous mobile robotics, characterized by a high uncertainty in the knowledge about the environment where a robot evolves. The design of a fuzzy controller is generally made using expert knowledge about the task to be controlled. Expert knowledge is applied in order to decide the number of linguistic labels for each variable, to tune the membership functions, to select the most adequate linguistic values for the consequents, and to define the rules in the fuzzy knowledge base. This process is tedious and highly time-consuming [36].

For this reason, automated learning techniques, such as evolutionary algorithms, have been employed for helping in some, or in all, of the tasks involved in the design process. In some of the approaches evolutionary algorithms are used just for tuning the membership functions. In others, the complete rule base is learned, starting from a hand designed data base (number and definition of the linguistic values and universe of discourse of the variables). But only in a few of them both the data base and the rule base are learned.

Mucientes, Moreno, Bugarıın and Barro describe the learning of a fuzzy controller for the wall-following behavior in a mobile robot [36]. The learning methodology is characterized by three main points. First, learning has no restrictions neither in the number of membership functions, nor in their values. In the second place, the training set is composed of a set of examples uniformly distributed along the universe of discourse of the variables.

Fuzzy logic techniques are commonly used for navigation of different types of robot vehicles [38]. The popularity of fuzzy logic is based on the fact that it can cope with the uncertainty of the sensors and the environment really well. By using it, the robotic vehicles are able to move in known or unknown environments, using control laws that derive from a fuzzy rule base. This base is consisted from a set of predefined IF– THEN rules, which remains constant during the operation of the robot. These rules along with the membership functions of the fuzzy variables are usually designed ad hoc by human experts [37].

Several researchers have used fuzzy logic for the navigation of mobile robots. In [39], a layer goal oriented motion planning strategy using fuzzy logic controllers has been offered, which uses sub-goals in order to move in a specific target point. Another approach is presented in [40], where the authors offer a control system consisting of fuzzy behaviors for the control of an indoor mobile robot. All the behaviors are implemented as Mamdani fuzzy controllers, except for one which is implemented as adaptive neuro-fuzzy. In [41] a combined approach of fuzzy and electrostatic potential fields is presented that assures navigation and obstacle avoidance. The main drawback of these approaches is that the design of the fuzzy controllers relies mainly on the experience of the designer. In order to overcome this problem several researchers have suggested tuning the fuzzy logic controller based on learning methods [42] and evolutionary algorithms [43–48], in an attempt to improve the performance and the behavior of the control procedure.

In [43], a fuzzy logic controller for a Khepera robot in a simulated environment evolved using a genetic algorithm, and the behaviors of the evolved controller were analyzed with a state transition diagram. The robot produces emergent behaviors by the interaction of fuzzy rules that came out from the evolution process. In [44], the authors suggested a three step evolution process to self-organize a fuzzy logic controller. The procedure initially tunes the output term set and rule base, then the input membership functions, and in the third phase it tunes the output membership functions. Hargas et al. in [45], suggested a fuzzy-genetic technique for the on-line learning and adaptation of an intelligent robotic vehicle. In [46] the authors present a methodology for tuning the knowledge base of the fuzzy logic controller based on a compact scheme for the genetic representation of the fuzzy rule base.

In [47] the authors present a scheme for the evolution of the rule base of a fuzzy logic controller. The evolution takes place in simulated robots and the evolved controllers are tested on a Khepera mobile robot. Nanayakkara et al. in [48], present an evolutionary learning methodology using a multi objective fitness function that incorporates several linguistic features. The methodology is compared to the results derived from a conventional evolutionary algorithm. An attempt to formulate a way of picking the suitable function for a task was made by Nolfi and Floreano in [49]. They suggested the concept of “fitness space”, which provides a framework for the description and development of fitness functions for autonomous systems.

An important issue not addressed in the literature, is related to the selection of the fitness function parameters used in the evolution process of fuzzy logic controllers. The majority of the fitness functions used for controllers evolution are empirically selected and (most of times) task specified. This results to controllers which heavily depend on fitness function selection.

The experience in the design of the nonlinear position control confirmed the remarkable potential of fuzzy logic in the development of effective decision laws capable of overcoming the inherent limitations of model-based control strategies [50].

Lacevic and Velagic [50] focused on the design of the fuzzy logic-based position control of the mobile robot that both meets a good position tracking requirements and has practically achievable control efforts.

With our previously designed CLF based controller a good tracking performance has been obtained. However, its significant shortcoming is unsatisfactory velocity/torque command values, particularly at the beginning of tracking. Control parameters of the CLF-based controller and the membership functions of the fuzzy position controller are evolved by the genetic algorithms. The advantage of the offered fuzzy controller lies in the fact that the velocity commands (and consequently, the torque commands) cannot exceed certain limits. Consequently, this controller radically decreased the control velocities without major impact on the tracking performance. Finally, from the obtained simulation results, it can be concluded that the proposed fuzzy design achieves the desired results. The future work will investigate the stability analysis of the system when the proposed fuzzy logic-based position controller is used.

VI. Other Methods

Apart for the above commonly used methodologies, several other evolutionary approaches have also been tested in the ER field in recent years. For example, classifier systems have been used as an evolution mechanism to shape the robotic controllers [51], [52]. Grefenstette and Schultz used the SAMUEL classifier system to evolve anti-collision navigation [53], [54]. Katagami and Yamada [55] suggested a learning method based on interactive classifier system for mobile robots which acquires autonomous behaviors from the interaction experiences with a human. Gruau and Quatramaran [56] developed robotic controllers for walking in the OCT-1 robot using cellular encoding. In the work of Berlanga et al. [57], the ES has been adopted to learn high-performance reactive behavior for navigation and collisions avoidance. Embodied evolution has been offered as a methodology for the automatic design of robotic controllers [58], which avoids the pitfalls of the simulate-and-transfer method. Most of the aforementioned ER approaches are essentially software based.

Nowadays, hardware-based robotic controllers using artificial evolution as training tools are also being used. The development of evolvable hardware (EHW) has attracted much attention from the ER domain, which is a new set of integrated circuits able to reconfigure their architectures using artificial evolution techniques unlimited times. Higuchi, Iba, and Manderick [59] used off-line model-free and on-line model-based methods to derive robot controllers on the logic programmable device. Attempting to exploit the intrinsic properties of the hardware, Thompson [20] used a Dynamic State Machine (DSM) to control a Khepera robot to avoid obstacles in a simple environment.

Tan, Wang, Lee and Vadakkepat in [60] discusses the application of evolvable hardware in evolutionary robotics, which is a new set of integrated circuits capable of reconfiguring its architecture using artificial evolution techniques. Hardware evolution dispenses with conventional hardware designs in solving complex problems in a variety of application areas, ranging from pattern recognition to autonomous robotics.

VII. Conclusion

Free-navigating mobile robotic systems can be used to perform service tasks for a variety of applications such as transport, surveillance, firefighting, etc. For such robotic application systems, it is crucial to derive simple robotic behaviors that guarantee robust operation despite of the limited knowledge prior to system execution, e.g., designing anti-collision behavior that is effective in the presence of unknown obstacle shapes. In recent years, autonomous mobile service robots have been introduced into various non-industrial application domains including entertainment, security, surveillance, and healthcare. They can carry out cumbersome work due to their high availability, fast task execution, and cost-effectiveness.

An autonomous mobile robot is essentially a computational system that acquires and analyzes sensory data or exterior stimuli and executes behaviors that may affect the external environment. It decides independently how to associate sensory data with its behaviors to achieve certain objectives.

Such an autonomous system is able to handle uncertain problems as well as dynamically changing situations. Evolutionary robotics appears to be an effective approach to realizing this purpose. In this paper some applications of evolutionary approach in autonomous robotics are considered. A general survey is reported regarding the effectiveness of a variety of artificial evolution based strategies in robotics. Some questions need to be answered if evolutionary robotics is to progress beyond the proof-of-concept stage. Furthermore, future prospects including combination of learning and evolution, inherent fault tolerance, hardware evolution, on-line evolution, and ubiquitous and collective robots are suggested.

References

[1] L. Wang, K. Chen Tan, Chee Meng Chew, “Evolutionary robotics: from algorithms to implementations,” World Scientific Series in Robotics and Intelligent Systems., vol. 28, 2006.

[2] S. Nolfi, “Adaptation as a more powerful tool than decomposition and integration: experimental evidences from evolutionary robotics,” Proceedings of the IEEE World Congress on Computational Intelligence, 1998, pp. 141 -146.

[3] D. Pratihar, “Evolutionary robotics – A review,” Sadhana, vol. 28, Springer India, in co-publication with Indian Academy of Sciences, 2003, pp. 999–1009.

[4] T. Fukuda, Yasuhisa Hasegawa, “Evolutionary computing in robotics,” Artificial Life and Robotics, vol. 6, Springer Japan, 2002, pp. 1-2.

[5] J.-A. Meyer, “Evolutionary approaches to neural control in mobile robots,” Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 1998, pp. 2418 -2423.

[6] O. Chocron, P. Bidaud, “Evolving walking robots for global task based design,” Proceedings of the International Congress on Evolutionary Computation, 1999, pp. 405-412.

[7] J. B. Pollack, H. Lipson, S. Ficci, et al., “Evolutionary techniques in physical robotics, Evolvable Systems: From Biology to Hardware,” Lecture Notes in Computer Science 1801 (Proc. of ICES2000), Springer- Verlag., 2000, pp. 175-186.

[8] S. H. Huang, “Artificial neural networks and its anufacturing application: Part I,” on-line, 2002.

[9] D. H. Ackley, M. L. Littman, “Interactions between learning and evolution. In Artificial Life II,” edited by C. G. Langton, J. D. Farmer, S. Rasmussen, C. E. Taylor. Addison-Wesley. Reading, Mass., 1991.

[10] D. Parisi, S. Nolfi, “How learning can influence evolution within a non-Lamarckian framework. In Plastic Individuals in Evolving Populations,” edited by R. K. Belew, M. Mitchell. SFI Series, Addison-Wesley, in press.

[11] O. Miglino, R. Pedone, D. Parisi, “A noise Gene for Econets,” In Proceedings of Genetic Algorithms and Neural Networks, edited by M. Dorigo, Reading, Mass.: Addison Wesley, 1993.

[12] D. T. Cliff, I. Harvey, P. Husbands, “Explorations in Evolutionary Robotics,” Adaptive Behavior, vol. 2, 1993, pp. 73-110.

[13] N. Jakobi, “Running across the reality gap: Octopod locomotion evolved in a minimal simulation,” Proceedings of the First European

Workshop on Evolutionary Robotics 98 (EvoRobot98), France, 1998, pp. 39-58.
[14] R. D. Beer, J. C. Gallagher, "Evolving dynamic neural networks for adaptive behavior," Adaptive Behavior, vol. 1, 1992, pp. 91-122.
[15] N. J. D. Parisi, S. Nolfi, F. Cecconi, "Leaning, behavior and evolution," Proceedings of the First European Conference on Artificial Life, Cambridge, MA: MIT Press/Bradford Books, pp. 1992, 207- 216.
[16] D. Floreano, F. Mondada, "Evolution of homing navigation in a real mobile robot," IEEE Transactions on Systems, Man and Cybernetics – Part B, vol. 26 (3), 1996.
[17] A. Chohra, A. Farah and C. Benmehrez, "Neural Navigation Approach for Intelligent Autonomous Vehicles (IAV) in Partially Structured Environments," Applied Intelligence, vol. 8, Springer Netherlands, 1998, pp. 219-233.
[18] J. H. Holland, "Adaptation In Natural And Artificial Systems," Ann Arbor: The University of Michigan Press. 1975.
[19] J. R. Koza, "Genetic Programming II," The MIT Press, Cambridge, Mass., USA, 1994.
[20] A. Thompson, "Evolving electronic robot controllers that exploit hardware resources," Proc. of the 3rd European Conf. on Artificial Life (ECAL95), Springer-Verlag, 1995, pp. 640-656.
[21] I. Harvey, "Species Adaptation Genetic Algorithms: A basis for a continuing SAGA," Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life, F. J. Varela and P. Bourgine (eds.), MIT Press/Bradford Books, Cambridge, MA, 1992, pp. 346-354.
[22] I. Harvey, "Artificial evolution: a continuing SAGA," T. Gomi, (ed.): ER2001, LNCS 2217, 2001, pp. 94-109.
[23] G. B. Parker, "The co-evolution of model parameters and control programs in evolutionary robotics," Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, 1999, pp. 162 -167.
[24] J. F. Walker, J. H. Oliver, "A survey of artificial life and evolutionary robotics," 1997.
[25] T. E. Revello, R. McCartney, "A cost term in an evolutionary robotics fitness function," Proceedings of the 2000 Congress on Evolutionary Computation, 2000, pp. 125 -132.
[26] P. Dittrich, A. Burgel, W. Banzhaf, "Learning to move a robot with random morphology," Proceedings of the First European Workshop on Evolutionary Robotics (EvoRobot 98), France, 1998, pp. 165-178.
[27] F. Hoffman, T. J. Koo, O. Shakernia, "Evolutionary design of a helicopter autopilot," Proceedings of the 3rd On-line World Conference on Soft Computing, Cranfield, UK., 1998.
[28] P. N. Paraskevopoulos, "Digital Control Systems," Prentice Hall., 1996.
[29] D. Driankov, A. (Eds). Saffiotti, "Fuzzy Logic Techniques For Autonomous Vehicle Navigation," Springer-Verlag., 2001.
[30] A. SafBotti, "The uses of fuzzy logic in autonomous robot navigation," Soft Computing, Springer-Verlag, 1997, pp. 180-197.
[31] H. Hagras, V. Callaghan, M. Colley, "Outdoor mobile robot learning and adaptation," IEEE Robotics and Automation Magazine, 2001, pp. 53-69.
[32] C. T. C. Arsene, A. M. S. Zalzala, "Control of autonomous robots using fuzzy logic controllers tuned by genetic algorithms," Proceedings of the International Congress on Evolutionary Computation 1999, 1999, pp. 428-435.
[33] D. K. Pratihar, K. Deb, A. Ghosh, "Fuzzy-genetic algorithms and mobile robot navigation among static obstacles," Proceedings of the International Congress on Evolutionary Computation 1999, 1999, pp. 327-334, 1999.
[34] J. A. Driscoll, R. A. Peters II, "A development environment for evolutionary robotics," Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 2000, pp. 3841-3845.
[35] P. Xiao, V. Prahlad, T. H. Lee, X. Liu, "Mobile robot obstacle avoidance: DNA coded GA for FLC optimization," Proc. of 2002 FIRA Robot World Congress, Korea, 2002, pp. 553 – 558.
[36] M. Mucientes, D.L. Moreno, A. Bugarín and S. Barro, "Evolutionary Learning of a Fuzzy Controller for Mobile Robotics," Soft Computing , vol. 10, Springer Berlin / Heidelberg, 2006, pp. 881-889.
[37] L. Doitsidis, N. C. Tsourveloudis and S. Piperidis, "Evolution of Fuzzy Controllers for Robotic Vehicles: The Role of Fitness Function Selection," Journal of Intelligent & Robotic Systems, vol. 56, Springer Netherlands, 2009, pp. 469–484.
[38] N.C. Tsourveloudis, L. Doitsidis, K.P. Valavanis, "Autonomous navigation of unmanned vehicles: a fuzzy logic perspective," In: Kordic, V., Lazinica, A., Merdan, M. (eds.) Cutting Edge Robotics, 2005, Pro Literatur Verlag, Mammendorf, pp. 291–310.
[39] X. Yang, M. Moallem, R.V. Patel, "A layered goal-oriented fuzzy motion planning strategy for mobile robot navigation," IEEE Trans. Syst. Man Cybern., vol. 35(6), 2005, pp. 1214–1224.
[40] P. Resu, E.M. Petriu, T.M.Whalen, A. Cornell, H.J.W. Spoelder, "Behavior-based neuro-fuzzy controller for mobile robot navigation," IEEE Trans. Instrum. Meas., vol. 52(4), 2003, pp. 1335–1340.
[41] N.C. Tsourveloudis, K.P. Valavanis, T. Hebert, "Autonomous vehicle navigation utilizing electrostatic potentional fields and fuzzy logic," IEEE Trans. Robot. Autom., vol. 17(4), 2001, pp. 490–497.
[42] C. Ye, N.H.C. Yung, D. Wang, "A fuzzy controller with supervised learning assisted reinforcement learning algorithm for obstacle avoidance," IEEE Trans. Syst. Man Cybern., vol. 33(1), 2003, pp. 17–27.
[43] S.-I. Lee, S.-B. Cho, "Emergent behaviors of a fuzzy sensory-motor controller evolved by genetic algorithm," IEEE Trans. Syst. Man Cybern., vol. 31, 2001, pp. 919–929.
[44] S.H. Kim, C. Park, F. Harashima, "A self-organized fuzzy controller for wheeled mobile robot using an evolutionary algorithm," IEEE Trans. Ind. Electron., vol. 48(2), 2001, pp. 467–474.
[45] H. Hagras, V. Callaghan, M. Colley, "Learning and adaptation of an intelligent mobile robot navigator operating in unstructured environment based on a novel online fuzzy-genetic system," Fuzzy Sets Syst., vol. 141, 2004, pp. 107–160.
[46] F. Hoffman, G. Pfister, "Evolutionary design of a fuzzy knowledge base for a mobile robot," Int. J. Approx. Reason., vol. 17(4), 1997, pp. 447–469.
[47] V. Matellan, C. Fernadez, J.M. Molina, "Genetic learning of fuzzy reactive controllers," Robot. Auton. Syst., 1998, vol. 25, pp. 33–41.
[48] D.P.T. Nanayakkara, K. Watanabe, K. Kiguchi, K. Izumi, "Evolutionary learning of a fuzzy behavior based controller for a nonholonomic mobile robot in a class of dynamical environments," J. Intell. Robot. Syst., vol. 32, 2001, pp. 255–277.
[49] S. Nolfi, D. Floreano, "Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines," MIT, Cambridge 2000.
[50] B. Lacevic, J. Velagic, "Evolutionary Design of Fuzzy Logic Based Position Controller for Mobile Robot," Journal of Intelligent & Robotic Systems, vol. 63, Springer Netherlands, 2011, pp. 595-614.
[51] M. Dorigo, M. Colombetti, "Robot shaping : Developing autonomous agent through learning," Artificial Intelligence, vol. 71, 1994, pp. 321-370.
[52] M. Dorigo, U. Schnepf, "Genetics-based machine learning and behavior-based robotics: A new synthesis," IEEE Transactions on Systems, vol. 23, Man, Cybernetics, 1993, pp. 141-154.
[53] J. J. Grefenstette, "Incremental learning of control strategies with genetic algorithms," Proceedings of the Sixth International Workshop on Machine Learning, 1989, pp. 340-344.
[54] J. J. Grefenstette, A. Schultz, "An evolutionary approach to learning in robots," In Proceedings of the Machine Learning Workshop on Robot Learning, 1994.
[55] D. Katagami, S. Yamada, "Interactive classifier system for real robot learning," Proceedings of the 2000 IEEE International Workshop on Robot and Human Interactive Communication, Japan, 2000, pp. 258-263.
[56] F. Gruau, K. Quatramaran, "Cellular encoding for interactive evolutionary robotics," Proceedings of the Fourth European Conference on Artificial Life. The MIT Press/Bradford Books, 1997.
[57] A. Berlanga, P. Isasi, A. Sanchis, et al., "Neural networks robot controller trained with evolution strategies," Proceedings of the International Congress on Evolutionary Computation, 1999, pp. 413-419.
[58] R. A. Watson, S. G. Ficici, J. B. Pollack, "Embodied Evolution: Embodying an evolutionary algorithm in a population of robots," Proceedings of the 1999 Congress on Evolutionary Computation, 1999.
[59] T. Higuchi, H. Iba, B. Manderick, "Applying evolvable hardware to autonomous agents," Parallel Problem Solving From Nature (PPSNIII), 1994, pp. 524-533.
[60] K.C. Tan, L.F. Wang, T.H. Lee and P. Vadakkepat, "Evolvable Hardware in Evolutionary Robotics", Autonomous Robots, vol. 16, Springer Netherlands,2004, pp. 5–21.