Design Optimization of Soft Pneumatic Actuators Using Genetic Algorithms*

Gundula Runge[1], Jan Peters[1], Annika Raatz[1]

***Abstract*— Recent trends in bioinspired robotic systems are paving the way for robots to become part of our daily lives. Soft robots, which are widely recognized as the next generation of human-friendly robots, are such a trend. Soft robots are generally more adaptable, more flexible, and safer than their rigid-link counterparts. Research in soft robotics has produced a broad variety of interesting solutions for all sorts of applications ranging from medical engineering and rehabilitation over exploration to industrial handling. This diversity together with a general lack of experience in designing with soft materials has contributed to a design flow that is highly empirical in nature. For soft robots to become mass-producible in the near future, more general design and modeling methods are needed. In this article, we present a method for the design optimization of soft robot modules that effectively combines finite element modeling and gradient-free optimization. To demonstrate the feasibility of the approach, a soft pneumatic actuator is designed and optimized. Performance analysis of the optimization scheme shows the robustness of the solution in the given case.**

I. INTRODUCTION

Present-day robots are still overwhelmingly caged in factory workplaces where interactions between humans and robots seldom take place. They excel at high-precision tasks in well-structured environments but perform poorly in highly dynamic environments where unforeseeable events may occur any time. Not only do they lack the adaptability to deal with uncertain situations, they also entail the risk of seriously harming those who come too close. Soft robotics is an exciting new research field which may alter our image of robotics altogether. Soft robots are mainly composed of materials whose Young's modulus is of the order of that of soft biological tissues [1]. They can easily conform their bodies to contoured surfaces and are sufficiently compliant so as to not cause grave damage.

While the body of literature on soft robotics is rapidly growing, there is no such thing as an archetypal soft robot and chances are there never will be. Unlike their rigid-bodied counterparts, these human-friendly robots may never populate large territories in industry or elsewhere. Rather, it is likely that they will occupy niches that have previously been devoid of autonomous agents. Nonetheless, soft robots bear the potential of revolutionizing our relationships with intelligent machines. Not only are soft robots likely to overcome the persisting barrier between humans and robots, they may also become an constitutive part of our lives. Research in soft robotics has recently produced a multitude of interesting solutions, which span applications from medical devices [2]–[4] over rehabilitation [5]–[7] to industrial handling [8], [9].

This diversity of possible applications and designs is certainly desirable. Yet the same diversity makes the systematic design of soft robots exceptionally cumbersome. A taxonomy of soft robots is not readily available and general design guidelines are not yet in place. Besides, engineers very often base their decisions on intuition that they have acquired from prior experience. When it comes to soft materials, however, engineering experience tends to be scarce. This lack of experience together with the absence of standardized methods has contributed to a design flow which is highly empirical in nature [10]. With a few exceptions [11]–[13], the functionality of a particular soft robot design is almost exclusively tested on a hardware prototype before a model is derived.

In this article, we introduce a model based optimization approach combining genetic algorithms and finite element analysis (FEA). The approach in itself is generic and can be applied to a variety of soft robot modules. Here, we demonstrate the feasibility on a soft pneumatic actuator (SPA). Owing to their versatility, SPAs have become the "pioneering genre of soft robotics" [14] in recent years. A broad spectrum of different SPA designs has been proposed in literature offering an extensive design space that can be explored by optimization techniques.

The article is structured as follows: In Sec. II, we provide an overview of the soft robotics design and modeling framework, where we focus on the optimization of soft robot modules with genetic algorithms. The proposed approach has been applied to a soft pneumatic actuator. The results are discussed in Sec. III. To demonstrate the utility of the method, the performance of the algorithm and the influence of different optimization settings are studied in Sec. IV. Finally, we summarize the results and give an outlook on future works.

II. SOFT ROBOTICS DESIGN AND MODELING FRAMEWORK

To counter the perceived lack of comprehensive development tools in soft robotics, we have previously proposed a holistic design and modeling framework that can be applied to a variety of soft robot modules. Following the approach outlined in [15], we have implemented a software tool that allows the user to parametrize a given soft robot module and automatically generate a model representing its forward kinematics [16]. The software tool effectively combines the computational capabilities of both MATLAB (MathWorks)

*This work was not supported by any organization

[1]Gundula Runge, and Annika Raatz are with the Institute of Assembly Technology, Leibniz Universität Hannover, 30823 Garbsen, Germany

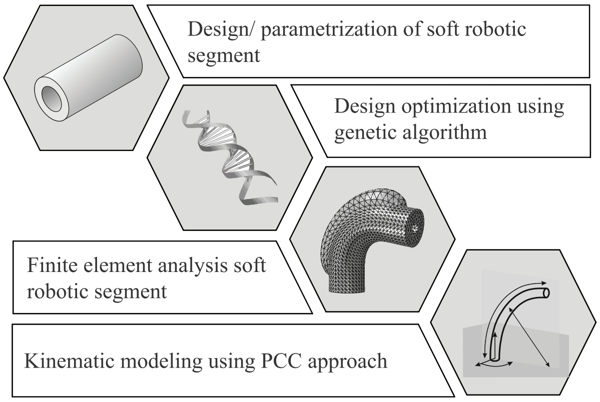

Fig. 1. Soft robotics design framework

and ABAQUS (Simulia, Dassault Systemes, RI) via a program interface.

The software framework, which is depicted in Fig. 1, consists of four building blocks. First, a three-dimensional model of a soft robot module is created in ABAQUS/CAE, whose parameters are subsequently optimized by a genetic algorithm using MATLAB. In each step of the optimization process, a finite element model of the soft module is created and the nonlinear differential equations are solved in ABAQUS/ANALYSIS. The results of the nonlinear finite element simulation are imported in MATLAB, where the configuration of the robot is evaluated using the piecewise constant curvature kinematics approach. Step 3 has already been discussed in [15] and [16] and is therefore only briefly summarized in the following. A detailed discussion of the kinematic modeling of a soft robot module in Step 4 is beyond the scope of this article; the interested reader is referred to [15] and [16].

A. *Design of Soft Robot Modules*

A large variety of soft robot designs have been proposed in literature, each with its unique capabilities regarding motion profiles and range of motion, stiffness, force output etc. Since a discussion of general design guidelines for soft robots would be beyond the scope of this article, we are focusing on the topology optimization of existing designs. Assuming that an initial design of a soft robot module is already in place, its morphology is optimized such that the final design closely matches the desired characteristics. This is achieved by fine-tuning the morphological parameters, that is, its shape, the geometrical arrangement of its elements, and its mechanical properties [17].

The first step of the proposed design and modeling framework starts with the design of an initial soft robot module, where we assume that a prototypical design already exists. First, a three-dimensional model of the module is created in ABAQUS/CAE. If a prototypical model for a module is already in place, its geometry can be easily parametrized by the user in MATLAB. Geometric parameters that can be adjusted in MATLAB include, for example, the length and cross section of a soft robot module, as well as the position of the actuators. Once these parameters have been properly defined, a 3D model can be automatically created in ABAQUS/CAE.

B. *Optimization of Soft Robot Modules*

In a second step, the initial design is optimized using a genetic algorithm. This is achieved by iteratively computing a series of finite element models and successively updating the design parameters. Note that the order of the second and third block in Fig. 1 may theoretically be reversed, since the execution of the finite element model is nested inside the optimization loop and initiated by the genetic algorithm.

In general, genetic algorithms are a popular choice for problems in stochastic optimization, in particular in the context of global optimization, where the best solution is to be found among multiple local minima [19]. Genetic algorithms simultaneously operate on a *population* of *individuals*, each possessing a set of parameter values, so-called *chromosomes*, where each set represents one solution to the problem at hand [18]. The effectiveness of a solution is measured by the *fitness* of the individual, a quantitative measure that is determined by a *fitness function* [18]. In every iteration of the optimization process, the fitness of each individual is computed and a new *generation* of individuals with different chromosomes is created. Over time, the chromosomes of all individuals are moved to a point where the fitness function has an optimum [19]. Different from gradient-based optimization algorithms, GAs randomly search the parameter space without taking into account the slope of the objective function. GAs employ a number of reproduction mechanisms inspired by natural evolution in order to refine the individuals from generation to generation [20].

Fig. 2 depicts a schematic of the optimization process and the evolutionary mechanisms used by the algorithm. The genetic algorithms starts by generating an initial population composed of randomly chosen individuals with different chromosomes (i.e. designs with different geometric parameters). It is after this step that the process enters a loop, where the following steps are performed in each iteration. First, a finite element model is created for each individual in ABAQUS/CAE and the nonlinear differential equations represented by the model are subsequently solved in ABAQUS/ANALYSIS. Once all the simulations are complete, the results are automatically imported in MATLAB and evaluated by the GA. Based on the prescribed fitness function, a fitness value is assigned to each individual. In the following, the next generation of individuals is created through a series of reproduction mechanisms:

1) Selection: Selection is a process in which a number of individuals are singled out from the current population one-by-one and inserted into an *intermediate population* [20]. When the maximum number of individuals in the pool has been reached, the process is stopped and variation is performed: i.e., reproduction, crossover, and mutation. The probability of an individual being selected for reproduction depends on it fitness value. Basically, the fitness value intro-

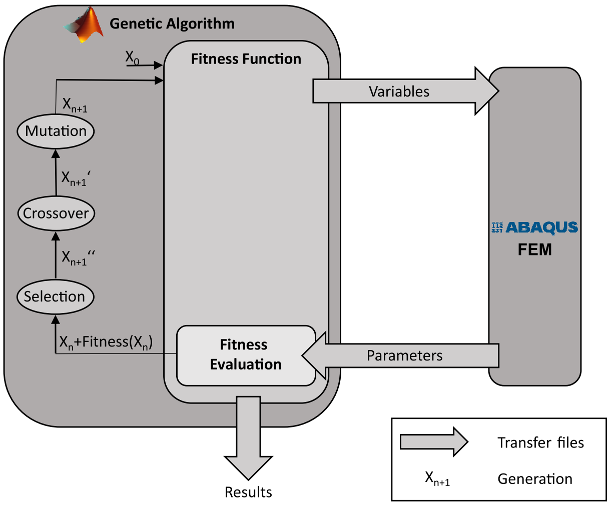

Fig. 2. Schematic of the optimization process

duces a bias that helps to differentiate between individuals in that individuals with higher fitness will achieve a higher probability of being selected for mating [21]. A variety of selection techniques have been proposed in literature, where the *roulette wheel* mechanism and the *universal sampling* mechanism are among the most common ones [21].

Reproduction is done by simply copying an individual from the intermediate population into the next one [20]. Variation of the chromosomes is then achieved through a two-stage process: crossover and mutation.

2) Crossover: Crossover exchanges some genes of two individuals to produce offspring whose chromosome is a combination of their parents' genes. Again, different crossover mechanisms have been proposed, the most common of which is the so-called *one-point crossover* [20]. This type of crossover is also the simplest one [21]. In one-point cross-over, the GA randomly picks a crossover point beyond which the gene strings are exchanged. Thereby, two new chromosomes are created [22]. *Multi-point crossover* is similar in that multiple string sections are exchange at a number of crossover points [21]. The idea behind multi-point crossover is that those parts of the chromosome that have the largest impact on the performance of an individual need not necessarily be adjacent [23].

3) Mutation: Mutation randomly chooses a subset of genes and alters their values [24]. This is realized either through perturbation of a gene's values or random selection of new values within the allowed parameter space [21]. As such, mutation serves as an innovation operator by introducing new genetic material into the population [20]. Mutation is the final reproduction mechanism in the cycle depicted in Fig. 2. Subsequently, the previous generation of individuals is replaced and the entire process starts anew.

The above process repeats until one of the termination criteria is met. This is usually the case when either the fitness value plateaus, the maximum number of generations is reached, or the allocated time has elapsed [18].

C. Finite Element Analysis of Soft Robot Segments

In a third step, finite element analysis is employed to determine the deformation of a soft robot design in response to the prescribed inputs. Given the highly nonlinear nature of the elastic materials used in soft robots, the mechanical relationships are usually difficult to derive analytically. As noted above, a finite element model is automatically created in ABAQUS/CAE when the model parameters have been properly defined in MATLAB (e.g. geometric parameters, material properties, loads, and boundary conditions). After the geometry of the soft robot module has been created and the loads and boundary conditions have been properly defined, the model is meshed into finite elements. In the following, the resulting nonlinear differential equations are solved using ABAQUS/ANALYSIS. Numerical integration of the nonlinear differential equations yields the nodal displacements of all the elements. Subsequently, the nodal displacements are automatically imported in MATLAB and evaluated such that the global deformation of the module is represented by a single curve coinciding with the approximate geometric centerline.

III. Case Study: Optimization of a Soft Pneumatic Actuator

In [16], we investigated the influence of different design parameters on the kinematics of a pneumatic actuator by systematically varying the length of the actuator and the chamber position. The results demonstrated that the position of the chamber has considerable influence on the curvature. Small changes in chamber position can lead to large deviations in the curvature. Since the design parameters also influence one another, they should be varied simultaneously. However, due to the large number of possible design parameters, systematic variation of the parameters is not always practical. In cases like these, optimization algorithms can be used to produce a set of optimized design parameters. In the following, the proposed model-based design optimization approach is demonstrated on a soft pneumatic actuator. Tab. I shows the design parameters and their limits. Fig. 3 depicts the exterior (left) and sectional view (right) of the actuator. The first prototype using the parameters identified in [16] is presented in Fig. III.

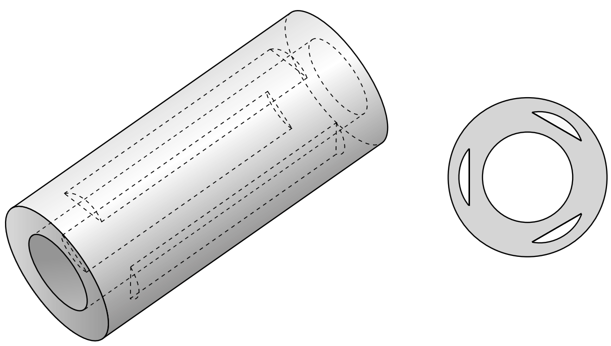

Fig. 3. Exterior and sectional view of the soft pneumatic actuator

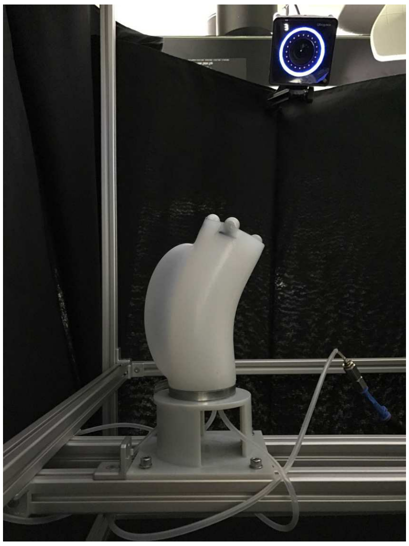

Fig. 4. Soft pneumatic actuator

TABLE I
LOWER AND UPPER BOUNDS OF THE DESIGN PARAMETERS

Parameter	Lower bounds in mm	Upper bounds in mm
Inner radius	10	18
Outer radius	20	50
Chamber position	15	40
Chamber thickness	2	8

The goal of the optimization was to minimize the radius of curvature, while keeping the ballooning as small as possible. Here, ballooning refers to the bulging of the inner chamber wall when the chamber is under pressure. Depending on the type of application, other target variables may be used as well. The main purpose of the present study was to demonstrate the efficacy of the design optimization rather than producing a specific design. The target values for the optimization, namely *radius of curvature* and *ballooning*, were combined into a single fitness for each individual. The corresponding fitness function is given by Eq. 1. Variable B_i denotes the ballooning of a design and R_i the radius of curvature when the actuator is under pressure. The factors a and b represent weighting factors, by which the influence of a variable on the optimization process can be adjusted. Here, the target *radius of curvature* with $b = 0.8$ was assigned a higher weight than the target *ballooning* with $a = 0.2$.

$$F_i = \frac{a \cdot \frac{B_i}{sum(B)} + b \cdot \frac{R_i}{sum(R)}}{sum(a \cdot \frac{B_i}{sum(B)} + b \cdot \frac{R_i}{sum(R)})} \tag{1}$$

Additionally, constraints were defined to rule out geometrically impossible actuator designs. The thickness of the ring-shaped actuator wall was constrained to be greater/equal to 10 mm, while the outer and inner chamber wall thickness was constrained to 2 mm and 4 mm minimum, respectively.

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} \leq \begin{bmatrix} -10 \\ -2 \\ -4 \end{bmatrix} \tag{2}$$

The settings for the genetic algorithm were as follows: For all experiments a population size of 100 was used, with a randomly created initial population and a maximum number of generations of 20. Unless stated otherwise, *stochastic uniform selection* was employed together with an additional elitism. The *adaptive-feasible mutation* property further allowed for optimization with linear constraints. Here, starting from the individual to be mutated, the algorithms randomly selects a direction and permissible range that does not violate the limits and shifts the parameters of the individual accordingly. For crossover, *intermediate crossover* was used, as this function can also handle bounds and constraints. With intermediate crossover, individuals are generated by averaging the genes of two parent individuals. For the sake of simplicity, the design parameters were defined as floating point numbers.

Fig. 5 shows the evolution of the four design parameters over the first ten generations. The values for the outer radius and the position of the chamber plateau after eight generations. The chamber thickness does not seem to influence the quality of the design significantly, since the rate of convergence is much lower.

Fig. 6 shows the convergence of the genetic algorithm with respect to the fitness value. For this purpose, the mean of the fitness values is plotted over the generations. Since the fitness function in Eq. 1 yields a value that accounts for an individual's fitness relative to the population within one generation only, we defined a globally comparable fitness value that accounts for the absolute fitness across all generations. The absolute fitness of an individual is calculated according to:

$$F_{i,abs} = \sqrt{(B_i + R_i)^2} \tag{3}$$

The mean of the fitness values of all the individuals in one generation is given by:

$$\overline{F}_{abs} = \sum_{i=1}^{n} \frac{F_{i,abs}}{n}, \tag{4}$$

where n is the number of individuals in a generation.

In Fig. 6, the standard deviation shows the divergence of the fitness values within one generation. The genetic algorithm converges very rapidly during the first ten generations, with the standard deviation decreasing from 800 to less than 20. Thereafter, the rate of convergence drops.

Fig. 7 depicts the best design of the first and last generation, respectively. The associated design parameters and target values are shown in Tab. II. Individual 2 is considered a much better solution given the higher weight

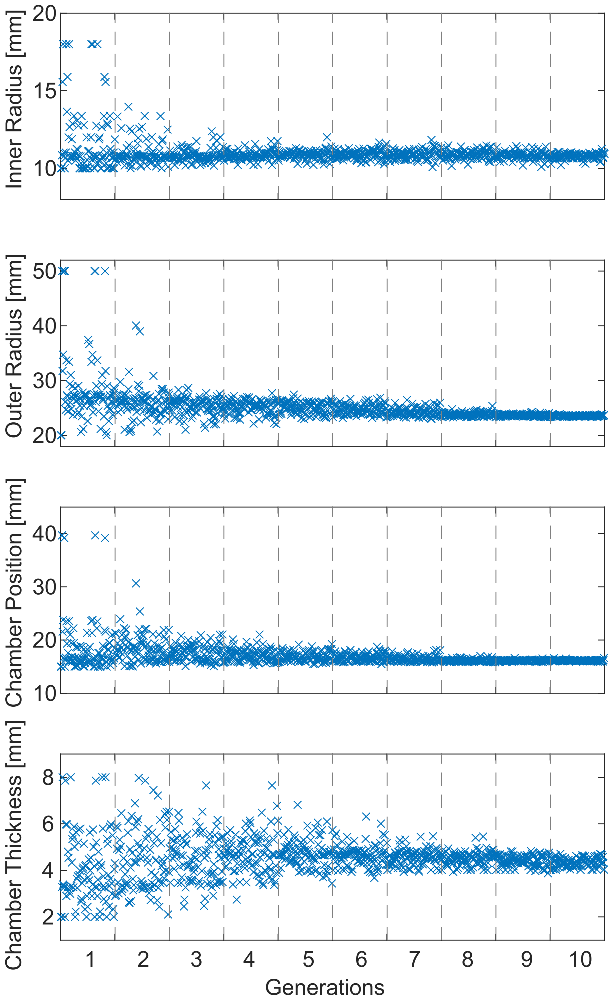

Fig. 5. Parameter development

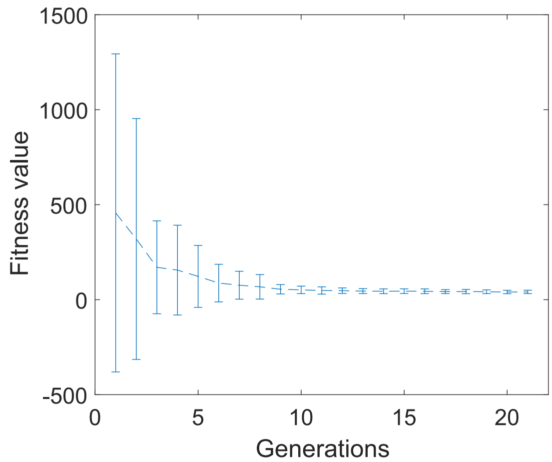

Fig. 6. Mean fitness for one trial

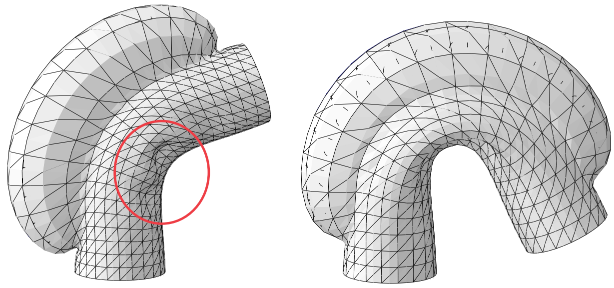

Fig. 7. Comparison of the best individual from the first and last generation

TABLE II

COMPARISON OF THE TWO ACTUATORS

	Individual 1	Individual 2
Inner radius [mm]	11.15	10.42
Outer radius [mm]	27.14	23.62
Chamber position [mm]	21.71	15.71
Chamber thickness [mm]	3.43	5.04
Optimum Ballooning [mm]	10.50	15.51
Optimum Radius [mm]	77.65	32.15

of the target value *radius*. Furthermore, individual 1 appears to have a rather unstable design, since the chamber opposite the pressurized chamber buckles (marked in red). In future works, it should be tested whether these results can be confirmed by real world prototypes.

Fig. 8 shows a comparison of the kinematic relations of the two actuators from Tab. II. The curvature of the actuation is plotted over the pressure. Individual 1 achieves a larger curvature with less pressure, since the chamber is located further outward and can therefore expand more easily. Individual 2 bends only at higher pressure but the maximum curvature is much larger. Compared to individual 3, i.e. the design that was studied in [16], the achievable bending angle of the final actuator is significantly higher.

IV. EVALUATION OF THE GENETIC ALGORITHM

A. Robustness

In order to investigate the influence of randomization on the results of the genetic algorithms, the same experiment was carried out 30 times. The algorithms were terminated after 11 generations. Each generation comprised 100 individuals. Fig. 9 shows the grand mean of the fitness values for each generation. The grand mean of the fitness values is given by:

$$\overline{\overline{F}}_{abs} = \sum_{j=1}^{k} \frac{\overline{F}_j}{k}, \tag{5}$$

where $\overline{F}_j$ is the group mean of the j-th trial and k the total number of trials.

The standard deviation in Fig. 9 denotes the deviation of the group mean with respect to the grand mean over all 30 trials. Fig. 9 indicates that the final results are very close to one another after only eleven generations, although the initial

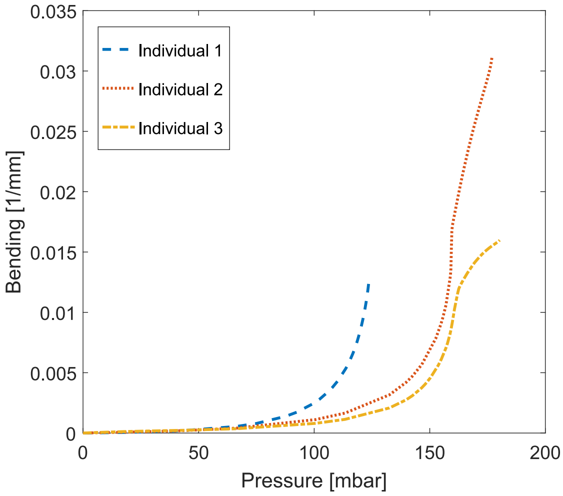

Fig. 8. Bending as a function of pressure Best soft pneumatic actuator

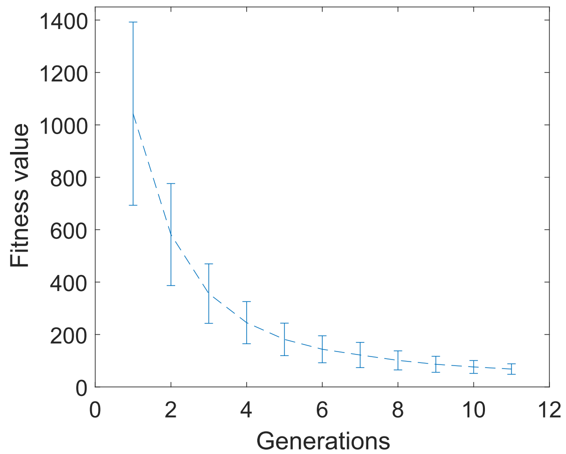

Fig. 9. Grand mean of the fitness values for 30 trials

populations tend to deviate from each other. Fig. 9 nicely illustrates the robustness of the algorithm, as the solution evolves in the same direction for all trials.

Tab. III summarizes the time and resources allocated for each experiment with a population size of 100 and a total of 11 generations. The target values of the best and worst trial as well as the average over all trials are also provided. As with any stochastic optimization, there are differences in the quality of the solution if the optimization is repeated. However, the small deviation of the final results shows that the algorithm is a reliable and relatively fast method for the design optimization of soft pneumatic actuators.

TABLE IV
COMPARISON OF DIFFERENT POPULATION SIZES

	Trial 1	Trial 2
Population size	100	16
Number of generations	11	70
Total number of individuals	1100	1120
Computation time	38 h	200 h
RAM	16 Gb	16 Gb
Cores	16	16
Optimum Ballooning [mm]	17.036	9.219
Optimum Radius [mm]	33.351	90.645

TABLE III
RESULTS OF THE 30 TRIALS

	Best trial	∅	Worst trial
Population size	100	100	100
Generations	11	11	11
Computing time	38 h	38 h	38 h
RAM	16 Gb	16 Gb	16 Gb
Cores	16	16	16
Optimum Ballooning [mm]	13.148	15.964	17.990
Optimum Radius [mm]	29.135	33.405	42.210

B. Population Size

In general, the coverage of the search space is an important criterion for the quality of an optimization process. If little or no information about the nature of the search and solution space is given, the entire search space should be covered evenly in case that the global optimum is located in a difficult-to-find area.

In Fig. 10 a two-dimensional subspace of the parameter space is shown. The parameters for the inner and outer radius, which were tested by the genetic algorithm, are depicted. The red lines represent the boundaries of the search space. In addition, the parameters of the inner and outer radius from the initial population and optimum are marked by red triangles and a yellow circle, respectively. The plot shows that not all regions within the search space are evenly covered.

In order to investigate the influence of the population size on the coverage of the search space, a genetic algorithm with a population size of 16 and the same settings as in the example above was run. Fig. 11 shows the corresponding parameters. To guarantee that the conditions are comparable to those in the above experiment, the number of generations was chosen such that number of individuals nearly matched those of the previous experiment. Tab. IV shows the settings and resources used in this experiment. The results show that the radius in Trial 2 is much worse than the one in Trial 1, although both experiments comprised nearly the same number of individuals. The radius in Trial 1, the one with the larger population size, is roughly three times as small as the radius in Trial 1. Even though the ballooning is nearly twice as large, the overall solution in the trial with the larger population size is better.

The above experiment indicates that the search space tends

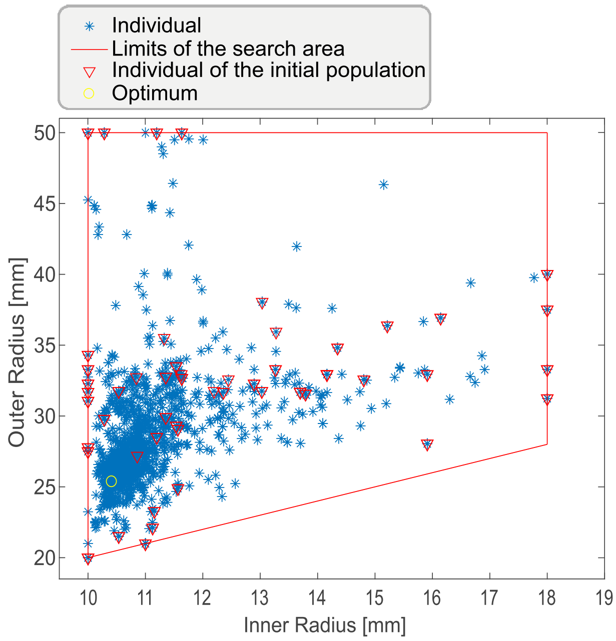

Fig. 10. Two-dimensional subspace for 100 individuals per generation

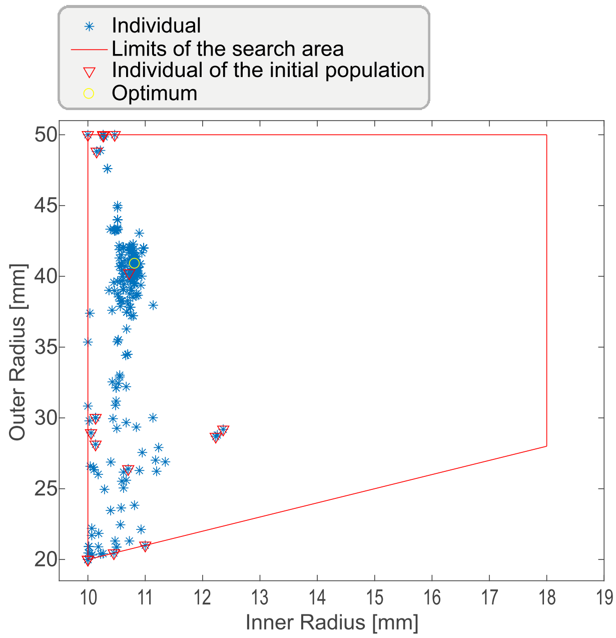

Fig. 11. Two-dimensional subspace for 16 individuals per generation

to be less evenly covered when the smaller population size is smaller. Since new individuals can only be generated by recombination, the initial population mainly determines the parameter range that can be explored by the genetic algorithm. Regions outside the range covered by the individuals of the initial population can only be reached through mutation. However, it seems that faraway regions are only seldom reached by mutation. The experiments indicate that a large initial population ensures a much better coverage of the search space for the problem at hand. Therefore, it is better to choose a smaller number of generations with a larger

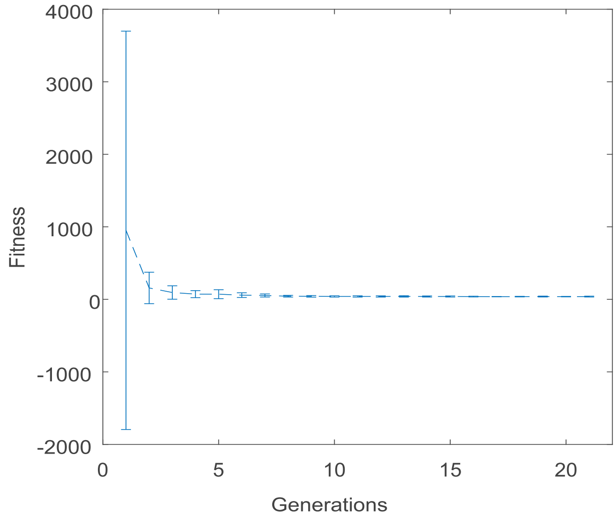

Fig. 12. Tournament selection

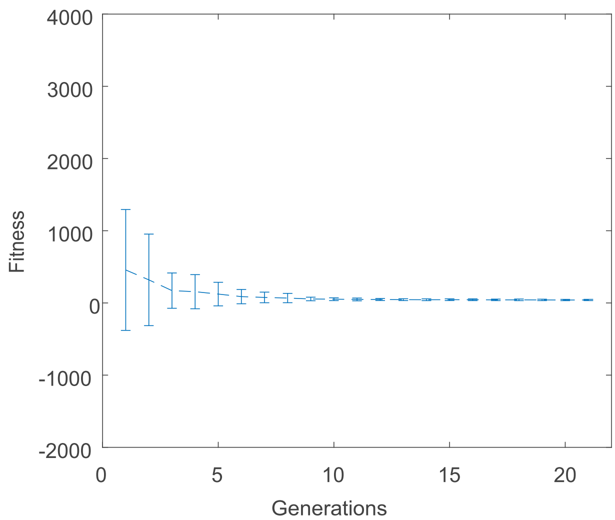

Fig. 13. Stochastic uniform selection

population size rather than a larger number of generations with a small population size.

C. Selection Mechanisms

One of the many possibilities for the parametrization of genetic algorithms is the choice of selection mechanism. In order to investigate the influence of the selection mechanism on the outcome of the optimization, two experiments with identical settings but different selection mechanisms were run. In the first experiment, the default *stochastic universal selection* was used, whereas *tournament selection* was employed in the second. With *stochastic universal selection*, individuals are selected with a probability corresponding to their fitness. By contrast, *tournament selection* picks a predefined number of individuals and selects those with the highest fitness value for the next generation. New tournament participants are selected and transferred to the next generation until it is complete.

Figs. 12 and 13 show a comparison of the results obtained with the two selection mechanisms. Each algorithm worked with a population size of 100. The faster convergence of the algorithm with tournament selection is clearly visible in Fig. 12.

V. CONCLUSIONS

In this article, we presented a method for the design optimization of soft pneumatic actuators using genetic algorithms. This work extends our previous and ongoing research [25] on the automated design and modeling of modular soft robots. In particular, we proposed a framework for a software tool that allows for the automatic optimization of soft pneumatic actuators and other soft robot modules. The validity and utility of the approach was demonstrated through optimization of a soft bending actuator. Compared to the initial design, the achievable bending angle of the actuator was significantly increased, while the ballooning was kept as low as possible. To further demonstrate the performance of the stochastic optimization approach, we analyzed the convergence of a series of trials thereby confirming the robustness of the method. Moreover, we analyzed the influence of the population size and the selection mechanism on the outcome of the optimization. The results indicate that a larger population size leads to a more even coverage of the parameter space and ultimately to better solutions, while the tournament selection mechanism produced faster convergence than stochastic uniform selection.

In future works, we will further explore the performance of the method and extend the capabilities of the tool to support other soft robot modules. Also, we aim at integrating the optimization and model generation parts of the framework into a comprehensive design, modeling, and simulation tool for modular soft robots. Whereas other approaches have mostly focused on either design optimization or kinematic modeling, the proposed tool is expected to yield models for optimized soft robot designs amenable to online-computation and control.

REFERENCES

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

[2] G. Gerboni, T. Ranzani, A. Diodato, G. Ciuti, M. Cianchetti, A. Menciassi, Modular soft mechatronic manipulator for minimally invasive surgery (MIS): Overall architecture and development of a fully integrated soft module, Meccanica, vol. 50, no. 11, pp. 2865–2878

[3] E. T. Roche, R. Wohlfarth, J. T. B. Overvelde, N. V. Vasilyev, F. A. Pigula, D. J. Mooney, K. Bertoldi, C. J. Walsh, A Bioinspired Soft Actuated Material, Advanced Materials, vol. 26, no. 8, 2014, pp. 1200–1206

[4] J. Ortega Alcaide, L. Pearson, M. Rentschler, Design, Modeling and Control of a SMA-Actuated Biomimetic Robot with Novel Functional Skin, Proc. IEEE Int. Conf. on Robotics and Automation, Singapore, June, 2017

[5] P. Polygerinos, Z. Wang, K. C. Galloway, R. J. Wood, C. J. Walsh, Soft robotic glove for combined assistance and at-home rehabilitation, Robotics and Autonomous Systems, vol. 73, 2015, pp. 135–143

[6] H. Zhao, R. Huang, R. F. Shepherd, Curvature control of soft orthotics via low cost solid-state optics, Proc. IEEE. Int. Conf. on Robotics and Automation, May, 2016, pp. 4008–4013

[7] H. K. Yap, J. H. Lim, J. C. H. Goh, C.-H. Yeow, Chen-Hua, Design of a Soft Robotic Glove for Hand Rehabilitation of Stroke Patients With Clenched Fist Deformity Using Inflatable Plastic Actuators, Journal of Medical Devices, vol. 10, no. 4, 2016, 044504

[8] J. Amend, H. Lipson, The JamHand: Dexterous Manipulation with Minimal Actuation, Soft Robotics, vol. 00, no. 00, 2017, pp. 1–11

[9] Z. Deng, M. Stommel, W. Xu, Soft Robotics Technology and a Soft Table for Industrial Applications, in Robot Intelligence Technology and Applications 4: Results from the 4th International Conference on Robot Intelligence Technology and Applications, J. H. Kim, F. Karray, J. Jo, P. Sincak, H. Myung (eds.), Springer, Cham, pp. 397–409

[10] D. Ross, M. P. Nemitz, A. A. Stokes, Controlling and Simulating Soft Robotic Systems: Insights from a Thermodynamic Perspective, Soft Robotics, vol. 00, no. 00, 2016, pp. 1–7

[11] P. Moseley, J. M. Florez, H. A. Sonar, G. Agarwal, W. Curtin, J Paik, Modeling, Design, and Development of Soft Pneumatic Actuators with Finite Element Method, Adv. Eng. Mater., vol. 18, no. 6, 2015, pp.6–11

[12] F. Connolly, C. J. Walsh, K. Bertoldi, Automatic design of fiber-reinforced soft actuators for trajectory matching, PNAS, vol. 114, no. 1, 2016, pp. 51–56

[13] E. Thompson-Bean, R. Das, A. McDaid, Methodology for designing and manufacturing complex biologically inspired soft robotic fluidic actuators: prosthetic hand case study, Bioinspiration & Biomimetics, vol. 11, no. 6, 2016, 066005

[14] Y. Sun, H. K. Yap, X. Liang, J. Guo, P. Qi, M. H. Ang, C.-H. Yeow, Stiffness Customization and Patterning for Property Modulation of Silicone-Based Soft Pneumatic Actuators, Soft Robotics, vol. 00, no. 00, 2017, pp. 1–10

[15] G. Runge and A. Raatz, A framework for the automated design and modeling of soft robotic systems, CIRP Annals Manufacturing Technology, vol. 66, no. 1, 2017, pp. 9–12

[16] G. Runge, M. Wiese, L. Günther, A. Raatz, A framework for the kinematic modeling of soft material robots combining finite element analysis and piecewise constant curvature kinematics, Proc. IEEE Int. Conf. on Control, Automation, and Robotics, April, 2017, pp. 714

[17] D. Zambrano, M. Cianchetti, and C. Laschi, The morphological computation principles as a new paradigm for robotic design, in Opinions and Outlooks on Morphological Computation (H. Hauser, R. M. Füchslin, and R. Pfeifer, eds.), 2014, pp. 214–225

[18] A. J. Clark, X. Tan, P. K. McKinley, Evolutionary multiobjective design of a flexible caudal fin for robotic fish, Bioinspiration & Biomimetics, vol. 10, no. 6, 2015, 065006

[19] J. C. Spall, Stochastic Optimization, in Handbook of Computational Statistics, Springer, Berlin, Heidelberg, 2012

[20] S. Cagnoni and L. Vanneschi, Evolutionary Computation: A Brief Overview, in Genetic and evolutionary computation, Wiley, Chichester West Sussex, 2011

[21] A. Chipperfield, Introduction to genetic algorithms, in Genetic Algorithms in Engineering Systems, 1997, pp. 1–45

[22] M. Srinivas, L. M. Patnaik, Genetic algorithms: A survey, Computer, vol. 27. no. 6, 1994, pp. 17–26

[23] L. Booker, Improving search in genetic algorithms, in Genetic algorithms and simulated annealing, L. Davis (ed.), Morgan Kaufmann Publishers, 1987, pp. 61-73

[24] C. Reeves, Genetic Algorithms, in Handbook of Metaheuristics, F. Glover and G. A. Kochenberger (ed.), Springer US, Boston MA, 2003, pp. 55–82

[25] G. Runge, M. Wiese, A. Raatz, FEM-Based Training of Artificial Neural Networks for Modular Soft Robots, submitted to IEEE Int. Conf. on Robotics and Biomimetics, December, 2017