Towards Automatic Design of Soft Pneumatic Actuators: Inner Structure Design Using CNN Model and Bézier Curve-Based Genetic Algorithm

Loïc Mosser[1], Laurent Barbé[1], Lennart Rubbert[1] and Pierre Renaud[1]

***Abstract*— In this paper, the development of a method for the design of soft pneumatic actuators is described. The focus is given on the interest of using a deep learning model to explore the design space with a genetic algorithm. In particular, we propose to perform the automatic synthesis of the inner structure of pneumatic actuators using Bézier curves and Gaussian Mixture Points, to have a simple representation of the actuator genotype. This makes it possible to represent a wide variety of structures and to take into account the presence of the actuator pneumatic supply. It is shown a CNN model can interestingly be used in conjunction with FEM. FEM is being used to train initially the CNN model and for the control of accuracy, while the CNN model reduces the computational cost, offering a sufficient accuracy during the synthesis thanks to transfer learning. Through two case studies, the capacity of generating geometrically complex designs such as a double-helix network for a twisting actuator is outlined. Its possible extension and further use are also discussed.**

I. Introduction

Thanks to their compliance, soft robots are considered in a wide range of contexts such as medical [1] or mobile [2] robotics, and to solve manipulation and grasping issues [3]. Pneumatic actuation of soft robots is largely investigated [4] with the design of soft pneumatic actuators (SPAs). One important aspect in the design of SPAs is the definition of their inner structure, where pressure applies as considered in [5]–[7]. A large variety of shapes and topologies of SPA inner structures has been considered, which includes tree-like pneumatic networks [8], and networks with variable sections [9], [10]. In parallel, manufacturing techniques are improved, in particular with recent advances in additive manufacturing of soft material such as silicone [11], [12]. The design space, *i.e.* the number of possible inner designs of SPA is then very large. Design methods are needed to cope accordingly with the synthesis of SPAs.

The elaboration of design methods for SPAs has been intensively considered [13]. One way to differentiate works is to consider the prior information introduced by the designer. The optimization of a predefined shape is one approach [5], [14]. In [6], the configuration of a given set of artificial muscles is determined by optimization. Interestingly, other works consider no initial knowledge on SPA design. Topology optimization [7], [8], genetic algorithms (GA) [10], [15], are then considered as detailed in [10]. In the latter work, GA is exploited with a compact representation of the SPA using computational pattern producing networks [16]. Such encoding makes it easier to explore a large design space, as the SPA shape is described by a limited number of parameters, instead of a direct discrete representation of all voxels composing the structure. However, there is no direct physical meaning of the network. To design the SPA inner structure, where pressure is applied to get a motion, it seems then difficult to explicit essential conditions such as the continuity of the pneumatic network, and the presence of an air supply at a fixed location. To circumvent this, as a first contribution, we propose in this paper to develop a design method based on GA but introducing a specific encoding for the description of the SPA inner structure. The proposed approach is based on Bézier curves and Gaussian Mixture Points (GMP), to build the internal structures as tree-like networks or channels with variable sections.

Design with no prior information using a GA can be limited by the very large design space, and the corresponding computational cost of exploration using finite element modeling (FEM) to assess SPA performance. Recent work in the field of composite material design [17] shows that the use of a Convolutional Neural Network (CNN) model can be very beneficial to the exploration of large design spaces. A GA is then implemented, with strong reduction of computational cost thanks to the substitution of FEM by the CNN model during most of the design exploration. CNN models have been also recently demonstrated to be accurate for the simulation of soft structures including large deformation of hyper-elastic materials [18], [19], as we can encounter with SPA. As a consequence, our second contribution is to build a design method for SPA using a behavioral model based on a CNN to accelerate the process. We integrate in particular an accuracy control of the model and a transfer learning mechanism of the CNN to ensure its accuracy during the whole design space exploration.

Our design method is presented and assessed in the following. Promising results are presented in terms of computational cost and capability to identify SPA inner structure without any prior information before the synthesis. The section II presents the SPA genotype built specifically for SPAs. The GA and the overall architecture of the design method are presented in section III. We describe then the implementation and assessment of the design method in section IV including experimental results. Discussion on the method and conclusions are drawn in section V.

[1] The authors are with ICube Laboratory, University of Strasbourg, CNRS, INSA Strasbourg, France (email: mosserl@unistra.fr; laurent.barbe@unistra.fr; lennart.rubbert@insa-strasbourg.fr; pierre.renaud@insa-strasbourg.fr)

This work has been partially supported by ROBOTEX 2.0 (Grants ROBOTEX ANR-10-EQPX-44-01 and TIRREX ANR-21-ESRE-0015) funded by the French program Investissements d'avenir.

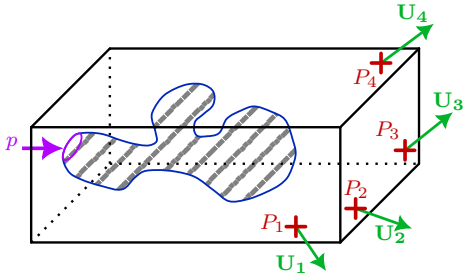

Fig. 1. Schematic representation of the design problem. A pressure p is applied to a pneumatic chamber (dashed). In this example, $n = 4$ points are being used to define the SPA output motion.

II. From the Design Problem to SPA Genotype

A. Definition of the design problem

The SPA is considered to be a deformable structure composed of a single material. The actuator performance is defined as a set of desired displacements at given locations on the SPA (fig. 1). This can be the displacement of a single point, or multiple points to describe for instance a desired surface shape or rotational motion. We designate $\mathbf{U}_i \in \mathbb{R}^3$, $i \in [1, n]$ as the set of desired displacements at n points on the external surface of the SPA (Fig. 1). They are to be obtained for a given pressure p, applied at a fixed location. The pressure level is usually predefined, as in [8], [16], [20]. The designer may also interactively select the pressure level, as in [21]. In the following, the set of displacements describes how relevant a solution is, so it can be used to define the phenotype of the SPA, following the terminology of GA [22].

The outer shape of the SPA is defined initially, as considered in other works using GA or topology optimization [17], [23], with for instance cylindrical [5], [24] or prismatic shape [6] to satisfy space requirements. The design method has to provide the inner structure of the SPA, *i.e.* the shape of the pneumatic chamber (fig. 1).As outlined in section I, we propose first to build a specific description of SPAs to elaborate their genotypes for the GA.

B. Genotype of a SPA

The SPA genotype is intended to describe the presence of an internal void structure which can be considered, without loss of generality, as a network of channels with variable section and topology. For obvious practical integration reasons, we need to ensure the presence of a pressure supply, at a predetermined fixed location.

Our proposition (fig. 2) is twofold. First, Bézier curves are used to describe the general shape and topology of the SPA inner structure (fig. 2-a). N_c curves denoted Γ_k, $k \in [1, N_c]$ are employed. The shape of the k-th curve is defined by N_k control points $\mathbf{C}_{i,k}$, $i \in [1, N_k]$. The coordinates of any point $\mathbf{M}_k(t)$, $t \in [0, 1]$ along the curve Γ_k are then expressed as [25]:

$$\mathbf{M}_k(t) = \sum_{i=1}^{N_k} \binom{N_k}{i-1} t^i (1-t)^{N_k-i+1} \mathbf{C}_{i,k} \qquad (1)$$

The definition of Bézier curves allows us to take into account explicitly the presence of the pressure supply, as

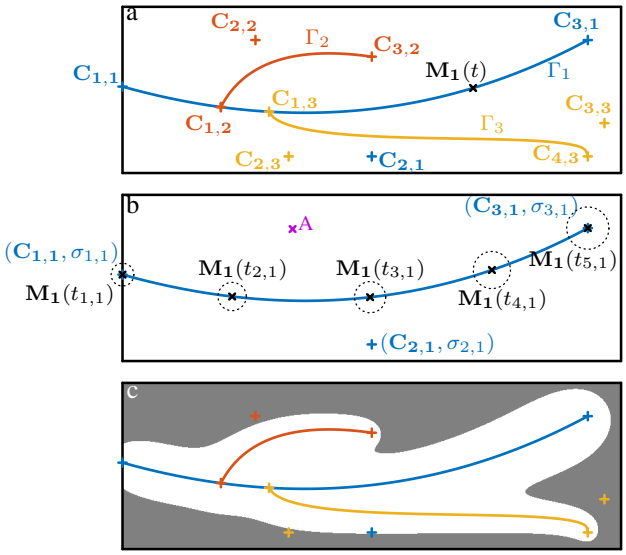

Fig. 2. Decomposition of the use of the proposed genotype to create one SPA individual (representation in 2D for sake of clarity). In (a), the inner network is created using Bézier curves. Here, $N_1 = 3$, $N_2 = 3$ and $N_3 = 4$. In (b), GMPs (blue) are equally distributed along the Bézier curve, their activation values being indicated by the size of blue circles (only curve #1 is represented for simplicity). In (c), the inner void structure is obtained after thresholding of the resulting distribution.

it is simply needed to express that $\mathbf{C}_{1,1}$, the first point of the first Bézier curve, is at the pneumatic supply location. Furthermore, the continuity of the network can be ensured by choosing $\mathbf{C}_{1,k}$, $\forall k > 1$, the initial point of a curve Γ_k, as a point of another existing curve.

The second aspect of our proposition is the use of Gaussian Mixture Points (GMPs), interestingly considered for soft robot design in [15]. A GMP is a point which is the center of a Gaussian probability density function, that describes the probability of material presence. The idea is to sum these probabilities at each point in space, and to define the presence or absence of material after a simple thresholding. It would be very difficult to obtain a continuity of the inner void structure using only GMPs scarcely distributed in the volume. So, we combine the use of GMP with Bézier curve representation. A so-called activation term $\sigma_{i,k} \in \mathbb{R}^{+*}$ is associated to each control point $\mathbf{C}_{i,k}$ (Fig. 2-b). An activation value $m_k(t)$, $t \in [0, 1]$ can then be computed along the Bézier curve Γ_k using a relationship similar to equation (1):

$$m_k(t) = \sum_{i=1}^{N_k} \binom{N_k}{i-1} t^i (1-t)^{N_k-i+1} \sigma_{i,k} \qquad (2)$$

The curve Γ_k is then discretized to have $N_{d,k}$ spatially equally distributed points. The implementation of the discretization process is developed in subsection II-C. It leads to a set of values $t_{j,k} \in \mathbb{R}$, $j \in [1, N_{d,k}]$ that represents the value of t for the j-th point $\mathbf{M}_k(t_{j,k})$ along the curve Γ_k after discretization (fig. 2-b). It is defined as a GMP with an activation term $m_k(t_{j,k})$ that defines its participation in the probability density map.

The probability of the presence of the inner void structure

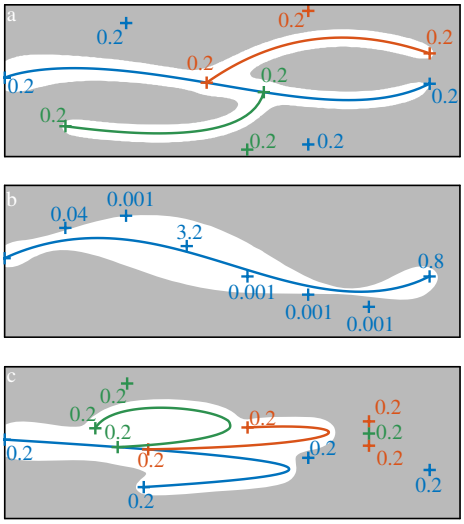

Fig. 3. 2D examples of possible inner structures defined using the proposed genotype. Color crosses represent the control points for the Bézier curve of the same color and associated parameters $\sigma_{i,k}$ are given in the figure next to the control points. In (a), an individual defined using $N_c = 3$ Bézier curves with $N_1 = 4$, $N_2 = 3$ and $N_1 = 3$. In (b) $N_c = 1$ Bézier curve with $N_1 = 8$. In (c) an individual with three curves with $N_1 = 4$, $N_2 = 4$, $N_3 = 3$.

at any point $\mathbf{A} \in \mathbb{R}^3$ in the volume (fig. 2-b) is defined by

$$p(\mathbf{A}) = \sum_{k=1}^{N_c} \sum_{j=1}^{N_{d,k}} \exp\left(-\frac{\|\mathbf{AM}_k(t_{j,k})\|^2}{2 \cdot m_k(t_{j,k})}\right) \quad (3)$$

After thresholding, the geometry of the inner structure is obtained (fig. 2-c). As a summary, the proposed genotype of SPA is defined by $(\mathbf{C}_{i,k}, \sigma_{i,k}), i \in [1, N_k]$, $k \in [1, N_c]$, used to compute the network of Bézier curves and the location of GMPs.

C. Implementation and examples of genotypes

Computing equally distributed points along a Bézier curve is commonly achieved using a numerical approximation of the curve length [26]. We use this approach, and we approximate the curve with segments between points of the Bézier curve. Then, we use this approximation to extract the set of t values $t_{j,k}$ that defines points on the Bézier curve Γ_k with a fixed distance L_d between them, *i.e.* with equal spatial resolution. L_d is common to all the designs and is a parameter of our method.

In fig. 3, the interest of the genotype is illustrated with 3 types of situations that are of interest for SPA design. In (a), a tree-like structure with a quasi-constant section can be generated by using one main curve and 2 other ones being connected to the latter. In (b), one curve with a variable section is generated, thanks to the selection of activation values. This can be seen as the creation of inner chambers connected through channels. In (c), Bézier curves tend to create loops, so we consequently generate an inner chamber of large size inside the volume. One may remark that using Bézier curves with only two control points, straight lines are obtained. So it is also possible to describe rectilinear structures by creating a network of straight lines, if for instance the designer wants to restrict the description to prismatic shapes with sharp corners.

Algorithm 1: Pseudocode of the proposed design method

Input: target phenotype **targPhen**;
Output: A set of individuals with a simulated phenotype close to **targPhen**;
Algorithm:
Initialization
do
  `/* Inner loop */`
  while *no individual is old enough* **do**
    Selection;
    Crossover and mutation;
    Evaluation;
  end
  `/* Generation control */`
  FEM Simulation $\rightarrow$ **FEMPhen**;
  if $\exists$ *1 individual* $\mid J(\textbf{\textit{targPhen}}, \textbf{\textit{FEMPhen}}) < \varepsilon_1$ **then**
    End: A solution has been found;
  else
    if $J(\textbf{\textit{FEMPhen}}, \textbf{\textit{CNNPhen}}) > \varepsilon_2$ **then**
      Transfer Learning;
    end
    Reset the age of individuals;
  end
while *no solution has been found*;

III. Design Method

A. Structure of the algorithm

The pseudocode of the proposed method is presented in algorithm 1. It is based on a GA, which is the *inner loop* in the pseudocode. Following the GA terminology [22], [27], the method has to provide a list of individuals, characterized by their genotypes, that exhibit a phenotype of interest. The genotype was introduced in section II. The targeted phenotype, *i.e.* the desired set of observable characteristics of the SPA, is denoted ***targPhen*** and it is defined from the displacements $\mathbf{U_i}$, $i \in [1, n]$. Our design space is very large because no prior information is included. To compensate for the computational time of FEM, it is proposed to estimate phenotypes using a CNN model. It is, however, expected that a CNN model obtained after an initial learning phase cannot remain accurate during the whole design process, as it is trained on a small initial population, unable to represent all the possible SPA. Therefore, a control of accuracy and updates are integrated in the algorithm. The latter is then the coupling between 2 processes: one dedicated to the identification of individuals of interest, based on GA, and another one to control and update the CNN model so it remains relevant for phenotype assessment of individuals. Both processes are first explained, before detailing how crossover and mutation are managed given the proposed genotype.

B. GA-based individual selection

To describe the GA-based selection, let us first describe the way to select individuals, based on a behavior that can

either be simulated with FEM or determined from a CNN learnt from simulated data.

The process corresponds to a rather standard implementation of a GA. The selection of individuals starts with the initialization step, which consists in the creation of an initial population of individuals whose genotypes are randomly generated. Then, the inner loop (see algorithm 1) starts. Each execution of the inner loop corresponds to the production of one generation of individuals. 3 operations are achieved at each execution of the inner loop.

First, in the **selection** step, pairs of individuals are selected based on their performance. Individual performance is expressed by a fitness value, which is the value of the cost function J (algorithm 1) when comparing the individual phenotype to the target phenotype. The choice of J is problem-dependent, and it will be described in the following for case studies.

Then the **crossover and mutation** step aims at generating new individuals, each one designated as a "child". Children are the result of the reproduction of two parent individuals, according to crossover rules, and it has a certain probability of mutation according to mutation rules. The crossover and mutation rules are defined in the next section. After this step, the total population is composed of both the parent population and children population.

Finally, the **evaluation** step is performed: the phenotype of each child is assessed, while the phenotype of parents is already known. A selection of individuals with the best fitness values is extracted. Their age, which represents the survival along the different generations, is incremented. The other individuals are removed.

Once an old enough individual has been identified, the inner loop of the algorithm is ended. Then, we enter in the generation control. In this phase, the closeness of the phenotype simulated with FEM ***FEMPhen*** and the desired phenotype ***targPhen*** is verified for each individual. The design process is successful if at least one individual phenotype is close enough such that $J(\boldsymbol{targPhen}, \boldsymbol{FEMPhen})$ is under a certain tolerance ε_1.

C. CNN elaboration and update

The CNN aims to predict the phenotypes of individuals. As the CNN is a behavioral model, it needs first to be trained. This is done in the initialization phase: the phenotype of all individuals of the initial population is simulated using FEM. From this data, the CNN is built and used during the whole phase corresponding to the inner loop.

The control of model accuracy is achieved every time we enter in the generation control phase. FEM simulation of the individuals selected at the end of the inner loop is first performed. Only a small set of individuals, typically 15% of the population, has then its phenotype simulated.

The simulated phenotypes are used to determine whether a solution has been found within the best individuals in the population. So we make sure the algorithm stops only if FEM-based SPA phenotype is relevant. Each time we enter in the generation control phase, but no solution is found yet, the simulated (from FEM) and estimated (from CNN) phenotypes, denoted respectively ***FEMPhen*** and ***CNNPhen***, are compared. If the difference between the two phenotypes is significant, *i.e.* above a given tolerance ε_2, for all tested individuals, a CNN update is needed. This is achieved using transfer learning which consists of training on a reduced number of individuals and a reduced number of learning iterations. It is conducted on individuals from the initial population and on the simulated individuals. The details about the CNN, its initial training and the transfer learning step are given in section IV-b. The population is then sent back in the inner loop and the age of all individuals is reset to zero.

D. Crossover and mutation

Crossover between two designs is achieved using a one-point crossover operation [28]. This crossover operator is selected as it is defined for GMPs in [15] and presented to have ability to preserve physical regions with useful functionality.

In our situation, as a first step, the design space is divided in 2 regions using a plane. The plane is defined a point, which is the barycenter of all the control points of both parent designs, and a randomly chosen orientation. A child design is then obtained by using the control points of one parent located in one region and from the second parent in the other region. One may notice that the use of the barycenter of all control points to define the plane helps to limit the chance of using only all the control points of only one parent. If the two parents are not defined by the same number of Bézier curves, the child design may keep the largest number of curves with a probability of 0.5 for each extra curve.

Mutation is defined with a 2-step decision. First, a child individual has a probability of 0.5 of undergoing a mutation. Second, if mutation is considered to occur, there is a probability of 0.5 for each control point to be relocated.

IV. Assessment of the method

A. Case studies

Two case studies are considered to assess the relevance of the method. The two SPA to be designed share the same external volume and boundary conditions (fig. 4). The overall shape is prismatic, with a size chosen equal to 10 mm $\times$ 10 mm $\times$ 30 mm. The pressure is applied on the left side, at its center. The pressure level is set to $p = 50$ kPa. The remaining part where the pressure inlet is located has no displacement. For ease of prototyping, we use Agilus 30 material obtained with Polyjet printing process. The material was characterized using ISO 37 standard with a Zwick/Roell Z005 testing machine, with specimens printed in the 3 possible directions of space. By identification on a 10% strain range, Young's modulus are equal to 0.76 MPa, 0.71 MPa, 0.38 MPa, which shows the anisotropy introduced by the process. In the following, an average value of 0.62 MPa, close to a silicone, is being used as the direction of printing is not known during the synthesis. The material is considered to be quasi-incompressible with $\nu = 0.46$.

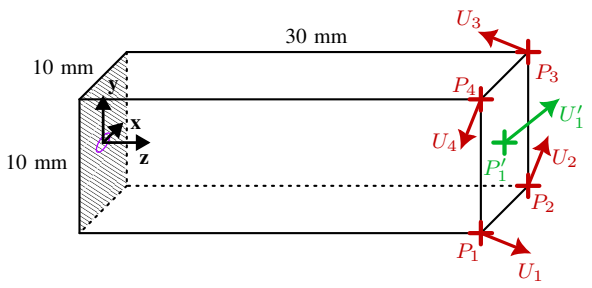

Fig. 4. Conditions for the two case studies, with one target displacement (green) or 4 target displacements (red) with highlighted pneumatic supply (purple) and fixed surface (dashed)

The case study #1 corresponds to a simple definition of SPA behavior: the goal is to respect a displacement at a single point P'_1 ($n = 1$) at the center of the right surface (fig. 4, green). The case study #2 is defined by the search for a set of displacement at four different points ($n = 4$). The phenotype is then composed of the displacements $(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \mathbf{U}_4)$ of vertices (P_1, P_2, P_3, P_4) (fig. 4, red). It will be used to assess the ability of the method to design SPA with specific motion, e.g. a twisting motion of the actuator.

In algorithm 1, the cost function $J(\boldsymbol{phen}_1, \boldsymbol{phen}_2)$ defines the way two phenotypes $\boldsymbol{phen}_1$ and $\boldsymbol{phen}_2$ are compared. In the following it is defined as

$$J(\boldsymbol{phen}_1, \boldsymbol{phen}_2) = \|\boldsymbol{phen}_1 - \boldsymbol{phen}_2\|_2$$

For case study #1, the individual phenotype is defined as $\mathbf{U}'_1$. The target phenotype is set to (0, 1.6 mm, 0.5 mm). For case study #2, our goal is to get a twisting motion along the Z axis, around point P'_1. We then focus on the displacement in the XY plane, and do not consider the Z component of the displacements of points. Accordingly, the individual phenotype is $(\mathbf{U_1}.\mathbf{x}, \mathbf{U_1}.\mathbf{y}, ...\mathbf{U_4}.\mathbf{x}, \mathbf{U_4}.\mathbf{y})$. The target phenotype is $0.22\times$(1, -1, 1, 1, -1, 1, -1, -1) mm.

B. Implementation of the algorithm

The CNN is based on a ResNet architecture. It is constructed using a trial and error approach as used in [17] and guided with observations of [29], [30]. It takes as an input the 3D matrix describing the presence of material within the SPA as obtained after spatial discretization of the volume. In the following, the voxel size is $0.4 \times 0.4 \times 0.4$ mm^3. This means that the volume is manipulated as a $(25, 25, 75)$ matrix which represents a set of 46875 voxels. The CNN output is a vector containing the phenotype, which is the 3D displacement of the n points under consideration (fig. 1). Thus, the output is a $3n$-values vector. The CNN is constituted of a convolution layer as input, 16 residual convolutional with full pre-activation, as it is reported to favor generalization performance [30], an average pooling layer as it is known to reduce the dimension before dense layers [29], two dense layers of 128 neurones and a dense output layer.

The initial training of the CNN uses a population of 100000 randomly generated individuals (60000 for training, 40000 as a test set to assess the CNN accuracy). The training is achieved on 20 epochs with a learning rate of 10^{-4}. After the initial training, the phenotype estimation using a CNN has a root mean square (RMS) error of $9 \cdot 10^{-3}$ mm and a coefficient of determination $R^2 = 0.996$ for case study #1, and a RMS error of $8 \cdot 10^{-3}$ mm and a $R^2 = 0.998$ for case study #2. These values can be considered as satisfactory, and they will be used as a reference to comment on the contribution of transfer learning as introduced in the method.

The implementation of the algorithm is achieved using Python on a PC (Intel i9-10900KF, 64 GB of RAM, NVIDIA RTX-3090 graphic card). The CNN is implemented using Tensorflow. FEM simulations are performed considering large displacements and large deformations, using COMSOL in batch mode through Matlab and Python. Computation times are reported after using parallelization on 4 cores for the simulation of the initial population used for learning.

The **selection** process selects the 200 best individuals as parents and randomly generate 500 couples from them. Thus, the **crossover and mutation** step creates 500 new individuals that are then **evaluated**. While the size of those populations increases the computing time of selection and crossover/mutation steps, it is expected to have a positive impact on the number of generations needed to find a solution. The values have been selected arbitrarily based on the successful search for solutions for case studies.

When the method engages the generation control stage (algorithm 1), 30 individuals are simulated using FEM. When necessary, the transfer learning step is applied with a reduced learning rate of 10^{-5} on 10 epochs and with 1500 individuals of the initial population and 50 times the population of 30 simulated individuals. In this way, the individuals of the initial and new populations have the same weight in the training population, so it favors diversification of the updated training population.

The limit age associated to the inner loop is set to 5 generations. This parameter has an impact on the number of simulations done along the generations. Thus it must be high enough to limit the use of FEM simulations and small enough to identify correctly the generations where the transfer learning step must be applied. The threshold values ϵ_1 and ϵ_2 are chosen equal to 0.1 mm. The parameter L_d is set equal to the size of a voxel to have the discretization of Bézier curves at every voxel.

C. Results

The assessment of the method is focused first on the interest of the transfer learning in the generation control phase, before focusing on computational cost and finally the possibility to generate specific behavior of SPA.

1) Transfer learning contribution: The fig. 5 represents for the case study #1 the phenotype values of the initial population (in black), the target phenotype (in red) and, as a red dashed line, the tolerance zone associated to the parameter ϵ_1. One can see the target phenotype is not present in the initial domain of phenotypes and the convergence towards the target phenotype. In fig. 6, the evolution of the fitness value during the synthesis is plotted. Blue box

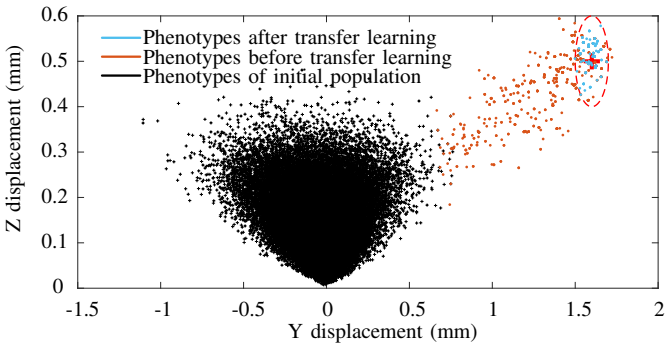

Fig. 5. Representation of the estimated phenotypes in the YZ plane for case study #1. Initial population in black, target phenotype in red, admissible region delimited by a red dashed line, individuals obtained after transfer learning in blue.

plots correspond to fitness values obtained using CNN with transfer learning, as implemented in the algorithm. Green box plots are generated using FEM. To assess the impact of transfer learning, the algorithm is also run using the CNN without transfer learning. Red box plots represent the values obtained in this situation. The same representation is given for case study #2 (fig. 6 bottom).

The contribution to the design process of the CNN and the associated generation control phase appear in both case studies. For case study #1, transfer learning occurs after 20 generations (orange vertical line). With transfer, there is convergence and a solution is found: the final fitness value obtained from CNN with transfer learning and from FEM during verification is below the dotted black line, defined by ϵ_1. Without transfer, the target fitness value is not obtained. For case study #2, the target phenotype is in the same way not part of the initial domain, as illustrated in the attached video. There are 2 activations of transfer learning.

For case study #1 (resp. #2), at the last generation, the mean relative error between $J(\boldsymbol{targPhen}, \boldsymbol{FEMPhen})$ and $J(\boldsymbol{targPhen}, \boldsymbol{CNNPhen})$ is equal to 46% (resp. 21%) (fig. 6, gaps blue/green), while it is equal to 97% (resp. 152%) without transfer learning (fig. 6, gaps red/green). This confirms quantitatively the interest of this learning phase as the design space is too large to have a single and accurate CNN model.

2) Computation time: The use of a CNN requires FEM simulation during an initial generation of population, an initial CNN model training, and transfer learning to maintain its accuracy. Corresponding computation times are reported in table I for the two case studies.

To assess the interest of using a CNN, we estimate the computation time that would be needed using the inner loop in the algorithm 1 with only FEM simulations. The number of individuals to be simulated is chosen equal to the number of individuals examined using the CNN. The mean time for a single FEM simulation is 63 seconds. Thus, an implementation of the genetic algorithm would have required 220.9 (553.9 for #2) hours for case study #1, *i.e.* 1.8 (4.3) times the duration of the proposed design algorithm. The use of CNN appears clearly of interest even though it requires

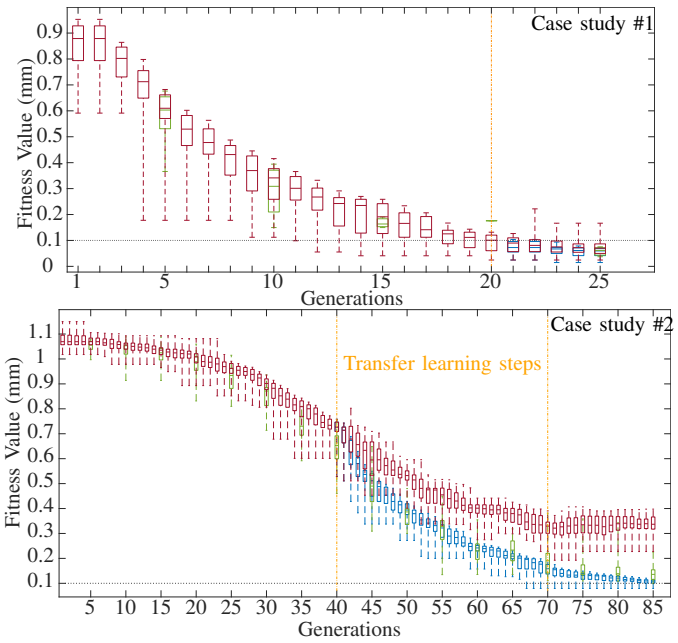

Fig. 6. Representation of the fitness value along the generations based on their simulated phenotype (green), their estimated phenotype with the CNN updated using transfer learning (blue) and their estimated phenotype with the CNN without update (red) (case #1 upper figure, and case #2 lower figure). Transfer learning steps are highlighted in orange.

Steps of the algorithm	Computation time for each step (h)		Proportion in the overall computation time for each step	
	case #1	case #2	case #1	case #2
Initial population simulation	96	96	79.3 %	74.8 %
CNN initial training	20	20	16.5 %	15.6 %
CNN Evaluation	0.05	0.23	4e-4 %	0.1 %
Selection Crossover/Mutation	0.86	2.69	0.7 %	2.1 %
FEM Simulation	2.64	6.61	2.2 %	5.1 %
Transfer learning	0.43	0.90	0.3 %	0.7 %
Other operations	1.01	1.94	0.8 %	1.5 %
Total Time	120.99	128.37	100 %	100 %

TABLE I

DISTRIBUTION OF THE COMPUTATIONAL TIME FOR THE DIFFERENT STEPS OF THE METHOD FOR CASE STUDY #1 AND #2

an initial learning phase.

3) Description of solutions: The obtained design for case study #1 is represented in fig. 7-a and b. One can note the cavity is created by combination of 6 Bézier curves with respectively $n_1 = 5$, $n_2 = 6$, $n_3 = 2$, $n_4 = 5$, $n_5 = 4$ and $n_6 = 3$. The genotype is in the end defined by 100 scalar parameters. The cavity has a symmetrical shape with respect to the plane of motion which is consistent with the search for zero displacement in the **x**-direction. It is interesting to note that the targeted displacement induces for the final design a maximal strain of 80 %, measured as the highest eigenvalue of the Green-Lagrange tensor. The CNN model is able in these conditions to describe the output displacement with an accuracy of 1%.

For case study #2, the design obtained using the design algorithm is represented in fig. 7-c and d. Interestingly, the phenotype chosen to get a twisting motion of the free surface

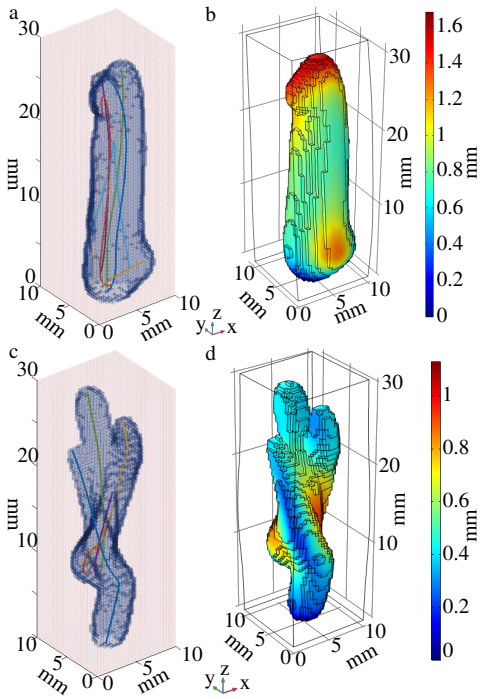

Fig. 7. Case study #1 and #2 solution representations with (a for #1 and c for #2) its definition with Bézier curves in the volume and (b for #1 and d for #2) the displacement field using FEM.

leads to a helical overall shape of the inner cavity. Such type of geometry is obtained using 5 Bézier curves with respectively $n_1 = 6$, $n_2 = 4$, $n_3 = 7$, $n_4 = 5$ and $n_5 = 5$. The geometry seems intuitively logical, and it is obtained without any prior information in the design process, nor any designer expertise. As outlined in the video associated to the paper, it largely outperforms even the best solutions in the initial population.

D. Experimental evaluation for case study #2

For further method assessment, the design obtained for case study #2 is implemented and tested. The SPA is produced using additive manufacturing, with Polyjet process (Stratasys, USA). A set of five specimens is printed. The SPA body is made out of Agilus30™ Black (FLX985) soft material. The fixed base is made of VeroMagentaV™ (RGD852) rigid material (Fig. 8).

For each specimen, a 100-Hz motion tracking system composed of 4 Arqus A5 cameras (Qualisys, Sweden) is used with markers attached at the location of points (P_1,P_2,P_3,P_4). Four other markers on the base plane are used to reconstruct the XY plane in which the displacements ($\mathbf{U_1}$,$\mathbf{U_2}$,$\mathbf{U_3}$,$\mathbf{U_4}$) are defined. They are plotted in figure 9, while pressure is applied from 0 to 50 kPa within 30 s. The standard deviation of the results for five pressurization cycles for the 5 specimens is equal to 0.03 mm, 0.027 mm, 0.017 mm and 0.019 mm for respectively P_1 to P_4, which shows a good consistency between the prototypes.

The amplitude of displacements of the 4 points of interest are similar (Fig. 9). This amplitude is in average 68% larger than the one expected from FEM simulation. The displacements measured at 25kPa, i.e. 50% of the target pressure,

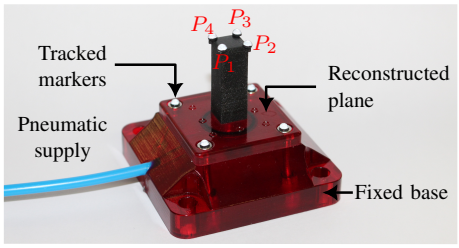

Fig. 8. Part of the experimental setup with the specimen on a fixed base, infrared markers tracked by the Qualisys system and a SPA. Points P_1 to P_4 are highlighted with respect to their corresponding marker placed on the SPA.

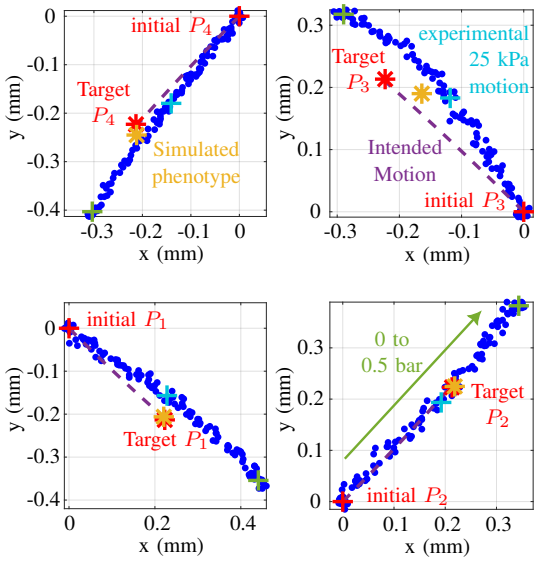

Fig. 9. Displacements of points P_1 to P_4. Blue markers are experimental data, light blue markers show the displacement at 25kPa. Final displacement at 50kPa corresponds to the green crosses.

correspond in average to 48% of the total displacement. So the material nonlinearity does not seem to be significant. The difference between simulation and experiment is probably here mainly due to the difficulty to calibrate the material constitutive model with this type of Polyjet-based material, which anisotropy was outlined [31]. Still, one can notice that in terms of direction of displacements, the path followed by the 4 points is very close to the motion associated to the desired twisting motion around the Z axis. This is very encouraging in terms of synthesis of SPA with the method.

V. Discussion and Conclusion

In this paper, we proposed a design method of SPA inner structures based on a GA incorporating a CNN model. The GA manipulates a representation of the SPA based on Bézier curves and GMPs. No prior information is used to determine the internal structure of an actuator. The introduction of a CNN model reduces significantly the computation time. At the same time, the CNN model accuracy remains sufficient during the synthesis thanks to the transfer learning, as noticed with the 2 case studies, including a non-trivial twisting SPA design. We could observe the possibility to design with no prior information a specific inner structure, even in presence of up to 80 % deformation with an accuracy of CNN model

of 1 %. For the second case study, the capacity to generate a twisting motion by working on the SPA inner structure was experimentally confirmed. This tends to confirm the interest of the approach we propose.

As such, the method is defined by several parameters, related to the GA and to the CNN. Their values were here mostly reused from a related work [17]. This is encouraging on the possibility to use the method as such, without major adjustment of these parameters. If such an adjustment is needed for other design problems, the user can tune the GA parameters following [27]. Concerning the CNN parameters and its architecture, well-known architecture such as described in [29] or automatic CNN model tuning as presented in [32] can be considered.

This work opens several perspectives. First, the synthesis problem is defined in a general way, as the search for desired displacements of a set of points. When the number of points increases, we may face difficulties related to the uneven distribution of permissible displacement errors, as described in [6]. Investigating this aspect will be of interest.

Also, the genotype we propose can be used to describe material removal on the outer part of the volume as well. Using GMPs placed at different locations, the creation of cellular materials with internal voids can also be described. A perspective is to build crossover and mutations rules for these operations, and to manage their simultaneous uses, as this could lead to the creation of non-functional individuals if the integrity of the pressurized inner cavity is not maintained.

Finally, the method could be extended to include as well multimaterial distributions, and to consider the integration in the definition of the fitness value of criteria beyond the motion behavior, such as manufacturing rules.

REFERENCES

[1] M. Cianchetti, C. Laschi, A. Menciassi, and P. Dario, "Biomedical applications of soft robotics," *Nature Reviews Materials*, vol. 3, no. 6, pp. 143–153, 2018.

[2] M. Calisti, G. Picardi, and C. Laschi, "Fundamentals of soft robot locomotion," *Journal of The Royal Society Interface*, vol. 14, no. 130, p. 20170101, 2017.

[3] J. Shintake, V. Cacucciolo, D. Floreano, and H. Shea, "Soft robotic grippers," *Advanced Materials*, vol. 30, no. 29, p. 1707035, 2018.

[4] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, May 2015.

[5] Y. Elsayed, A. Vincensi, C. Lekakou, T. Geng, C. M. Saaj, T. Ranzani, M. Cianchetti, and A. Menciassi, "Finite element analysis and design optimization of a pneumatically actuating silicone module for robotic surgery applications," *Soft Robotics*, vol. 1, no. 4, pp. 255–262, 2014.

[6] G. Maloisel, E. Knoop, C. Schumacher, and M. Bächer, "Automated routing of muscle fibers for soft robots," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 996–1008, 2021.

[7] E. M. de Souza and E. C. N. Silva, "Topology optimization applied to the design of actuators driven by pressure loads," *Structural and Multidisciplinary Optimization*, vol. 61, no. 5, pp. 1763–1786, May 2020.

[8] Y. Chen, Z. Xia, and Q. Zhao, "Optimal design of soft pneumatic bending actuators subjected to design-dependent pressure loads," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2873–2884, 2019.

[9] F. Chen, Y. Miao, G. Gu, and X. Zhu, "Soft twisting pneumatic actuators enabled by freeform surface design," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5253–5260, 2021.

[10] L. Smith, T. Hainsworth, J. Haimes, and R. MacCurdy, "Automated synthesis of bending pneumatic soft actuators," in *2022 IEEE 5th International Conference on Soft Robotics*, 2022, pp. 358–363.

[11] S. Walker, O. Yirmibeşoğlu, U. Daalkhaijav, and Y. Mengüç, "Additive manufacturing of soft robots," in *Robotic Systems and Autonomous Platforms*. Woodhead Publishing, 2019, pp. 335–359.

[12] O. D. Yirmibesoglu, J. Morrow, S. Walker, W. Gosrich, R. Cañizares, H. Kim, U. Daalkhaijav, C. Fleming, C. Branyan, and Y. Menguc, "Direct 3d printing of silicone elastomer soft robots and their performance comparison with molded counterparts," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, 2018, pp. 295–302.

[13] J. Pinskier and D. Howard, "From bioinspiration to computer generation: Developments in autonomous soft robot design," *Advanced Intelligent Systems*, vol. 4, no. 1, p. 2100086, 2022.

[14] R. Su, Y. Tian, M. Du, and C. C. L. Wang, "Optimizing out-of-plane stiffness for soft grippers," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 430–10 437, oct 2022.

[15] J. Hiller and H. Lipson, "Automatic design and manufacture of soft robots," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 457–466, 2012.

[16] L. Smith, T. Hainsworth, Z. Jordan, X. Bell, and R. MacCurdy, "A seamless workflow for design and fabrication of multimaterial pneumatic soft actuators," in *IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 2021, pp. 718–723.

[17] Y. Kim, Y. Kim, C. Yang, K. Park, G. X. Gu, and S. Ryu, "Deep learning framework for material design space exploration using active transfer learning and data augmentation," *npj Computational Materials*, vol. 7, no. 1, p. 140, Sep 2021.

[18] A. Odot, R. Haferssas, and S. Cotin, "Deepphysics: A physics aware deep learning framework for real-time simulation," *International Journal for Numerical Methods in Engineering*, vol. 123, no. 10, pp. 2381–2398, 2022.

[19] S. Deshpande, J. Lengiewicz, and S. P. Bordas, "Probabilistic deep learning for real-time large deformation simulations," *Computer Methods in Applied Mechanics and Engineering*, vol. 398, p. 115307, Aug 2022.

[20] H. Zhang, M. Y. Wang, F. Chen, Y. Wang, A. S. Kumar, and J. Y. Fuh, "Design and development of a soft gripper with topology optimization," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, pp. 6239–6244, 2017.

[21] Y. Yao, L. He, and P. Maiolino, "A Simulation-Based Toolbox to Expedite the Digital Design of Bellow Soft Pneumatic Actuators," in *5th International Conference on Soft Robotics*. IEEE, 2022, pp. 29–34.

[22] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[23] B. Caasenbrood, A. Pogromsky, and H. Nijmeijer, "A computational design framework for pressure-driven soft robots through nonlinear topology optimization," in *IEEE International Conference on Soft Robotics (RoboSoft)*, 2020, pp. 633–638.

[24] G. Runge, M. Wiese, L. Günther, and A. Raatz, "A framework for the kinematic modeling of soft material robots combining finite element analysis and piecewise constant curvature kinematics," in *3rd International Conference on Control, Automation and Robotics (ICCAR)*, 2017, pp. 7–14.

[25] H. Prautzsch, W. Boehm, and M. Paluszny, *Bézier and B-spline techniques*. Springer, 2002, vol. 6.

[26] J. Gravesen, *IV.7 - The Length of Bézier Curves*. Boston: Academic Press, Jan 1995, pp. 199–205.

[27] E. K. Burke and G. Kendall, *Search methodologies: introductory tutorials in optimization and decision support techniques*. Springer, 2014.

[28] G. Pavai and T. V. Geetha, "A survey on crossover operators," *ACM Comput. Surv.*, vol. 49, no. 4, dec 2016.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[30] ——, "Identity mappings in deep residual networks," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Springer International Publishing, 2016, pp. 630–645.

[31] M. W. Barclift and C. B. Williams, "Examining variability in the mechanical properties of parts manufactured via polyjet direct 3d printing," in *2012 International Solid Freeform Fabrication Symposium*. University of Texas at Austin, 2012.

[32] N. M. Aszemi and P. Dominic, "Hyperparameter optimization in convolutional neural network using genetic algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 6, 2019.