



Research Article

Using neuroevolution for designing soft medical devices

Hugo Alcaraz-Herrera^{a,*}, Michail-Antisthenis Tsompanas^{a,b}, Igor Balaz^c,
Andrew Adamatzky^a

^a Unconventional Computing Laboratory, University of the West of England, Bristol BS16 1QY, United Kingdom

^b School of Computing & Creative Technologies, University of the West of England, Bristol BS16 1QY, United Kingdom

^c Laboratory for Meteorology, Physics and Biophysics, Faculty of Agriculture, University of Novi Sad, Novi Sad 21000, Serbia

ARTICLE INFO

Article history:

Received 19 September 2024

Revised 25 November 2024

Accepted 5 December 2024

Available online 19 December 2024

Keywords:

NEAT

HyperNEAT

AFPO

Soft robot

Actuator

Catheter

ABSTRACT

Soft robots can exhibit better performance in specific tasks compared to conventional robots, particularly in healthcare related tasks. However, the field of soft robotics is still young, and designing them often involves mimicking natural organisms or relying heavily on human experts' creativity. A formal automated design process is required. The use of neuroevolution-based algorithms to automatically design initial sketches of soft actuators that can enable the movement of future medical devices, such as drug-delivering catheters, is proposed. The actuator morphologies discovered by algorithms like Age-Fitness Pareto Optimisation, NeuroEvolution of Augmenting Topologies (NEAT), and Hypercube-based NEAT (HyperNEAT) were compared based on the maximum displacement reached and their robustness against various control methods. Analysing the results granted the insight that neuroevolution-based algorithms produce better-performing and more robust actuators under diverse control methods. Specifically, the best-performing morphologies were discovered by the NEAT algorithm.

© 2024 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Although soft robotics is a promising research area with extensive applicability, it entails considerable challenges. One of the most intricate challenges is discovering a proper design: finding the most suitable robot design requires substantial time and material resources, because, it implies numerous prototypes being tested in real life [1]. Unfortunately, for design of soft robots this is not the only complex aspect, since they use flexible materials whose mechanical properties are non-linear and complex to characterise [2]. For instance, *biohybrid machines*, a particular type of soft robots, are built employing biological components, such as cells or tissues, which are inherently difficult to control with conventional methods. This brings another set of challenges, since several parameters need to be considered during the design process and the reality gap problem is becoming more profound.

One promising strategy to counteract these intrinsic challenges of soft robots is using NeuroEvolution (NE). This methodology was previously utilised to design reconfigurable biohybrid machines without strict definitions of specific targets [3]. NE consists of evolving the topology and weights of artificial neural networks (ANNs) through a genetic algorithm (GA). One of the most effective and well-known approaches is NeuroEvolution of Augmenting Topologies (NEAT) [4], which has demonstrated

an adequate performance regarding morphology generation [5]. Moreover, NEAT was extended under the rationale that natural structures are composed of patterns and shape repetition. This extension is called Hypercube-based NeuroEvolution of Augmenting Topologies (HyperNEAT) [6]. This approach evolves a particular type of networks, i.e., Compositional Pattern-Producing Networks (CPPNs), which use, among others, periodic functions, such as sine and cosine, to generate symmetry and patterns that help to evolve topologies [7].

The primary objective of this research is to design from scratch soft actuators that can be implemented in medical devices, such as catheters for targeted drug delivery in areas of the human body that are otherwise difficult to reach. In order to define a trustworthy design engine, NEAT and HyperNEAT are validated as tools to design soft actuator morphologies. We chose the catheter scenario as an exemplary case study for automated morphology generation due to its straightforward design. The inputs for the proposed methodology are the physical characteristics of two different types of material used as building blocks and the constraints that the actuator morphology needs to follow to resemble a catheter actuator. The fitness function of the possible solutions considered is the maximum bending angle that the actuator morphology can reach *in silico*. The output of the evolutionary methodologies is the best combination of the two predefined material building blocks that are composed together inside the predefined design space to produce the actuator morphology. This streamlined focus on bending, which allows other automated

* Corresponding author.

E-mail address: Hugo.Alcaraz@uwe.ac.uk (H. Alcaraz-Herrera).

systems to concentrate on optimising flexibility, ensuring that generated models are both functional and adaptable to various medical scenarios.

NE algorithms were utilised here to evolve the design of soft actuators morphologies under an indirect representation of the three-dimensional space. Specifically, instead of using a bitstream of all possible dimensional segments to encode the characteristics of every point in the available design space (i.e. the type of material used for that point), the NE evolves the artificial networks that when queried with all the points in the design space, they return the material characteristic for each point of the morphology design.

The validation methodology implemented in this research compares the performance of NEAT and HyperNEAT against Age-Fitness Pareto Optimisation (AFPO), an evolutionary algorithm designed to avoid premature convergence [8]. Three metrics are used to assess the performance of the algorithms: (i) the general performance in terms of the displacement observed in the free end of the actuator during a specific time-frame; (ii) the robustness of the designs of actuators produced; and (iii) the total volume of actuators. The fitness of the possible solutions is provided by a physics engine simulating the actuator. The aforementioned metrics and an updated set of constraints applied in the physics engine, present an altered search space for the algorithms which proved to be efficient in previous works of designing soft and biohybrid robots tested on purely the longer distance covered in unconstrained locomotion tasks [3,9].

Whereas the methodology of NE in optimisation of morphologies of soft robots and biohybrid machines was previously studied [3,10,11], the contribution of this work is towards targeting a specific real-world application. Namely, previous works used evolution towards unconstrained locomotion with a generic fitness function (i.e., the longest point reached by a morphology and some variations of that but without any constraints), the present study describe an environment with specific constraints, in order to better reflect the application under study. In specific, here one end of the morphology is fixed, to simulate the insertion point of a drug-delivery catheter. Moreover, while previous works [3] used AFPO with mutation only operators, here we use NEAT and HyperNEAT with mutation and recombination operators to explore in more efficient way the design space of the problem.

The rest of the paper is organised as follows. Section 2 describes background research focusing on soft robot related tasks and in specific utilising NEAT and HyperNEAT algorithms. Section 3 introduces in detail the automated design methodology and the mechanisms of the three algorithms utilised for experimentation in this research. Section 4 describes the experiments performed in the task of designing soft actuators for catheters. Finally, Section 5 concludes this work by presenting insights that emerged from the research and outlines future work.

2. Background

The concept of *soft robots* can be traced back to Isaac Asimov's books published in the 1940s–1950s [12]. However, the first scientific and engineering papers related to soft grippers [13] and soft architecture machines [14] started in the late 1970s. The domain has started to flourish in late 1980s with further publications on soft fingers [15–17], electrostatic actuators [18], and fluid based grippers [19]. Unlike traditional and rigid robots, these robots are built utilising flexible and ductile materials [20,21]. Soft robots have the potential to exhibit better performance in specific tasks that are related to search and rescue [22–25], space exploration [26–29], and healthcare [30–32] due to their mechanisms and morphology, which are inspired by the movement and behaviour of living organisms.

Publications on soft robotic catheters, and relevant surgical devices, started to emerge from late 1980s [33] and by 2020 there was a range of catheters developed for cardio-vascular surgery [34–36], urinary system [37–39], endovascular treatment [40], *in situ* bioprinting [41], endoscopy [42] and drug delivery [43]. Plantoid robots proposed by Mazzolai and colleagues [44–49] can be considered relevant to the research on robotic catheters. These robots are soft-bodied mechanical devices capable of physically growing at the tip (thanks to embedded 3D printing techniques), similar to a plant root, and can change the direction of their growth due to incorporated environmental condition sensors.

The use of soft robots in medical education is also noteworthy. For instance, a recent work [50] implements McKibben artificial muscle on a soft robot inspired by the large intestine and demonstrates similar behaviour to its biological counterpart. This soft robot was designed and 3D printed in a prototype that was tested along with a simulation of finite element analysis to validate its performance. As a result, this soft robot may be employed by medical students to improve their understanding of the functionality of large intestine via optical interaction and touch. Moreover, the same robotic architecture was considered with an updated control mode to achieve higher speeds and loads of object interaction [51]. Drawing inspiration from the movement of inchworm, rather than the intestine peristalsis, enhanced its transport capabilities. The investigation of several of the design and control aspects provided significant insights into the enhanced efficiency and adaptability of its application on object transport tasks.

Several research studies have been conducted to find suitable strategies for soft robot design. For instance, in [52], a large-scale benchmark platform for optimising the design and control of soft robots is introduced. The software is called *Evolution Gym*, and robots can be composed of different types of voxels such as soft, rigid actuators. The benchmark environments contained in *Evolution Gym* provide a wide range of tasks, such as locomotion and object manipulation. Robots are designed in a two-dimensional layout.

Furthermore, another design tool for multi-material soft actuators, *SoRoForge*, is introduced in [53]. This platform comprises the major stages of this type of actuator's design, evaluation, and fabrication process. The software allows: (i) an interactive exploration of implicit geometry functions represented by computational networks; (ii) seeing an online three-dimensional static geometry preview of designs; and (iii) the creation of three-dimensional printable design files.

Another design automation system for soft actuators is presented in [54]. The design engine is the well-known multi-objective evolutionary approach called *Non-dominated Sorting Genetic Algorithm-II* (NSGA-II) [55] that was implemented to find suitable morphologies of soft actuators. Results suggest that under this approach, it is necessary to define adequate fitness functions to find suitable morphologies of soft robots since the ones found in the series of experiments described in this work did not outperform those solutions that were *a priori* manually designed and were used as a baseline.

Evolutionary approaches are not the only methods employed to design soft robots. For instance, in [11], a gradient-based approach, called *novo optimisation*, is utilised to find suitable structures of robots. The algorithm is tested using terrestrial locomotion tasks and shown to be capable of: (i) assessing the fitness of the behaviour of the soft robot; (ii) identifying deficiencies in its overall shape, topology, number and shape of limbs, mass distribution, musculature, and behavioural control; and (iii) simultaneously changing all these aspects which leads to an improvement in evaluation.

In the same path of non-evolutionary approaches, a study introduces a soft robot design methodology based on topology optimisation [56]. The authors implement their methodology to design a soft gripper capable of undergoing free travel and delivering a grasping force. Moreover, the design can be printed using three-dimensional additive manufacturing. However, the optimisation model assumes the homogeneity of materials.

Regarding the two neuroevolution-based algorithms studied here, NEAT has been implemented in numerous studies focusing on soft robots. An example of these studies presents a three-link planar arm model driven by nine muscles, whose purpose is to simulate the human arm [57]. A neurocontroller trained by NEAT controls the muscular-skeleton arm. The training sets applied to NEAT are built considering the forward kinematics, geometry relationships, and muscle mechanic equations. NEAT is compared against a fixed-topology ANN. Results indicate that the neurocontroller trained by NEAT outperforms the traditional ANN in critical aspects such as moving the arm to a specific position and converging the distance between the endpoint and the target point with a minimum distance error.

NEAT has not only been studied for controlling bioinspired mechanisms. It also has been utilised as a design engine for soft robots. For instance, in [58], NEAT, alongside CPPNs, is utilised to design soft robots capable of reaching or squeezing through a small aperture. The experiments consisted of simulations where each soft robot is placed within a cage whose dimensions are $15 \times 15 \times 11$ voxels (arbitrary volume in the physics simulator), leaving a gap of 1 voxel in the x and y axes between the edge of the cage and the maximum size of the soft robot: $11 \times 11 \times 11$. The cage is rigid, immobile and indestructible. Each side has an opening, approximately a circle of diameter 10. Although this research is primarily considered as a proof-of-concept, it is demonstrated that more compliant and deformable soft robots are more capable of reaching or squeezing through small apertures, such as tunnels, than soft robots that are less flexible.

Following the same path of being utilised as morphology generator, in [59], NEAT (and CPPNs) is used to produce three-dimensional physical structures capable of conserving momentum to achieve maximum displacement due to gravity. Structures are composed of spherical cells which fuse to make rigid bodies. The growth process commences with a single cell called *root* and is located at the origin of the structure. A cloud of n points is set around the cell with the n points distributed on the root's surface. Once the cloud is over the cell, every point is used to query a CPPN whose output represents the concentration of matter at that point. A matter threshold determines where to allocate a cell: the more the output value exceeds the matter threshold, the denser the cell allocated at that point will be. Results suggest that the proposed method can produce artefacts that suitably capture the non-obvious relationship between function and physical structure.

Another study where NEAT helps to design morphologies of soft robots capable of navigating across environments with different levels of viscosity is described in [60]. The approach consists of evolving ANNs that observe the virtual environment and respond to it by controlling the muscle force of the soft robot. The performance of morphologies is tested in three different environments: (i) low-viscosity drag; (ii) intermediate-viscosity drag; and (iii) high-viscosity drag. Results indicate that the properties of the environment have a strong influence on soft robot design. For instance, it is possible to observe that the evolutionary pressure flats the parts with big surfaces of the morphologies, which exploits the viscosity drag [60].

NEAT has not only been used for soft robots focused on one task. In [61], this method is utilised to build morphologies of

soft robots capable of performing more than one specific task. The study argues that by combining the genotype of soft robots focused on single-functional tasks is feasible to generate multi-functional robots. The experiments involved in this study consisted of simulating voxel-based creatures performing tasks in terrestrial and aquatic environments. Results suggest that the proposed method: (i) enables the efficient exploration of the morphology search space; (ii) can find morphologies that satisfy two tasks simultaneously faster than existing methods.

On the other hand, HyperNEAT has been primarily utilised in tasks targeting the control of robots. An example of these studies presents a method where morphologies are the input, and the output is neural network controllers capable of working on these different morphologies [62]. Each neural controller is tested in three different quadruped morphologies whose lengths are 0.25, 0.30, and 0.37 m. The neural controllers generated by HyperNEAT are compared against static controllers. Results indicate that HyperNEAT can identify the relationship between morphology and controller architectures during evolution, which leads to a suitable performance.

In the same direction of controller design, in [63] legged robots are studied, and particularly four-legged robots. The rectangular torso of the robot morphology is 0.15 wide, 0.3 long, and 0.5 tall (in arbitrary ODE physics simulator units). Each leg has three cylinders (radius 0.02, length 0.075) and three hinge joints. HyperNEAT is compared against *Fixed-Topology* NEAT (FT-NEAT), also known as Perceptron NEAT. Results indicate that HyperNEAT outperforms FT-NEAT since its morphologies: (i) vastly outperform those generated by FT-NEAT in every generation and (ii) exhibit better coordination than those created by FT-NEAT. Another relevant insight of this research is that generative encodings can outperform direct encodings on regular problems.

Another study, described in [64], focuses on creating gaits for quadruped robots. HyperNEAT is compared against six parameterised learning strategies: (a) uniform random hill climbing; (b) Gaussian random hill climbing; (c) policy gradient reinforcement learning; (d) Nelder–Mead simplex; (e) a random baseline; and (f) a new method that builds a model of the fitness landscape with linear regression. Nine servos actuate the robot used for experimentation, each commanded to a position in the $[0, 1023]$ range, corresponding to a physical range of $[-120^\circ, +120^\circ]$. Two on-board batteries power the servos. Results advocate the superiority of HyperNEAT over all parameterised learning strategies since the gaits it generates exhibit significantly better performance.

Not only standalone neurocontrollers can be evolved through HyperNEAT. In [65], it is stated that it is possible to evolve neurocontrollers encoded by a single genome representing a team of predator agents that work together to capture prey. In the experiments conducted, agents are not aware of their teammates, and due to prey avoidance from nearby predators, it is feasible for one predator to undermine another's pursuit by knocking its prey off its path. Consequently, predators are forced to learn coherent roles that complement the roles of their allies. Results indicate that encoding a team of agents as a unified genome brings benefits such as: (i) critical skills do not need to be re-discovered for separate agents; (ii) due to CPPNs representing multiagent policies, they are assigned to separate agents as a function of their relative geometry and simultaneously exploiting the internal geometries of agents.

HyperNEAT has been scarcely used to design morphologies of soft robots. For instance, in [10], HyperNEAT is employed to evolve controllers and morphologies simultaneously. Robots are evaluated considering their capabilities to adapt in four different scenarios: (i) moving in a flat terrain in a limited time; (ii) moving in a highly uneven terrain; (iii) climbing through a narrow step-wise channel; and (iv) throwing a solid box. Results suggest that

the method generates robots that show a suitable performance in all the aforementioned tasks.

More recently, another study tests the suitability of HyperNEAT (and NEAT) as a design engine for soft robot morphologies, which is described in [9]. The experiments focused on locomotion tasks, particularly reaching the maximum displacement possible in 10 s. Morphologies generated by AFPO are considered as a baseline. Results indicate that both neuroevolution-based approaches can outperform AFPO, and HyperNEAT can perform more suitably than NEAT, despite the absence of geometrical aspects of the problem domain in the design of the substrate. This confirms what is argued in [66]: HyperNEAT can obtain suitable results, even if there is no defined geometrical representation of the problem.

Thus, based on the aforementioned research findings, it is feasible to consider NEAT and HyperNEAT as the core strategy to design adequate morphologies for soft actuators for catheters. Furthermore, considering the three-dimensional nature of the morphologies, Voxelyze is a suitable software to simulate their physical behaviour. Although Voxelyze has limited capacity in capturing complicated physical properties (i.e., does not take into account fluid dynamics), it was selected as an efficient alternative against more complicated simulators. The basic functionality (i.e., interactions between adjacent voxels, volumetric actuation and collision detection) is considered adequate under the trade off between a lower reality gap and a computational efficient evolutionary optimisation. Nonetheless, all candidate solutions in the evolutionary optimisation will undergo the same inaccuracies imposed by the simulator. Moreover, Voxelyze has demonstrated adequate accuracy in simulating biohybrid machines [3] that exhibit inherently high levels of behavioural uncertainty. This has been achieved with only one round of fine-tuning based on *in vitro* observations, aligning simulations more closely with physical prototypes.

In the medical domain, however, the accuracy required for final designs is paramount, as high fidelity in biohybrid actuator performance is essential. Therefore, before advancing any selected design to manufacturing prototyping, these preliminary designs should undergo a secondary stage of rigorous testing in simulations that account for more precise physics-based properties, including fluid dynamics and nuanced biological interactions. By coupling our approach with a more detailed simulator at next developmental stage, we can better approximate real-world performance, mitigating risks associated with physical prototyping and ensuring that any final designs meet the strict reliability and performance standards needed for medical applications.

Some previous studies focus in a two-dimensional layout design of soft robots [10,52], the present work aims towards more realistic three-dimensional designs, thus, Voxelyze is utilised. Whereas some non-evolutionary approaches were previously proposed for soft robot design [11,56], the utilisation of evolutionary methods is preferred here, due to their simplicity and enhanced capacity to avoid local optima in rugged optimisation problems. Nonetheless, the direct representation of individuals in evolutionary soft robot design techniques has proved not as efficient as generative representation (i.e. CPPNs) [67]. The implementation of AFPO has proved very efficient when employed in the designing process of biohybrid machines [3]. However, it was previously applied using only a mutation operator, without crossover, which limited the method's exploratory capacity. The authors of the prior study recommended incorporating recombination-based methods, such as NEAT and HyperNEAT, for future improvements. Finally, the implementation of NE algorithms have been tested in specific applications, namely achieving the longest distance on an unconstrained plane or with some obstacles located on the plane and object manipulation [3,10,11,67]. Here, we study the application of NE algorithms on a more reserved and specialised application (namely, the angular movement of a soft actuator enabling a medical catheter).

3. Methodology

The design of soft actuators in a given three-dimensional space ($[x, y, z]$) was produced automatically via an evolutionary process over an indirect representation and evaluated through a physics engine (i.e. Voxelyze) to fit the initially set target. The target (or the fitness function) here was set to achieve the higher possible bending angle, since, the application target is a drug-delivering catheter. To represent the insertion point of the catheter, one end of the morphology was fixed and not able to move, thus, emulated an anchoring point for the actuator. Moreover, instead of using a direct representation in the evolutionary optimisation for the design of the soft actuator, namely a one-to-one correspondence of all possible positions in the three-dimensional design space with their physical characteristics, an indirect representation was preferred. The indirect representation used in all of the algorithms described in the following, is constituted by neural networks, that have as input all the possible combination of the three-dimensional space ($[x, y, z]$) and as outputs the physical characteristics of the material comprising the actuator morphology in the corresponding area.

The fitness function consists of a physics engine called *Voxelyze* (also available in [68]), which simulates the physical response of morphologies under determined conditions. Morphologies are defined within *Voxelyze* by stacking voxels (the minimum building block in *Voxelyze*) of different materials together. Here, two types of voxels were considered for the simplicity of the designing landscape: one active and one passive voxel, which can represent the response of contractile muscles and immotile cells, respectively. The physics simulation parameters for both these building blocks were set to Young's modulus of $5 * 10^6$ Pa, Poisson's Ratio of 0.35, and coefficients of static and dynamic friction of 1.0 and 0.5, respectively. Moreover, the active voxel is characterised by volumetric actuation of $\pm 50\%$ of the volume at rest at 4 Hz. These parameters were selected based on previously published and *in vitro* validated works [3], but do not limit the generalization of the algorithmic methodologies.

The original version of *Voxelyze* was modified (changes in the source code available in [69]) to trace the position of the catheter actuator in the x, y, z axes during a time t . Furthermore, one end of actuators is fixed in such a way that they only move vertically (i.e., one degree of freedom) in the yz plane. In addition, actuators have a passive enclosure to follow the findings of previous simulation works [70] and constraints of laboratory experiments, i.e. the bioreactor that a muscle actuator requires for the supply of nutrients in case of a biohybrid machine. The output of *Voxelyze* contains three values that describe the position of the actuator free tip (with respect to x, y, z dimensions) for the initial position and the final one (after the predefined simulation time), and the number of voxels that compose the actuator.

Three algorithms were utilised in this research. Namely, Age-Fitness Pareto Optimisation (AFPO), NeuroEvolution of Augmenting Topologies (NEAT), and Hypercube-based NeuroEvolution of Augmented Topologies (HyperNEAT), which will be described in Section 3.1, Section 3.2, and Section 3.3 respectively.

3.1. AFPO

This approach was conceived as an alternative method to avoid premature convergence in some evolutionary algorithms [71]. One of the fundamental aspects of AFPO is *age*, which, under this scope, can be understood as how long genotypic material has existed in the population [8]. Based on this, the age of a solution (i.e., an individual) is measured in generations.

The algorithm commences initialising the individuals composing the population at random, with age of all individuals set to

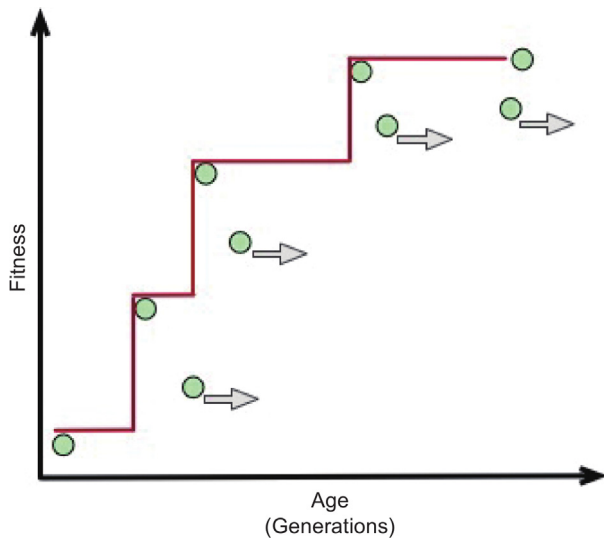


Fig. 1. An example of a population moving in a two-dimensional Pareto space where the vertical axis is fitness and the horizontal axis is age. Figure adapted from [71].

one. If an individual survives to the next generation, its age is incremented by one. When genetic operators (i.e., crossover and mutation) take place, the age is inherited as the maximum age of the parents.

Since AFPO employs a single population, it tracks each individual's fitness similarly to a standard evolutionary algorithm, including the genotypic age. This multi-objective optimisation procedure aims to determine the non-dominated Pareto front of the problem domain, where the objective is maximising fitness with minimum age. One way of visualising the mechanism of AFPO is the population (i.e., individuals) evolving in a two-dimensional plane of age and fitness. Fig. 1 illustrates this dynamic.

3.2. NEAT

This algorithm was designed to counteract on three main issues that previous NE algorithms exhibit: (i) a solution representation capable of allowing arbitrary topologies to recombine; (ii) avoiding the premature disappearance of topological novelties discovered throughout the evolutionary process; and (iii) avoiding complex topologies without using a fitness function focused on punishing topological complexity [4].

The first issue was tackled using *gene tracking by historical markings*. This is achieved with a list of connection genes which represent a connection between two nodes containing: (a) the "origin" node and the "destination" node, (b) the weight of the connection, and (c) a Boolean state of the connection (enabled or disabled). Moreover, each gene has one *global innovation number*, a unique numerical ID and the core of the crossover operator.

Two genes having the same global innovation number (also known as "historical origin") represent the same structure; therefore, the genes of both genomes that have the same innovation number are aligned. The genes that do not match are inherited from the fittest genome or chosen at random. When a new gene is generated by mutation, the global innovation number increases and is associated with the new gene. Innovation numbers are generally the chronology of every gene during the evolutionary process. Fig. 2 presents an example of a genotype (i.e., a neural network) and its associated phenotype.

The second issue, was addressed by *speciation to protect innovation*, which generates species within the population aiming at the interaction of individuals that belong to the same species. NEAT creates species utilising topological similarities that are identified by the use of historical markers. The canonical way to measure compatibility of two genomes (i.e., individuals) is by the number of excess and disjoint genes. Thus, the less disjoint genes are, the more compatible the two individuals are. The compatibility distance (δ) between two genomes is as follows [72]:

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \times \overline{W} \quad (1)$$

where E is the number of excess genes, D represents the disjoint genes, \overline{W} is the mean weight differences of matching genes, c_1 , c_2 , c_3 are utilised to determine the relevance of E , D , and \overline{W} , respectively. Regarding N , it represents the number of genes in the individual with the largest number of genes; it is used to normalise the genome size during the evolutionary process. The distance δ determines in which species an individual is allocated utilising a compatibility threshold δ_t . Individuals are allocated to the first species (taken at random) where their distance δ is $< \delta_t$, assuring that individuals do not belong to more than one species.

NEAT evaluates individuals utilising *explicit fitness sharing*. Under this schema, individuals within the same species share the fitness, which implies a limitation for species to expand over the population [73]. Since this mechanism adjusts the fitness of individuals by dividing an original fitness by individuals in each species, the species grow or shrink if their average adjusted fitness is either over or under the population average.

The third issue was alleviated by *minimising network structures*, which is feasible due to the mechanism of speciation to protect innovation. Since NEAT can initialise individuals whose input neurons are fully connected to the output neurons (i.e., no hidden nodes are involved), new structures are incrementally generated when structures are mutated during the evolutionary process. The structures that exhibit a suitable performance can survive. The effect induced by this approach narrows down the search towards a minimal number of weight dimensions, which implies: (a) a reduction of runtime to find solutions [74] and (b) avoiding the generation and evaluation of complex structures during the evolutionary process.

3.3. Hyperneat

This approach is an extension of NEAT [6], which utilises the evolutionary engine of the aforementioned algorithm to evolve a particular class of neural networks known as CPPNs [7]. CPPNs are used due to their ability to generate patterns, such as repetition and symmetry. There exist two key differences between NEAT and HyperNEAT:

- i. *Activation functions*. Typically, NEAT generates ANNs containing hidden nodes exclusively using the sigmoid function. HyperNEAT, nevertheless, allows the use of other activation functions, such as Gaussian, trigonometric, and periodic, in the nodes composing CPPNs. In this way, CPPNs can evolve, exploring a significantly more expansive search space of network functionality.
- ii. *Substrate*. HyperNEAT is capable of embodying the geometry of the domain problem due to the nature of CPPNs; therefore, the topology of ANNs is computed, taking into account their geometry. Thus, the geometric layout where HyperNEAT operates is called a *substrate*, which can be configured in diverse manners. One of the most popular

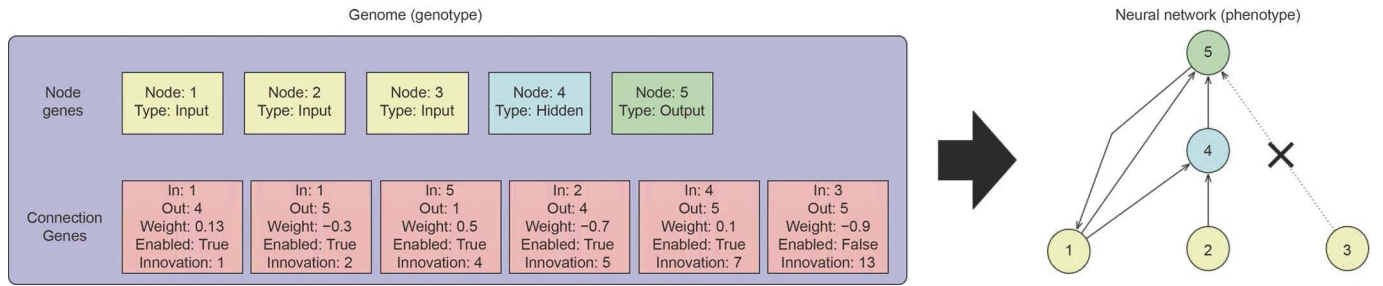


Fig. 2. Example of a genotype and its phenotype under NEAT. Figure adapted from [72].

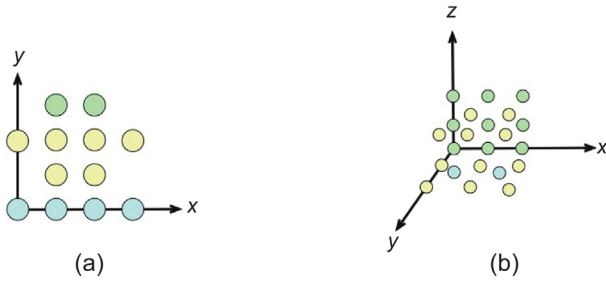


Fig. 3. Two typical configurations of substrates utilised in HyperNEAT: (a) two-dimensional, and (b) three-dimensional.

configurations consists of a set of nodes allocated in a two-dimensional plane, known as *grid*. Another widely used configuration is the *three-dimensional grid*, which contains a set of nodes allocated in a three-dimensional space. Fig. 3 (a) depicts an example of a grid, whereas Fig. 3 (b) presents an example of a three-dimensional grid.

Under HyperNEAT, topologies of ANNs embodied in the substrate are generated by CPPNs employing the position of the neurons. Consequently, when a two-dimensional substrate is used, the weight between the neuron m and the neuron h is calculated using their position in the substrate:

$$CPPN(x_m, y_m, x_h, y_h) = weight_{mh} \quad (2)$$

Regarding the bias of neurons, they are computed, providing the coordinates of the neuron in the “origin” position, whereas the “destination” coordinates are set to zero. Thus, the bias of neuron m is obtained by

$$CPPN(x_m, y_m, 0, 0) = bias_m \quad (3)$$

Furthermore, if a three-dimensional grid is being utilised, the weight between the neuron h and the neuron m is computed as follows:

$$CPPN(x_h, y_h, z_h, x_m, y_m, z_m) = weight_{hm} \quad (4)$$

The bias of neurons is calculated analogously to the manner described in Eq. (3). Therefore, the bias of neuron h is calculated by:

$$CPPN(x_h, y_h, z_h, 0, 0, 0) = bias_h \quad (5)$$

In general, the mechanism of HyperNEAT starts by defining the substrate. The next step consists of generating the initial population of CPPNs. Then, for each CPPN, the connections (i.e., weights) and bias of the substrate are calculated using Eqs. (2) and (3) for a two-dimensional grid, and Eqs. (4) and (5) for a three-dimensional grid. Once the substrate is built, its performance (and hence the performance of the CPPN) is evaluated, and a fitness score is assigned to the CPPN. After the evaluation stage, NEAT

Table 1

Parameters utilised to evolve CPPNs under NEAT and HyperNEAT for the actuator generator domain.

| Parameter | Value |
|---|---------|
| Compatibility threshold | 3 |
| Compatibility disjoint coefficient | 1.0 |
| Compatibility weight coefficient | 0.5 |
| Maximum stagnation | 25 |
| Survival threshold | 0.6 |
| Activation function mutate rate | 0.4 |
| Adding/deleting connection rate | 0.3/0.2 |
| Activating/deactivating connection rate | 0.5 |
| Adding/deleting node rate | 0.3/0.2 |

is applied to evolve the population of CPPNs. It is important to emphasise that the outputs of CPPNs are usually in the $[-1.0, 1.0]$ range, which, in case it is necessary, implies a normalization or mapping procedure.

Summarising, HyperNEAT aims to evolve the topology (i.e., the connectivity) and weights of ANNs that are represented by a substrate with a specific geometry through CPPNs.

4. Experimental setup and results

In this research, three metrics are used to determine the suitability to generate soft actuator morphologies of NEAT and HyperNEAT: (i) the general performance in terms of the displacement of the actuator in the yz plane during a simulated time period; (ii) the robustness of the fittest actuators found; and (iii) the number of voxels composing the actuator. Furthermore, AFPO (see Section 3.1) is utilised as a baseline actuator generator in these experiments, and the implementation employed is available on GitHub [68] and is fully described in [3].

4.1. Experimental setup

The population is composed of 100 individuals (i.e., CPPNs) for the three approaches. Each evolutionary run lasted 3000 generations. The set of activation functions utilised for experimentation is the same as used in [3], and is composed of: *sigmoid*, *sine*, *negative sine*, *square*, *negative square*, *square root of absolute*, *negative square root of absolute*, *absolute*, and *negative absolute*. Activation functions are randomly chosen during CPPN initialization and mutations. In order to perform a fair comparison for all approaches, the implemented initialization procedure of the population is described in [3]. Moreover, for NEAT and HyperNEAT, the parameters employed for the evolutionary process of CPPNs are shown in Table 1. For each approach, 20 independent experimental runs were performed, by changing only the initial random populations.

Regarding the dimensions of actuators, in terms of voxels, there are 20 units in the x axis and 8 units in the y and z axes. Thus, the axes range of the three-dimensional layout where

morphologies are designed are as follows: for x axis, $[0, 20]$. For y and z axes, $[0, 8]$.

Since simulations require a significant amount of computational time, it is essential to mitigate this limitation. Thus, the implementation related to AFPO was previously written under the multiprocessing paradigm [3] and was used here under the same principle. The implementation for NEAT and HyperNEAT was designed under a client-server architecture to take advantage of distributed computing capacities [9]. The hardware utilised for experimentation in this study is described as follows: *Processor*: ARM (virtualised), nine cores (18 threads), 3.20 GHz, *RAM Memory*: 16 GB, LPDDR5. Provided that the evolutionary optimisation is implemented under a client-server architecture, the utilisation of High-performance computing (HPC) can significantly accelerate this methodology, so that it complies with time constraints of real-world optimisation problems. Additionally, some alternations in the algorithmic implementation, like the initialization methodology, have proved to contribute with an acceleration of 30% on the computations [74].

4.1.1. NEAT configuration

Due to actuators being designed in a discrete three-dimensional layout, it is necessary to provide for each point across the layout: (i) the presence or not of a material voxel, and (ii) the material type of each voxel. Thus, under NEAT, CPPNs are queried as follows:

$$CPPN(x_i, y_i, z_i) = v_i, m_i \quad (6)$$

where the tuple (x_i, y_i, z_i) represents the coordinates of the i -th point in the three-dimensional layout, v_i , refers to the presence of a voxel in the i -th point of the layout, and it is processed as the presence or absence of a voxel by the following equation:

$$Presence(x_i, y_i, z_i) = \begin{cases} \text{yes,} & |v_i| \geq 0.5 \\ \text{no,} & \text{otherwise} \end{cases} \quad (7)$$

Furthermore, m_i represents the type of material of the voxel in the i -th point of the layout. In the scope of this research, two types of materials are considered: (i) *passive*, encoded as 1, and (ii) *contractile*, encoded as 3. Eq. (8) presents the mechanism to map m_i as an encoded material ID.

$$Material\ code(x_i, y_i, z_i) = \begin{cases} 1, & |m_i| < 0.5 \\ 3, & \text{otherwise} \end{cases} \quad (8)$$

4.1.2. HyperNEAT configuration

The first element of HyperNEAT to determine is the substrate. Thus, based on the discrete three-dimensional layout used to design actuators, the substrate has three neuron inputs. Furthermore, since the data required to determine are the presence of a voxel in the i -th point and the material of that voxel, the amount of output neurons is two. Eq. (9) presents how substrates are queried.

$$substrate(x_i, y_i, z_i) = v_i, m_i \quad (9)$$

where the tuple (x_i, y_i, z_i) is the coordinates of the i -th point in the three-dimensional layout. The v_i variable is associated with the presence or absence of a voxel in the i -th point of the layout and is mapped using Eq. (7), and m_i is related to the type of the voxel material in the point i of the layout and is processed by Eq. (8). Fig. 4 depicts the design of the two-dimensional substrate used for experimentation. The neuron allocation used for the substrate was decided by conducting a series of experiments varying the number of neurons per hidden layer and the number of hidden layers in the $[1, 5]$ range. Furthermore, the activation function implemented in the substrate is *SELU* due to its fast convergence and self-normalising properties [75].

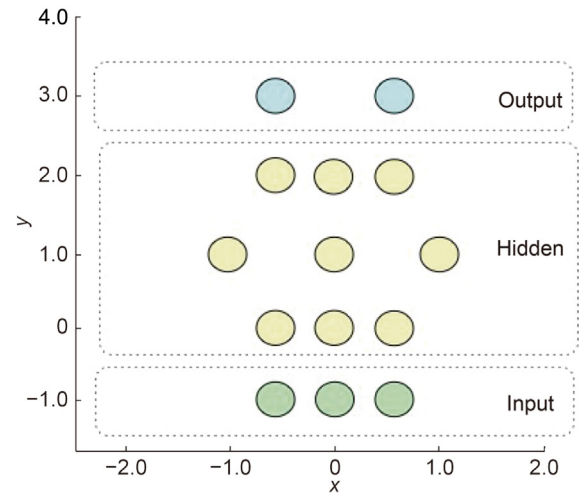


Fig. 4. Substrate employed under HyperNEAT for actuator generator.

Once the substrate is defined, the following step defines how CPPNs are queried. Since the substrate was defined in a two-dimensional space, CPPNs are queried utilising Eq. (2) for calculating the weights among neurons, and Eq. (3) to compute the bias of the neurons allocated in the substrate. It is essential to point out that based on the definition of HyperNEAT [76], determining whether a connection is calculated is by a threshold. Thus, if $|weight_{mh}| > 0.2$, it is normalised in the $[-3.0, 3.0]$ range. Otherwise, there is no connection between the neuron m and the neuron h .

4.2. Analysis of the performance

In order to validate the suitability of catheter actuators generated by AFPO, NEAT and HyperNEAT, three experiments are conducted: (i) analysing the general performance in terms of finding the actuator with the maximum displacement possible (i.e., the fittest) in the yz plane; (ii) testing the robustness of the fittest actuators found; and (iii) identifying the fittest actuators with the minimum volume (i.e., the minimum number of voxels) possible. It is essential to highlight that during all experimental stages, the displacement was measured in terms of the length of voxels.

4.2.1. Finding the fittest actuator

One vital feature of actuators is the displacement in the yz plane, since they have one end fixed. This displacement can be understood as an indicator of the bending capacity exhibited by actuators: the more displacement the actuator reaches, the more bendable the catheter is.

For instance, if a soft actuator exhibits no displacement despite a determined stimulation, the angular range where it can operate is $[0, \alpha]$, where $\alpha = 0^\circ$, which can be interpreted as the soft actuator presents a poor performance. On the other hand, if the movement observed in the soft actuator allows to satisfy $\alpha > 0^\circ$, then the soft actuator can operate in a more comprehensive angular range (i.e., a better performance). In general, the bigger the displacement of the soft actuator (and hence α) is, the more comprehensive the angular range where the soft actuator can operate.

In this experiment, the evaluation of actuators is as follows: for each independent experimental run, 25 different controller strategies (or phase offsets in the oscillations of the simulated active voxels) are generated at random and do not change throughout the evolution. For each morphology of the population, the 25

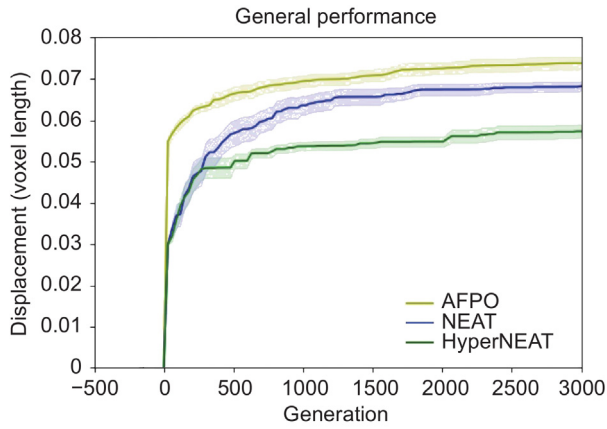


Fig. 5. Mean performance of AFPO, NEAT, and HyperNEAT showing $\pm 95\%$ confidence interval (shaded region).

controller phase offsets previously generated are used to stimulate the displacement of the actuator. In other words, each morphology is stimulated 25 times with different controller phase offsets. This procedure occurs for each generation until the experimental run is complete. Thus, the aptitude apt of the actuator a is calculated as follows:

$$apt_a = \frac{\sum_{i=1}^{25} displacement_i}{25} \quad (10)$$

It is important to emphasise that this experiment does not consider the number of voxels as a penalty for the aptitude value.

Fig. 5 shows the mean performance in finding the fittest individual (i.e., the fittest actuator) with 95% confidence intervals depicted by the shaded regions across 20 experimental runs under AFPO, NEAT and HyperNEAT. In general, AFPO exhibits a significantly better performance than both neuroevolution-based approaches. Furthermore, NEAT performs better than HyperNEAT. All the data gathered were tested and are not normally distributed (Shapiro–Wilk test; $p < 0.01$). Using the Kruskal–Wallis test, it is feasible to confirm that significant differences exist among the performance of the three approaches ($p < 0.01$). Therefore, a ranking in terms of performance can be performed: AFPO $>$ NEAT $>$ HyperNEAT (Dunn’s test: $p < 0.01$).

These results suggest that under 25 different controller phase offsets, AFPO can find the fittest actuators morphologies than those found by NEAT and HyperNEAT. Moreover, HyperNEAT’s unexpected performance may be influenced by the absence of domain geometric elements that can be embodied in the substrate design.

4.2.2. Exploring the robustness of the fittest actuator

A catheter could be suitable to deliver drugs in numerous angles under a specific scenario (i.e., a specific controller phase offset of each active voxel composing the actuator). However, it might not exhibit similarly high performance if the scenario changes. This experiment focuses on testing the robustness (i.e., how suitable the performance is regardless of the external conditions) of the three fittest actuators found across the 20 experimental runs conducted using AFPO, NEAT and HyperNEAT.

Each top actuator generated by all approaches is tested utilising 1000 different controller phase offsets of each active voxel. In other words, the controller phase offset scenarios were randomly generated and used by all the morphologies studied in this experiment (i.e., the three fittest actuator morphologies generated

by each approach). Thus, the robustness (i.e., the aptitude apt) of the actuator a can be measured as follows:

$$apt_a = \frac{\sum_{i=1}^{1000} displacement_i}{1000} \quad (11)$$

Fig. 6 presents violin plots comparing the performance (i.e., the displacement observed in the yz plane of the free end of the morphology) of the three fittest actuators found by AFPO, NEAT and HyperNEAT. Each violin plot exhibits the maximum, minimum, median and kernel density estimation of the frequency distribution of values under 1000 different offset controller scenarios.

When AFPO morphologies are considered (see Fig. 6 (a)), all morphologies exhibit significant differences, a fact that is determined by testing all the data collected, and realising that they are not normally distributed (Shapiro–Wilk test; $p < 0.01$). Then by the Kruskal–Wallis test, it is possible to confirm significant differences among the displacement observed in the three actuators ($p < 0.01$). Thus, a performance ranking can be performed: Morphology 12 $>$ Morphology 19 $>$ Morphology 7 (Dunn’s test: $p < 0.01$).

Under NEAT, all morphologies exhibit significant differences, and the data gathered are non-normal (Shapiro–Wilk, Kruskal–Wallis; $p < 0.01$). Based on the previous result, it is feasible to produce a ranking of the performance of the three fittest actuators generated by NEAT: Morphology 9 $>$ Morphology 16 $>$ Morphology 15 (Dunn’s test: $p < 0.01$).

On the other hand, when HyperNEAT results are analysed, the three fittest morphologies generated show significant differences and non-normality is observed in the data gathered (Shapiro–Wilk, Kruskal–Wallis; $p < 0.01$). Therefore, a ranking focused on performance can be assumed: Morphology 2 $>$ Morphology 18 $>$ Morphology 11 (Dunn’s test: $p < 0.01$).

To conclude, Fig. 7 shows violin plots comparing the performance of the fittest actuator morphology produced by AFPO, NEAT and HyperNEAT. Each violin presents the kernel density estimation of the frequency distribution, maximum, minimum and median values of the 1000 controller phase offset scenarios. All the morphologies present significant differences, confirmed through the Kruskal–Wallis test ($p < 0.01$). Based on this result, it is possible to rank the performance of the actuators as follows: NEAT $>$ HyperNEAT $>$ AFPO (Dunn’s test: $p < 0.01$).

In general, results suggest that neuroevolution-based approaches, such as NEAT and HyperNEAT, are more adept at producing bendable actuators than those generated by AFPO, regardless of the controller phase offset scenario. This realization comes as a contradiction to the previous test, where AFPO seemed as the most appropriate method, whereas, only 25 phase offsets were tested during the optimisation evolutionary process. This implies that NEAT and HyperNEAT have the potential to create more robust actuators than AFPO. Furthermore, despite the comparable performance of NEAT and HyperNEAT, the former has the edge in producing more robust and, consequently, more bendable actuators. As mentioned previously, HyperNEAT can prove advantageous over problems that have a strong geometric regularity, based on its ability to evolve network structures over geometric substrates. On the other hand, the NEAT algorithm provided better results in this robustness analysis. This can be attributed to two factors. Firstly, the HyperNEAT substrate was only fine-tuned in a limited range of (i) hidden layers and (ii) neurons per hidden layer (i.e., [1,5] for both), because of the high computational requirement related with testing higher ranges. As a result, the best substrate configuration for the given problem can be more complicated than the ones tested within this study. Secondly, NEAT is based on evolving CPPNs (CPPN-NEAT), that have been proved to be very efficient in emerging behaviours that can facilitate movement [67].

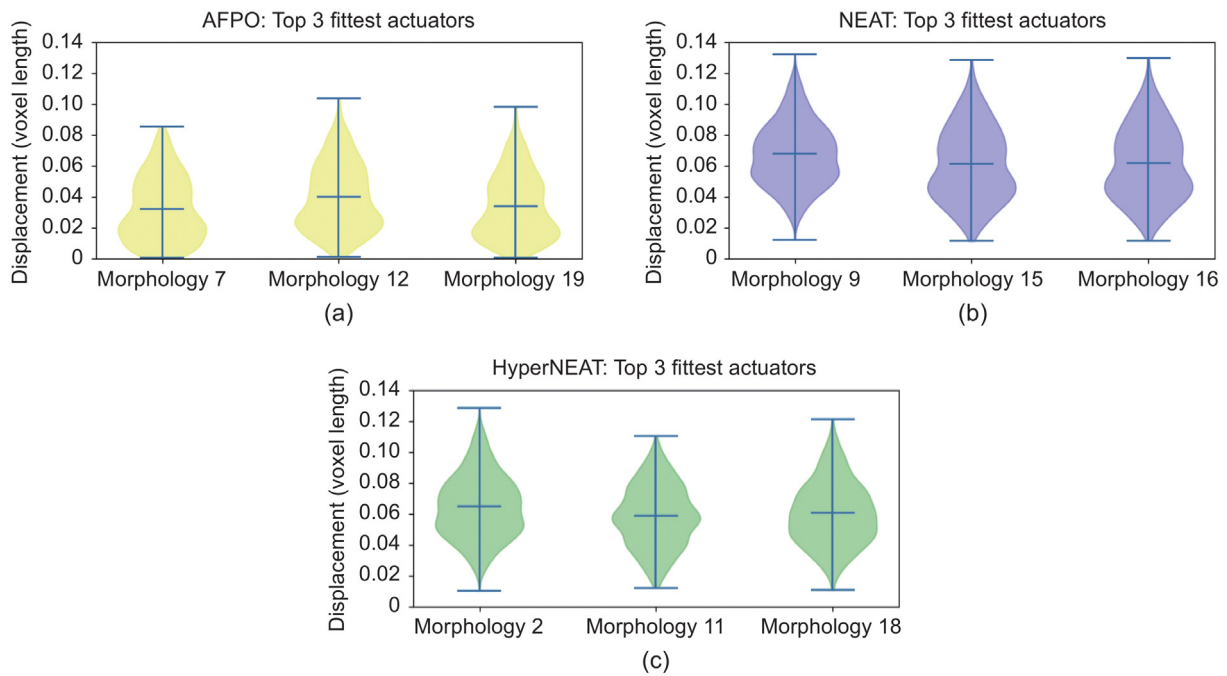


Fig. 6. Displacement observed in the yz plane under 1000 different controller phase offset controllers of the top three fittest actuators found in 20 different experimental runs under: (a) AFPO; (b) NEAT; and (c) HyperNEAT.

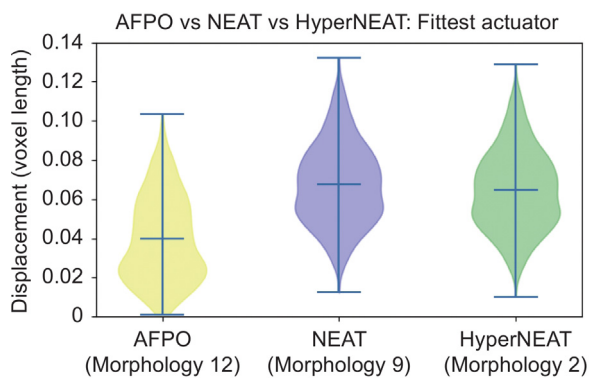


Fig. 7. Fittest actuator under: AFPO (left), NEAT (centre), and HyperNEAT (right).

4.2.3. Identifying the minimum volume of the fittest actuators

Since the morphologies analysed in this research should inform the design of devices in real life whose objective is to manoeuvre within narrow parts of the human body (e.g., blood vessels), the scale of the manufacturing must be significantly minuscule. From this point of view, the fewer voxels the actuator has, the more suitable the catheter employing that actuator. Table 2 presents the number of voxels (passive and contractile) composing the fittest actuators produced by AFPO, NEAT, and HyperNEAT.

Even though AFPO produced the actuator with fewer voxels, its performance is less suitable than the approaches based on NE. In contrast, HyperNEAT produced an actuator whose performance is acceptable; however, the number of voxels, particularly the contractile, is significantly high. Regarding the actuator generated by NEAT, it presents a suitable trade-off between performance and the number of voxels since its performance is the best among the three, and the number of voxels is lower than the number of voxels of the actuator designed by HyperNEAT.

Table 2

Number of passive and contractile voxels of the fittest morphologies produced by AFPO, NEAT, and HyperNEAT.

| Approach | Total voxels | Passive voxels | Contractile voxels |
|-----------|--------------|----------------|--------------------|
| AFPO | 900 | 632 | 268 |
| NEAT | 1063 | 632 | 431 |
| HyperNEAT | 1253 | 632 | 621 |

For illustration reasons, Fig. 8 depicts the fittest actuator morphologies generated by all three approaches. The fittest actuator designed by AFPO (see Fig. 8 (a)) exhibits a pyramidal pattern consisting of having a few contractile voxels at the bottom and gradually incrementing the number of voxels towards the top of the actuator. Arguably, the lack of contractile support at the bottom of the actuator affects its bending properties. Furthermore, the fittest actuator generated by NEAT (see Fig. 8 (b)) presents a striped diagonal pattern with no voxels throughout the contractile body composing the actuator. This pattern possibly allows the catheter actuator to bend suitably regardless of the controller phase offset. Finally, the fittest actuator created by HyperNEAT (see Fig. 8 (c)) also exhibits a striped diagonal pattern with no voxels, only in the free end of the actuator. The striped pattern allows the actuator to bend; however, the solid contractile mass on the other end of the actuator arguably attenuates the bending movement.

To sum up, the NE-based methodologies have produced the most robust actuator morphologies, that directly relates to a high chance of performing efficiently on real-world applications. The random phase offsets used in this study can be regarded to assess morphologies for noise resistance in the simulations that is a way to trivially predict the success of their physical counterpart when manufactured in a real-world application [77]. The superior performance of AFPO on finding the fittest morphology can be preferred only in cases that the actuators can be controlled with high accuracy, due to the small variety of phase offsets tested throughout the evolutionary optimisation (for computational efficiency reasons). The proposed methodology can

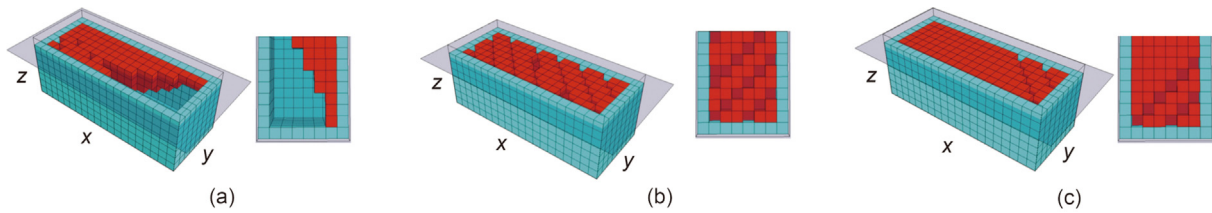


Fig. 8. Morphology of the fittest actuator under: (a) AFPO; (b) NEAT; and (c) HyperNEAT. Blue voxels represent passive tissue; red voxels represent contractile tissue.

be instantiated with the parameters of the building blocks (or physical characteristics of available materials) to be utilised for the soft actuator manufacturing process, the physical constraints of the problem, the physical description of the environment that will accommodate the actuator and a target behaviour (or fitness function). Then, through the evolutionary optimisation of the NE-based algorithms, an efficient morphology of the actuator will be produced, that will be in the form of what type of material should be placed at each point in the three-dimensional design space.

4.3. Error analysis

To understand the variability of the performance of actuators and, consequently, of the algorithms used in this research, two experiments are performed: (i) varying the properties of two simulated materials and (ii) altering the control strategy used in previous experiments. It is important to emphasise that the fittest actuator morphology found in the experiments described in Section 4.2.2 was utilised for the experiments described in the following sections. Namely, Morphology 9, found by NEAT (N-M9).

4.3.1. Varying material properties

This experiment changes the value of two mechanical properties of the contractile material used for the actuator morphologies during simulations. These two properties are:

- i. *Young's modulus*. It describes the stiffness of materials. This property is related to the elasticity of materials under compression or tension [78].
- ii. *Poisson's ratio*. It is the negative ratio of transverse strain to axial strain. In other words, this property is the deformation of a material perpendicular to the load direction [78].

During the previous experiments (see Section 4), Young's modulus and Poisson's ratio were set to 5×10^6 and 0.35, respectively. In this experiment, each value is varied in the following range: $[-10\%, -5\%, 5\%, 10\%]$.

Fig. 9 depicts violin plots comparing the displacement exhibited (i.e., the performance) by N-M9 varying the values of the Young's modulus and the Poisson's ratio. Each violin plot shows the median, minimum, maximum and kernel density estimation of the frequency distribution under 1000 different offset controller scenarios.

When the value of Young's modulus is altered (see Fig. 9 (a)), significant differences can be observed among the performances. All the data gathered are non-normally distributed (Shapiro-Wilk; $p < 0.01$). It is possible to confirm significant differences between the performances whose Young's value is $< 5 \times 10^6$ (Dunn's test; $p < 0.01$). Furthermore, significant differences are exhibited when these two performances are compared against those whose Young's value is $\geq 5 \times 10^6$ (Dunn's test; $p < 0.01$). However, there are no significant differences among the performances whose Young's value is $\geq 5 \times 10^6$ (Dunn's test; $p > 0.05$).

As expected, the results indicate that the material's stiffness significantly influences the actuator morphology's performance.

In general, the less stiff the material is, the less controllable it is, and hence, the less performance is observed. On the other hand, if the material is stiffer, there is no significant improvement in performance.

Furthermore, when the value of the Poisson's ratio is varied, there are no significant differences among the performances. This is confirmed by first testing the collected data using the Shapiro-Wilk test and obtaining $p < 0.01$, which indicates that the data is non-normally distributed. Then, applying Dunn's test, the result is $p > 0.05$, suggesting that the deformation of the material does not significantly affect the performance of the actuator morphology.

4.3.2. Testing the control strategy

In this experiment, a random controller phase offset is used. This controller phase offset is copied 18 times. Each copy represents a scenario where a *slice* in the plane yz of N-M9 is affected. The controller phase offset values associated with the voxels composing the slice are set to 0. It is essential to point out that although the dimension of N-M9 in the x axis is 20, only 18 slices are available due to the passive enclosure embodied (see Section 3). For visual reference, see Fig. 8 (b). Furthermore, the value of Young's modulus and Poisson's ratio are set to 5×10^6 and 0.35 respectively (see Section 4.3.1).

Fig. 10 presents a bar plot showing the displacement reached by N-M9 when the values of the controller phase offset associated with n slice are set to 0 (blue bars). In addition, a green bar shows the performance of N-M9 with no modification in the values of the controller phase offset; it is used as a baseline.

As expected, no significant variation in terms of performance is observed in the slices that are near the fixed end of N-M9. This behaviour occurs from slices 1 to 14. Furthermore, those slices that are near the free end of N-M9 exhibit a significant variation, which is observed in slices 16 to 18.

In general, despite the specific noise applied to the controller input, the performance of the actuator morphology is consistent (i.e., there are no noticeable variations) in the vast majority of scenarios. On the other hand, the variations in performance observed are arguably due to external interactions that affect the material's mechanical properties, such as the absence of physical support (i.e., no fixed end).

5. Conclusions

This research studied the suitability of NEAT and HyperNEAT as generators of soft actuator morphologies. Their capabilities were compared against an implementation of AFPO, a popular multi-objective optimisation approach. The performance analysis was done utilising three metrics: (i) analysing the general performance to find actuators with the maximum displacement possible in the yz plane; (ii) testing the robustness of those actuators exhibiting the maximum displacement reached (i.e., the fittest actuators); and (iii) determining the fittest actuators with the minimum number of voxels possible. For this analysis, 20 different actuators were generated (i.e., 20 experimental runs were conducted) by each approach.

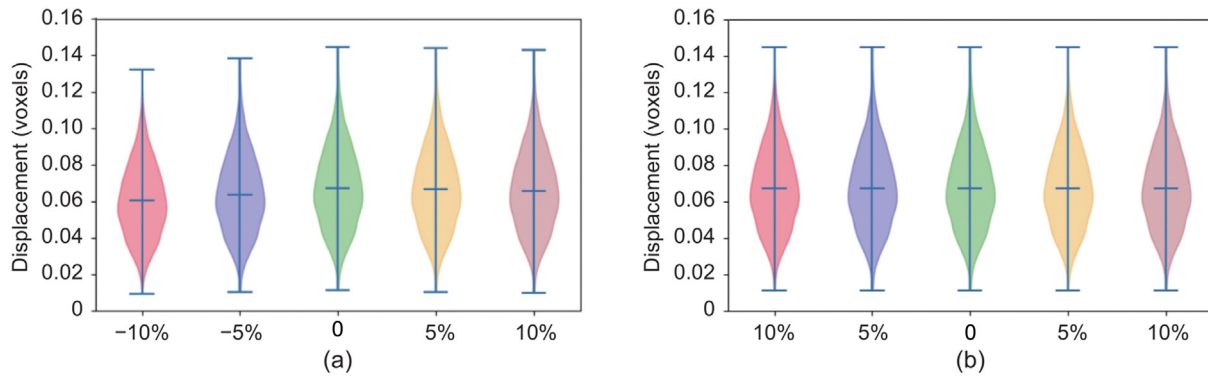


Fig. 9. Performance of N-M9 varying the values of: (a) Young's modulus variation based on 5×10^6 . (b) Poisson's ratio variation based on 0.35.

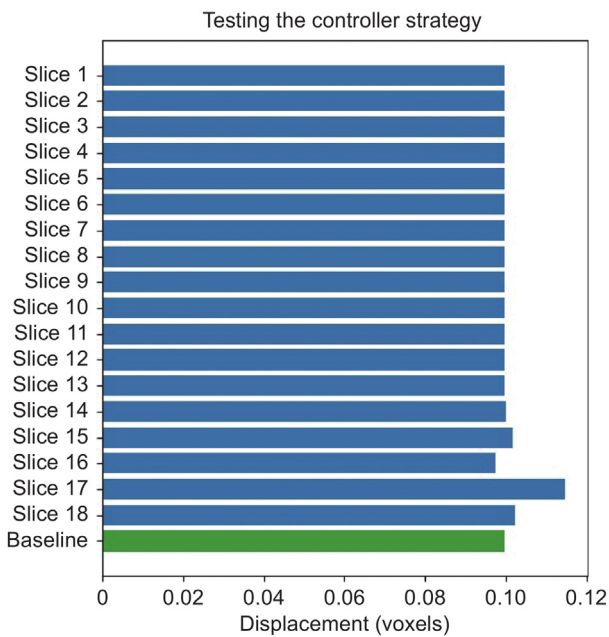


Fig. 10. Performance of N-M9 under 18 different scenarios where the values of the controller phase offset associated with the n slice are set to 0.

Results suggest that neuroevolution-based approaches are generally more capable of producing robust catheter actuator morphologies than AFPO due to the genetic operators (i.e., crossover and mutation) induced a broader exploration in the search space. In addition, the actuator produced by NEAT could outperform the one generated by HyperNEAT. Arguably, the design of the substrate utilised during experimentation affected the performance of HyperNEAT.

Furthermore, the number of voxels plays a crucial role in bending tasks; the fittest actuator found by AFPO exhibited significantly fewer contractile voxels; however, its performance was not superior when considering the multitude of possible controller strategies. On the other hand, the fittest actuator generated by NEAT and HyperNEAT showed at least twice the number of voxels of the actuator generated by AFPO. Their performance, nevertheless, overcomes the one from the actuator generated by AFPO. Arguably, the striped no-voxel pattern exhibited by NEAT and HyperNEAT actuators helped to enhance the bending movement. Although, the designs produced by the proposed NE-based methodologies were not validated *in vitro*, this should not hinder their applicability or lessen their better performance

demonstrated when compared with AFPO. The validation of AFPO applied on Voxelyze has been previously proved after fine-tuning the environment simulation parameters [3]. The selection of materials to build the actuator are considered as the material parameters inputs to Voxelyze, while the manufacturing process of artificial material should not hinder the generalization of this methodology.

The fact that the fitness function of the morphologies considered in the evolutionary optimisation methodology is provided by Voxelyze, a voxel-based physics simulator, may give rise to some concern around the translation of designs to real-world applications through concurrent manufacturing processes. Voxelyze utilises cube elements as building blocks and, thus, the morphologies produced may be predominately comprised by orthogonal boundaries; a characteristic that is not ideal for real-world applications. Namely, soft material (i.e., plastic melted through a 3D printer nozzle) or biohybrid material (i.e., living cells that are subject to tissue adherence/contraction) can not formulate perfectly right angles in the micro-scale. Nonetheless, the implementation of the evolutionary optimisation through indirect representations (i.e., CPPNs) can alleviate part of this limitation. These representations are scale free. Consequently, the anatomical resolution of the morphology may be increased without any further tests of the computational demanding evolutionary optimisation. The enhanced resolution may develop convex morphologies that would appear as orthogonal in lower resolutions [3]. The alleviation of right angles can be further enhanced through the use of appropriate activation functions on NEAT algorithm. On the other hand, this is not true for all cases. Therefore, an implementation of filtering the morphologies for manufacturing compatibility can follow the **morphology generator** proposed here, in the automated design pipeline envisioned. These filters will remove morphologies that are not possible or trivial to build with the available manufacturing techniques. However, the implementation of these filters are considered as an aspect of future work.

Considering the results and insights gathered from this research, future work may include the addition of more periodic activation functions, such as cosine and tangent. Following that tactic will help to induce more diverse morphological patterns produced by CPPNs. Another avenue of future work consists of a broader exploration of the number of hidden layers and neurons that may lead to better performance under HyperNEAT. In order to improve the design of the substrate, an approach called *ES-HyperNEAT*, focused on evolving the location of every neuron and the pattern of weights among them, can be implemented [79].

Moreover, a more robust performance comparison against other NE-based approaches will be conducted. For instance, Evolutionary eXploration of Augmenting Memory Models (EXAMM),

which is capable of evolving ANNs (including recurrent ANNs) through a wide variety of memory structures [80,81]. Taking into account that real-world optimisation problems can be time-constrained, the required computational time of further methodologies studied will be a key factor. Furthermore, the utilisation of surrogate-model assisted optimisation [82] will be considered, to minimise the need for computationally demanding physics simulators. Finally, another crucial aspect of future work will be focused on evolving controllers for the fittest catheters actuators found in this research.

CRedit authorship contribution statement

Hugo Alcaraz-Herrera: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation. **Michail-Antisthenis Tsompanas:** Writing – review & editing, Methodology, Funding acquisition, Conceptualization. **Igor Balaz:** Writing – review & editing, Project administration, Funding acquisition, Conceptualization. **Andrew Adamatzky:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This project has received funding from the European Union's Horizon Europe research and innovation programme (101070328). UWE researchers were funded by the UK Research and Innovation (10044516).

References

- [1] A. Schulz, C. Sung, A. Spielberg, W. Zhao, R. Cheng, E. Grinspun, D. Rus, W. Matusik, Interactive robogami: An end-to-end system for design of robots with ground locomotion, *Int. J. Robot. Res.* 36 (10) (2016) 1131–1147, <http://dx.doi.org/10.1177/0278364917723465>.
- [2] J. Hiller, H. Lipson, Dynamic simulation of soft multimaterial 3d-printed objects, *Soft Robotics* 1 (1) (2014) 88–101.
- [3] S. Kriegman, D. Blackiston, M. Levin, J. Bongard, A scalable pipeline for designing reconfigurable organisms, *Proc. Natl. Acad. Sci.* 117 (4) (2020) 1853–1859, <http://dx.doi.org/10.1073/pnas.1910837117>.
- [4] K.O. Stanley, R. Miikkulainen, Evolving neural networks through augmenting topologies, *Evol. Comput.* 10 (2) (2002) 99–127, <http://dx.doi.org/10.1162/106365602320169811>.
- [5] J.E. Auerbach, J.C. Bongard, Evolving complete robots with CPPN-NEAT: The utility of recurrent connections, in: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, Association for Computing Machinery, 2011, <http://dx.doi.org/10.1145/2001576.2001775>.
- [6] K.O. Stanley, D.B. D'Ambrosio, J. Gauci, A Hypercube-Based Encoding for Evolving Large-Scale Neural Networks, *Artif. Life* 15 (2) (2009) 185–212, <http://dx.doi.org/10.1162/artl.2009.15.2.15202>.
- [7] K.O. Stanley, Compositional pattern producing networks: A novel abstraction of development, *Genet. Program. Evol. Mach.* 8 (2) (2007) 131–162, <http://dx.doi.org/10.1007/s10710-007-9028-8>.
- [8] G.S. Hornby, ALPS: the age-layered population structure for reducing the problem of premature convergence, in: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, 2006, pp. 815–822, <http://dx.doi.org/10.1145/1143997.1144142>.
- [9] H. Alcaraz-Herrera, M.-A. Tsompanas, A. Adamatzky, I. Balaz, Neuroevolution algorithms applied in the designing process of biohybrid actuators, 2024, <https://arxiv.org/abs/2408.07671>.
- [10] F. Tanaka, C. Aranha, Co-evolving morphology and control of soft robots using a single genome, in: 2022 IEEE Symposium Series on Computational Intelligence (SSCI), 2022, pp. 1235–1242, <http://dx.doi.org/10.1109/SSCI51031.2022.10022230>.
- [11] D. Matthews, A. Spielberg, D. Rus, S. Kriegman, J. Bongard, Efficient automatic design of robots, *Proc. Natl. Acad. Sci.* 120 (41) (2023) <http://dx.doi.org/10.1073/pnas.2305180120>.
- [12] I. Asimov, *I, Robot*, Gnome Press, United States, 1950, p. 253.
- [13] S. Hirose, Y. Umetani, The development of soft gripper for the versatile robot hand, *Mech. Mach. Theory* 13 (3) (1978) 351–359.
- [14] N. Negroponte, *Soft architecture machines*, The MIT Press, 1976.
- [15] P. Akella, M. Cutkosky, Manipulating with soft fingers: modeling contacts and dynamics, in: Proceedings, 1989 International Conference on Robotics and Automation, IEEE, 1989, pp. 764–769.
- [16] A. Bicchi, Intrinsic contact sensing for soft fingers, in: Proceedings, IEEE International Conference on Robotics and Automation, IEEE, 1990, pp. 968–973.
- [17] F.Y. Chen, Gripping mechanisms for industrial robots: an overview, *Mech. Mach. Theory* 17 (5) (1982) 299–311.
- [18] C. Gillespie, C. Gillespie, Optical control of electrostatic actuators using amorphous silicon, *Strain* 1 (12) (1985) 300.
- [19] G.L. Kenaley, M.R. Cutkosky, Electrorheological fluid-based robotic fingers with tactile sensing, in: ICRA, 1989, pp. 132–136.
- [20] H. Lipson, Challenges and opportunities for design, simulation, and fabrication of soft robots, *Soft Robotics* 1 (1) (2014) 21–27.
- [21] D. Rus, M.T. Tolley, Design, fabrication and control of soft robots, *Nature* 521 (7553) (2015) 467–475.
- [22] E. Milana, Soft robotics for infrastructure protection, *Frontiers in Robotics and AI* 9 (2022) 1026891.
- [23] R.R. Murphy, Marsupial and shape-shifting robots for urban search and rescue, *IEEE Intell. Syst. their Appl.* 15 (2) (2000) 14–19.
- [24] X. Wang, Q. Zhang, D. Shen, J. Chen, A novel rescue robot: Hybrid soft and rigid structures for narrow space searching, in: 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE, 2019, pp. 2207–2213.
- [25] T. Wen, J. Hu, J. Zhang, X. Li, S. Kang, N. Zhang, Design, performance analysis, and experiments of a soft robot for rescue, *J. Mech. Robot.* 16 (7) (2024).
- [26] G. Methenitis, D. Hennes, D. Izzo, A. Visser, Novelty search for soft robotic space exploration, in: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, 2015, pp. 193–200.
- [27] S. Aracri, F. Giorgio-Serchi, G. Suaria, M.E. Sayed, M.P. Nemitz, S. Mahon, A.A. Stokes, Soft robots for ocean exploration and offshore operations: A perspective, *Soft Robotics* 8 (6) (2021) 625–639.
- [28] C.S.X. Ng, G.Z. Lum, Untethered soft robots for future planetary explorations? *Adv. Intell. Syst.* 5 (3) (2023) 2100106.
- [29] S. Mintchev, D. Zappetti, J. Willemin, D. Floreano, A soft robot for random exploration of terrestrial environments, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 7492–7497.
- [30] T. Patino, R. Mestre, S. Sanchez, Miniaturized soft bio-hybrid robotics: a step forward into healthcare applications, *Lab Chip* 16 (19) (2016) 3626–3630.
- [31] C. Majidi, Soft robotics: a perspective—current trends and prospects for the future, *Soft Robotics* 1 (1) (2014) 5–11.
- [32] J.-H. Hsiao, J.-Y. Chang, C.-M. Cheng, Soft medical robotics: clinical and biomedical applications, challenges, and future directions, *Adv. Robot.* 33 (21) (2019) 1099–1111.
- [33] Y.S. Kwok, J. Hou, E.A. Jonckheere, S. Hayati, A robot with improved absolute positioning accuracy for CT guided stereotactic brain surgery, *IEEE Trans. Biomed. Eng.* 35 (2) (1988) 153–160.
- [34] C.C. Nguyen, M.T. Thai, T.T. Hoang, J. Davies, P.T. Phan, K. Zhu, L. Wu, M.A. Brodie, D. Tsai, Q.P. Ha, et al., Development of a soft robotic catheter for vascular intervention surgery, *Sensors Actuators A* 357 (2023) 114380.
- [35] J. Rogatinsky, D. Recco, J. Feichtmeier, Y. Kang, N. Kneier, P. Hammer, E. O'Leary, D. Mah, D. Hoganson, N.V. Vasilyev, et al., A multifunctional soft robot for cardiac interventions, *Sci. Adv.* 9 (43) (2023) eadi5559.
- [36] C.C. Nguyen, T. Teh, M.T. Thai, P.T. Phan, T.T. Hoang, H. Low, J. Davies, E. Nicotra, N.H. Lovell, T.N. Do, Bidirectional soft robotic catheter for arrhythmia treatment, in: 2022 International Conference on Robotics and Automation (ICRA), IEEE, 2022, pp. 9579–9585.
- [37] V. Levering, Q. Wang, P. Shivapooja, X. Zhao, G.P. López, Soft robotic concepts in catheter design: an on-demand fouling-release urinary catheter, *Adv. Healthc. Mater.* 3 (10) (2014) 1588–1596.
- [38] P.I. Baburova, D.V. Kladko, A. Lokteva, A. Pozhitkova, V. Rummyantseva, V. Rummyantseva, I.V. Pankov, S. Taskaev, V.V. Vinogradov, Magnetic soft robot for minimally invasive urethral catheter biofilm eradication, *ACS Nano* 17 (21) (2023) 20925–20938.
- [39] Y. Yang, J. Wang, L. Wang, Q. Wu, L. Ling, Y. Yang, S. Ning, Y. Xie, Q. Cao, L. Li, et al., Magnetic soft robotic bladder for assisted urination, *Sci. Adv.* 8 (34) (2022) eabq1456.
- [40] T. Gopesh, J.H. Wen, D. Santiago-Dieppa, B. Yan, J.S. Pannell, A. Khalessi, A. Norbash, J. Friend, Soft robotic steerable microcatheter for the endovascular treatment of cerebral disorders, *Sci. Robotics* 6 (57) (2021) eabf0601.

- [41] C. Zhou, Y. Yang, J. Wang, Q. Wu, Z. Gu, Y. Zhou, X. Liu, Y. Yang, H. Tang, Q. Ling, et al., Ferromagnetic soft catheter robots for minimally invasive bioprinting, *Nature Commun.* 12 (1) (2021) 5072.
- [42] Y. Li, J. Peine, M. Mencattelli, J. Wang, J. Ha, P.E. Dupont, A soft robotic balloon endoscope for airway procedures, *Soft Robotics* 9 (5) (2022) 1014–1029.
- [43] X. Hu, Y. Zhou, M. Li, J. Wu, G. He, N. Jiao, Catheter-assisted bioinspired adhesive magnetic soft millirobot for drug delivery, *Small* 20 (11) (2024) 2306510.
- [44] B. Mazzolai, V. Mattoli, L. Beccai, E. Sinibaldi, Emerging technologies inspired by plants, *Bioinspired Approaches Human-centric Technol.* (2014) 111–132.
- [45] B. Mazzolai, P. Salvini, On robots and plants: The case of the plantoid, a robotic artifact inspired by plants, in: *Plant Ethics*, Routledge, 2018, pp. 221–230.
- [46] M. Friedman, K. Krauthausen, Interview with barbara mazzolai: Plants, plantoids, and active materials, *Active Mater.* (2021) 129.
- [47] B. Mazzolai, G. Meloni, A. Degl'Innocenti, Can a robot grow? Plants give us the answer, in: *Bioinspiration, Biomimetics, and Bioreplication 2017*, 10162, SPIE, 2017, pp. 24–33.
- [48] B. Mazzolai, C. Laschi, P. Dario, S. Mugnai, S. Mancuso, The plant as a biomechatronic system, *Plant Signal. Behav.* 5 (2) (2010) 90–93.
- [49] A. Sadeghi, A. Mondini, E. Del Dottore, V. Mattoli, L. Beccai, S. Taccola, C. Lucarotti, M. Totaro, B. Mazzolai, A plant-inspired robot with soft differential bending capabilities, *Bioinspiration Biomim.* 12 (1) (2016) 015001.
- [50] Y. Peng, H. Nabae, Y. Funabora, K. Suzumori, Peristaltic transporting device inspired by large intestine structure, *Sensors Actuators A* 365 (2024) 114840.
- [51] Y. Peng, H. Nabae, Y. Funabora, K. Suzumori, Controlling a peristaltic robot inspired by inchworms, *Biomim. Intell. Robotics* 4 (1) (2024) 100146.
- [52] J. Bhatia, H. Jackson, Y. Tian, J. Xu, W. Matusik, Evolution gym: A large-scale benchmark for evolving soft robots, *Adv. Neural Inf. Process. Syst.* 34 (2021) 2201–2214.
- [53] L. Smith, R. MacCurdy, SoRoForge: End-to-end soft actuator design, *IEEE Trans. Autom. Sci. Eng.* 20 (3) (2023) 1475–1486, <http://dx.doi.org/10.1109/TASE.2023.3241866>.
- [54] L. Smith, T. Hainsworth, J. Haines, R. MacCurdy, Automated synthesis of bending pneumatic soft actuators, in: *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, 2022, pp. 358–363, <http://dx.doi.org/10.1109/RoboSoft54090.2022.9762105>.
- [55] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.* 6 (2) (2002) 182–197, <http://dx.doi.org/10.1109/4235.996017>.
- [56] H. Zhang, M.Y. Wang, F. Chen, Y. Wang, A.S. Kumar, J.Y.H. Fuh, Design and development of a soft gripper with topology optimization, in: *2017 IEEE/RIS International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 6239–6244, <http://dx.doi.org/10.1109/IROS.2017.8206527>.
- [57] R. Wen, Z. Guo, T. Zhao, X. Ma, Q. Wang, Z. Wu, Neuroevolution of augmenting topologies based muscular-skeletal arm neurocontroller, in: *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2017, pp. 1–6, <http://dx.doi.org/10.1109/I2MTC.2017.7969727>.
- [58] N. Cheney, J. Bongard, H. Lipson, Evolving soft robots in tight spaces, in: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO '15*, 2015, pp. 935–942, <http://dx.doi.org/10.1145/2739480.2754662>.
- [59] J.E. Auerbach, J.C. Bongard, Evolving CPPNs to grow three-dimensional physical structures, in: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO '10*, 2010, pp. 627–634, <http://dx.doi.org/10.1145/1830483.1830597>.
- [60] S.K. Jha, F. Josheski, Artificial evolution using neuroevolution of augmenting topologies (NEAT) for kinetics study in diverse viscous mediums, *Neural Comput. Appl.* 29 (12) (2018) <http://dx.doi.org/10.1007/s00521-016-2664-2>.
- [61] T. Kimura, R. Niiyama, Y. Kuniyoshi, Modularized genotype combination to design multiobjective soft-bodied robots, in: *2021 IEEE 4th International Conference on Soft Robotics (RoboSoft)*, 2021, pp. 295–301, <http://dx.doi.org/10.1109/RoboSoft51838.2021.9479428>.
- [62] S. Risi, K.O. Stanley, Confronting the challenge of learning a flexible neural controller for a diversity of morphologies, in: *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, GECCO '13*, 2013, pp. 255–262, <http://dx.doi.org/10.1145/2463372.2463397>.
- [63] J. Clune, B.E. Beckmann, C. Ofria, R.T. Pennock, Evolving coordinated quadruped gaits with the HyperNEAT generative encoding, in: *2009 IEEE Congress on Evolutionary Computation*, 2009, pp. 2764–2771, <http://dx.doi.org/10.1109/CEC.2009.4983289>.
- [64] J. Yosinski, J. Clune, D. Hidalgo, S. Nguyen, J.C. Zagal, H. Lipson, Evolving robot gaits in hardware: the HyperNEAT generative encoding vs. parameter optimization, in: *Proceedings of the 2023 Artificial Life Conference*, 2011, p. 134, <http://dx.doi.org/10.7551/978-0-262-29714-1-ch134>.
- [65] D.B. D'Ambrosio, K.O. Stanley, Generative encoding for multiagent learning, in: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, 2008, pp. 819–826, <http://dx.doi.org/10.1145/1389095.1389256>.
- [66] J. Clune, C. Ofria, R.T. Pennock, The sensitivity of hyperneat to different geometric representations of a problem, in: *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, 2009, pp. 675–682, <http://dx.doi.org/10.1145/1569901.1569955>.
- [67] N. Cheney, R. MacCurdy, J. Clune, H. Lipson, Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding, *ACM SIGEVOlution* 7 (1) (2014) 11–23.
- [68] Reconfigurable Organisms, https://github.com/skriegman/reconfigurable_organisms, Accessed: 2024-05-19.
- [69] BioMeld Reconfigurable Organisms, https://github.com/Antisthenis/reconfigurable_organisms/tree/biomeld_dev2, Accessed: 2024-05-19.
- [70] M.-A. Tsompanas, I. Balaz, Outline of an evolutionary morphology generator towards the modular design of a biohybrid catheter, *Frontiers in Robotics and AI* 11 (2024).
- [71] M.D. Schmidt, H. Lipson, Age-fitness pareto optimization, in: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, 2010, pp. 543–544, <http://dx.doi.org/10.1145/1830483.1830584>.
- [72] K. Stanley, R. Miikkulainen, Efficient evolution of neural network topologies, in: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, 2, 2002, pp. 1757–1762 vol.2, <http://dx.doi.org/10.1109/CEC.2002.1004508>.
- [73] D.E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in: *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and their Application*, 1987, pp. 41–49.
- [74] M.-A. Tsompanas, Incremental growth on compositional pattern producing networks based optimization of biohybrid actuators, in: S. Smith, J.a. Correia, C. Cintrano (Eds.), *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, Springer Nature Switzerland, Cham, 2024, pp. 275–289.
- [75] G. Klambauer, T. Unterthiner, A. Mayr, S. Hochreiter, Self-normalizing neural networks, in: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 972–981.
- [76] E.C.R.G. (UCF), The hypercube-based NeuroEvolution of augmenting topologies (HyperNEAT) users page, 2016, <http://eplex.cs.ucf.edu/hyperNEATpage/>, Accessed: March 24th, 2024.
- [77] N. Jakobi, Evolutionary robotics and the radical envelope-of-noise hypothesis, *Adapt. Behav.* 6 (2) (1997) 325–368.
- [78] Z.D. Jastrzebski, *The Nature and Properties of Engineering Materials*, Wiley, California, United States, 1987, p. 636.
- [79] S. Risi, K.O. Stanley, An enhanced hypercube-based encoding for evolving the placement, density, and connectivity of neurons, *Artif. Life* 18 (4) (2012) 331–363, http://dx.doi.org/10.1162/ARTL_a_00071.
- [80] A. Ororbia, A. ElSaid, T. Desell, Investigating recurrent neural network memory structures using neuro-evolution, in: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '19*, 2019, pp. 446–455, <http://dx.doi.org/10.1145/3321707.3321795>.
- [81] A.S. Thakur, A.B. Awari, Z. Lyu, T. Desell, Efficient neuroevolution using island repopulation and simplex hyperparameter optimization, in: *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2023, pp. 1837–1842, <http://dx.doi.org/10.1109/SSCI52147.2023.10371872>.
- [82] R.J. Preen, L. Bull, A. Adamatzky, Towards an evolvable cancer treatment simulator, *Biosystems* 182 (2019) 1–7.