Project:

BioMeld

Grant Agreement (GA) No. 101070328

"A MODULAR FRAMEWORK FOR DESIGNING AND PRODUCING BIOHYBRID MACHINES"

Call: HORIZON-CL4-2021-DIGITAL-EMERGING-01
Type of action: Research and Innovation action (RIA)
Start date of project: 01/10/2022 Duration: 36 months

D4.3: Performance simulations and analysis of results

Deliverable factsheet

Project title | Acronym | Number		A Modular Framework for Designing and Producing Biohybrid Machines | BioMeld | 101070328	
Due Date:	31/03/2024	**Date of submission:**	30/03/2024
Month of Project	18	**Month of submission:**	18
Title of deliverable:	D4.3: Performance simulations and analysis of results	**Work Package:**	WP4 – Manufacturing - simulation feedback loop and BIMC establishment
Dissemination level:	Public	**Version/Status**	V1
Deliverable leader (Name|Organisation)	UWE Bristol	**Editor(s)**	Michail-Antisthenis Tsompanas, Andrew Adamatzky, Hugo Alcaraz, Igor Balaz
Contribution of partners	UNSPF has reviewed the initial draft and contributed to following versions of this deliverable.		
Final review and approval	All partners in the consortium		
Keywords	Biohybrid machines, simulation, modelling, optimization, statistical analysis		

Abstract	
	The methodology to evaluate the predictability of the models produced under the BioMeld project and their robustness in the designing process is introduced here. Namely, simulation results will be compared with the corresponding experimental results for each of the test cases with measures that are described in this Deliverable. Also, the methodology of training Neural Networks (NNs) to predict the behaviour of the system is briefly described. This is the first version of the Deliverable, where the methodology and statistical measures are described. The second version of the Deliverable that is due at the end of the project (M36), will include detailed analysis of experimental and simulation results with the aforementioned statistical measures and the description and implementation of the proposed NNs.

Document change history

Date	Authors	Description
29/02/2024	Michail-Antisthenis Tsompanas, Hugo Alcaraz	Initial version drafted.
21/03/2024	Michail-Antisthenis Tsompanas	Additions throughout the text.
22/03/2024	UWE, UNSPF	Additions throughout the text.
30/03/2024	Michail-Antisthenis Tsompanas, Andrew Adamatzky, Igor Balaz	Final version agreed.

CONSORTIUM

	Name	Short Name	Country
1.	UNIVERZITET U NOVOM SADU, POLJOPRIVREDNI FAKULTET NOVI SAD	UNSPF	Serbia
2.	SCUOLA SUPERIORE DI STUDI UNIVERSITARI E DI PERFEZIONAMENTO S ANNA	SSSA	Italy
3.	FUNDACIO INSTITUT DE BIOENGINYERIA DE CATALUNYA	IBEC-CERCA	Spain
4.	SMART SENSING S.R.L.	SMART SENSING	Italy
5.	UNIVERSITA DEGLI STUDI DI CAGLIARI	UNICA	Italy
6.	LEVERETTE LANCE	Lance Leverette	Belgium
7.	The University of the West of England	UWE Bristol	United Kingdom

EXECUTIVE SUMMARY

The utilization of simulations in the design process of a diverse range of systems has proved to be advantageous in a variety of fields, especially when compared with building actual prototypes. Firstly, simulations can operate faster and in wider parametric ranges than real-life prototypes. Secondly, the simulated system is often inexpensive when compared with a real-world experiment, as it does not require physical materials and specialized resources. Thirdly, simulations render a safer environment for testing, as they eliminate potential risks such as personnel injuries or equipment damage. Lastly, simulations can be more informative as they provide detailed data and insights of the systems' behaviour, since the simulated environment is easier to control and measure than a real-world scenario. Due to all the aforementioned advantages, simulations are powerful tools that are utilized for system design and optimization, despite the fact that their accuracy may not be ideal most of the times.

To evaluate the predictability of the models produced under the BioMeld project and their robustness in the designing process, simulation results will be compared with the corresponding experimental results for each of the test cases. The suitability of the simulator will be evaluated based on the metrics and methodology described in detail in this Deliverable. Specifically, Spearman correlations will be calculated and compared by Zou's method for dependent overlapping correlations. Then, the exact McNemar's test for correlated proportions will assess the agreement between simulated and experimental data.

To analyse the predictability over the Bio-Hybrid Machine (BHM) behaviour, a primary choice is the penalized likelihood logistic regression. Another alternative will be parameter tuning by cross-validation. To quantify the extent of correlation between the prediction results and experimental ground truth, we will use several metric criteria, such as the Brier score, Calibration Curve, Concordance probability and Adjusted R^2.

On top of the utilization of simulations for the design process of BHMs, they can also be used in the development of their controlling strategies. Recent advancements in combining learning and simulations have demonstrated encouraging outcomes in addressing real-world challenges, such as autonomous driving, grasping, and in-hand manipulation, using policies exclusively trained in simulation. However, there are key challenges when moving from pure simulation to real-world application.

To address these challenges in this project, the obtained simulation results of the BHM behaviour will be compared with corresponding validation experiments and error time series will be stored for further use. Namely, they will be used to develop Policy DNN (deep neural network) and Actuator DNN that will guide further refining of the BHM design. A Policy DNN will be trained using the so-called *curriculum* method (that is briefly described in this Deliverable). It will be implemented because in DNNs finding the right penalty values on joint torque and velocity is far from a trivial task. Low penalty values often result in physically impossible motions while high penalty results in a standing behaviour, because it is already a good local minimum when there is a high penalty associated with motion. With a *curriculum* we will shape the initial cost landscape in a way that the policy is strongly attracted to the desired motion policy and later polish the motion to satisfy the other criteria. An Actuator DNN will be trained so that its output is the estimated torque at the catheter joints, given a history of position errors (the actual position subtracted from the simulated position) and velocities.

This is the first version of this Deliverable, where the methodology and statistical criteria are described. The second version of the Deliverable that will be produced at the end of the project (M36), will include detailed analysis of experimental and simulation results with the aforementioned statistical measures, along with the description and implementation of the proposed DNNs.

LEGAL NOTICE
This project has received funding from the European Union’s Horizon Europe research and innovation programme under grant agreement number 101070328.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
BHM	Biohybrid machine
NN	Neural network
DNN	Deep neural network
MLP	Multilayer perceptron
CI	confidence interval
AUC	area under the curve
FEM	finite element model

1 Introduction

State-of-the-art simulators are still not powerful enough to mimic with sufficient precision many of the key physical phenomena that dictate natural processes, i.e. deformation and impact. There are two popular approaches to tackle this issue, known as the **reality gap**: (i) to focus towards enhancing models with higher simulation precision and (ii) to focus towards artificial intelligence (AI) algorithmic methods that utilise abstract and inexact simulations as training targets.

Achieving higher simulation precision plays a significant role against the problem of limited extrapolation and generalization, i.e. application of patterns or trends to broader or future scenarios. Nonetheless, high accuracy on simulations is not a trivial task and requires great amount of computational resources, while it can be impossible to simulate physical processes without some level of abstraction, i.e. omitting some details and mimicking the real process at a target level of accuracy. Thus, inexact simulations may be used as sufficient tools that provide resilient evidence for decision making.

On the other hand, the approach of utilizing abstract and inexact simulations in training reinforcement learning algorithms allows the extraction of knowledge that will be important in the phase of developing a system. Thus, the provided value functions and learned policies are of high usefulness, despite the fact that they may be affected by the accuracy of the simulation. Moreover, this approach can be further enhanced by performing a definite uncertainty analysis, for instance identifying the areas that the simulator produces less accurate future state predictions. Although this is a complicated procedure, it has strong advantages on pinpointing preliminary failure areas and tackling them by acquiring more real-life data to improve the simulator specifically in these areas.

Simulation of physical processes and AI techniques can be considered as complementary methods, provided that Deep Learning requires the training over large amounts of data and these amounts are easily produced by simulations. Moreover, when compared with real-life data, simulated data are acquired faster, safer and with less financial resources, thus, lowering the entry barrier for researchers to study specific fields of science. Nonetheless, a significant challenge remains the transferability of predictive models from the *in silico* to *in vitro* or even *in vivo* experiments.

Despite the existence of the **reality gap**, large-scale automation relies heavily on computer simulation for reliable performance and continuous improvement of complex processes. Simulation has been successfully utilized in various industrial applications, as it can guide the processes of designing, validating and gaining insights of the underlying real systems. Some examples of robust simulation applications in industrial settings are finite element methods, computer-aided design and electrical circuit simulation.

It is well-established that simulation outcomes alone cannot be conclusive, thus, real-world testing is necessary to validate automation solutions. However, simulation outcomes offer valuable insights, offering repeatable benchmarks and aiding in algorithm analysis. Hence, simulation-only studies should not be easily dismissed outright, but compelling evidence of their real-world relevance needs to be provided.

As a result, in this Deliverable, the methods of thorough comparison between simulation results and the corresponding experimental data for each test case are introduced. As this is the first version of the Deliverable, the methods are briefly described and in the second version of this Deliverable (due on M36) the results of this comparison will be presented. Furthermore, an initial explanation of the methodology employed to train DNNs for predicting the system's behaviour will be provided here.

2 DESCRIPTION OF WORK AND MAIN ACHIEVEMENTS

To evaluate the predictability of the software models developed within the project and, thus, their robustness as optimization designing tools, the simulation results and the relevant experimental results need to be directly compared. More specifically, to ensure the trustworthiness of the different modules of the software framework (SF), a set of *in vitro* experimental tests will be imitated by their *in silico* counterparts. The comparison, under several statistical metrics, will be one aspect of this Deliverable. The other aspect will be the utilization of the laboratory produced experimental results as a database for training DNNs that will be employed in the automatic machine-learning based optimization module of the SF (as described in Task 2.6). Thus, it is important to emphasise that the two sub-tasks previously described are not isolated, since the output of the first one will be the input of the latter.

In the next subsections, we describe the statistical tests that we will implement for analysing the degree of correspondence between the simulator outcome and the experimental outcome (or different simulators outcomes). Also, we introduce the approach that will be utilised to design the policies that command bio-hybrid machines (BHMs) behaviour.

2.1 EVALUATION OF THE SIMULATION FRAMEWORK

Manufacturing BHMs and testing them *in vivo* or *in vitro* implies the investment of considerable time and material resources, along with the required significant human effort and expertise. As a result, it is imperative to adequately design and optimize the BHM morphologies before attempting to manufacture the final product. One way to achieve this is by developing efficient and accurate simulator that will replicate the behaviour of the BHM morphologies in specific scenarios and environments.

Minimizing the **reality gap** (Salvato et al., 2021) will increase the reliability of the SF to design optimized and efficient BHMs morphologies. Thus, the correspondence between simulated results and real experiments under the same environmental conditions will be considered under three aspects. First, the correlation analysis of both real and simulated results with the same independent parameter will be studied. Second, analysis of the predicted behaviour of the BHM in SF with alternative values of independent parameter will be performed. Finally, the extent of correlation between predicted results from the SF and experimental ground truth results (not evaluated until this step) will be investigated using several metric criteria.

2.1.1. Correlation Analysis

Initial experimental and simulated results will be compared to evaluate the predictability of the SF in mimicking BHM morphologies. Namely, tree metrics will be utilised to analyse the specific SF module performance: (i) Spearman's rank correlation coefficient, (ii) Zou's method, and (iii) McNemar's test.

Spearman correlation coefficient [1]: This non-parametric metric is commonly used to calculate the strength of a monotonic relationship between paired data (Spearman, 2010). The coefficient is denoted by r_s and is defined in the $[-1.0, 1.0]$ range. The closer to -1 or +1 the value of r_s is, the stronger the monotonic relationship between the paired data. On the other hand, if $r_s = 0$, there is no monotonic relationship between the data being compared. Furthermore, no hypothesis is required that the data analysed are normally distributed.

For instance, this measure can determine how strong is the correlation of the force produced by the active component of the BHM and the angle achieved by the catheter tip. It can provide the correlation of these parameters in an experimental setting and, possibly, a different correlation of these parameters in the simulator.

Zou's method [2]**:** The objective of this non-parametric approach consists of defining trustworthy confidence intervals for differences between two overlapping correlations (Zou, 2007). However, other approaches, such as independent correlations and non-overlapping correlations, can be managed by this technique. The outcome of this metric consists of two curves with a confidence interval (CI) each, showing whether there is an overlap between the curves. The overlap can be understood as the existence (or not) of the significant differences between the two curves (i.e., the correlations).

In order to adequately use this method, it is necessary to use the data of the xy correlation and the data of the xz correlation, where y and z are the dependent variables (in the example used here the actual angle achieved by the manufactured catheter and the simulated angle) and x is the independent variable (namely, the force of the active component of the BHM - both in reality and the simulation). Furthermore, the number of samples in both data sets should be the same.

Thus, if a monotonic correlation is observed, Zou's method will be employed to visually confirm whether significant differences exist between the experimental and simulated data.

McNemar's test [3]: This metric is non-parametric, and its purpose is to determine if there are significant differences in the proportions of categories in paired data (McNemar, 1947). It typically uses a *contingency table* to operate. The data that is being analysed should be mutually exclusive and nominal. The outcome of this approach consists of a p-value. If $p < 0.05$, there is a significant difference with 95% confidentiality. Otherwise, there are no significant differences between the data in the categories.

Table 1 depicts one possible structure of the contingency table that will be used to analyse the experimental and simulated data. The assumption is that a minimal target angle is aimed to be achieved by the catheter tip; given different parameters in experimental and simulated tests. So, the required variables are gathered to populate Table I; i.e., one nominal variable with two categories (achieved the minimal target angle or not) and one independent variable (force produced by active component of the BHM) with two dependent groups (laboratory and simulated tests).

Table I: Contingency table for experimental and simulated data.

Variables	Experimental	Simulated
Minimal target achieved		
Minimal target not achieved		

2.1.2. Analysis of predicted behaviour

To analyse the predicted behaviour of the BHM, the primary approach is *penalised likelihood logistic regression* (Firth, 1993). This technique is utilised to alleviate the effects of separability, small sample sizes, and bias of the parameter estimates in maximum likelihood estimation. When *logistic regression* is utilised, the *penalised likelihood* produces consistent and finite regression parameters, whereas the maximum likelihood estimates are not present, due to a complete or almost complete separation. In terms of the example given, the probability of achieving the minimal target angle will be evaluated when the simulator is queried under specific variables.

Another technique, which is considered as an alternative to *penalised likelihood logistic regression*, is *hyperparameter tuning by cross-validation*. In the context of predictive models, cross-validation consists of validating the efficiency of a prediction model by creating multiple subsets of datasets called *folds* which are used to run an iterative training-evaluation procedure of models using different training and testing datasets each time (Allen, 1974). Regarding hyperparameter tuning (also, known as hypermarameter optimisation), it focuses on finding the set of optimal hyperparameters for a predictive model. Cross-validation is regularly utilised to find the optimal hyperparameter configuration that enhances the performance of predictive models (Bergstra and Bengio, 2012). Examples of this technique are described as follows:

- ***Grid search cross-validation*** [4] (GSCV) selects the optimal set of hyperparameters by evaluating all possible combinations of the hyperparameters.
- ***Randomised search cross-validation*** [5] (RSCV) randomly generates n sets of hyperparameters, which are evaluated. The best set of hyperparameters is returned.
- ***Nested cross-validation*** [6] has internal and external cross-validation procedures. While the internal cross-validation is utilised to find the optimal set of hyperparameters through whether GSCV or RSCV, the external is used to define the subsets of the datasets. Then, the entire dataset is utilised to train the best model found using the optimal hyperparameter configuration.

2.1.3. Correlation between prediction results and experimental results

Four metric criteria will be employed to quantify the extent of correlation between the prediction results and experimental ground truth: (i) Brier score; (ii) Calibration curve; (iii) Concordance probability; and (iv) *Adjusted* R^2.

Brier score [7]: This metric focuses on evaluating the accuracy of probabilistic predictions (Brier, 1950). The output of this approach falls in the $[0.0, 1.0]$ range, and it can be interpreted as follows: the closer to 0 the value is, the better the prediction (i.e., suitable accuracy). On the other hand, if the value tends to be 1.0, the prediction may not be accurate. In order to use the Brier score, the data (i.e., the predictions) to be analysed should be binary or categorical and, consequently, mutually exclusive.

Calibration curve [8]: This metric is primarily utilised to evaluate how well the probabilistic predictions (usually binary) are calibrated (Wilks, 1990). The outcome of this metric is a plot showing the mean predicted probability on the *x*-axis. The *y*-axis presents the ratio of the positive predictions. The ideal curve is a linear straight line moving linearly. Similarly to the Brier score, to use this metric, the data being analysed should be mutually exclusive, which implies they should be binary or categorical.

Concordance probability [9]: The Concordance Probability is used to assess the degree of agreement between the predicted probabilities of an event and the actual outcomes (Harrell, Lee and Mark, 1996). This metric is usually calculated using the C-statistic or the area under the curve (AUC). The output range of the calculation is $[0.5, 1.0]$, where 0.5 can be interpreted as the predictions are randomly made, whereas 1.0 indicates a suitable predictive ability. In addition, when the Concordance Probability is being utilised, the data should be binary or categorical and mutually exclusive.

Adjusted R^2 [10]: The R^2 metric is typically used to determine if the proportion of the variance in the response variable can be explained by the predictor variables in the model (Steel and Torrie, 1960). The outcome of the calculation operates in the $[0.0, 1.0]$ range. It can be interpreted as follows: the closer to 0 the value is, the less explainable the response variable by the predictor variables is. On the other hand, the closer to 1 the value is, the more explainable the response variable by the predictor variables is. Despite its usefulness, R^2 exhibits certain limitations regarding measuring the impact of independent variables (i.e., predictors) on the correlation. The *Adjusted* R^2 metric is a variation of R^2 that shows whether adding additional predictors improves a regression model (Mordecai, 1930). In general, the *Adjusted* R^2 tends to decrease when adding a new predictor does not help improve the model's performance. On the other hand, it increases when adding a new predictor to the model improves its performance. Furthermore, *Adjusted* $R^2 \leq R^2$.

2.2. Developing deep neural network policies

All the data generated by the tests described previously will be used to develop the Policy DNN and the Actuator DNN that will help to refine the design of BHMs. Policy DNN will be trained by the *curriculum method*

(Bengio et al., 2009). Its objective consists of enhancing the learning process of NNs in significantly complex domains and, even, achieving better convergence. In general, this learning strategy works with tasks related to the problem domain, which start from very simple ones and gradually increase in complexity as the model progresses.

Regarding Actuator DNNs, they will be trained considering their output as an estimated torque at the catheter joints, given a history of position errors (the actual position subtracted from the simulated position) and velocities. It is essential to point out that the introduced approach assumes the dynamics of the actuators are independent of each other. Hence, models for each actuator can be trained separately.

The motivation behind *curriculum learning* implementation is that under the DNNs approach, finding the correct penalty values on joint torque and velocity is an intricate and arduous task. Usually, low penalty values lead to physically impossible motions. In contrast, a high penalty results in standing behaviour because it is already a suitable local minimum when a high penalty is associated with motion. The *curriculum* strategy shapes the initial cost landscape that allows the policy to be significantly attracted to the desired motion policy and subsequently helps to polish the motion to satisfy other criteria.

This strategy of training DNNs is inspired by how biological creatures (i.e. animals and humans) learn more efficiently. Namely, the tasks are not randomly presented to the learner, rather they are organized in an appropriate manner to match an increasing degree of complexity. Consequently, the same concept was applied in machine learning for a variety of problems (i.e. computer vision and language models) and important advancements towards enhanced generalization and faster convergence of the NNs were observed (Bengio et al., 2009).

Curriculum learning has more profound effect in Deep Learning as it is based on the conception of extracting feature hierarchies. Specifically, in Deep Learning the features in lower level can be combined and provide higher level features. Thus, it is not heavily relying on manually engineered features, but allows the automatic extraction of multiple levels of abstractions that enable the system to derive complex functions.

The idea of starting simple, acquiring sufficient competency in the easier subtasks and, then, progressively advancing to more difficult tasks, can be attributed to even earlier works (Elman, 1993). Nonetheless, the same concept was applied for robotic control and proved that gradually incrementing the difficulty of the task leads to promising results (Sanger, 1994). Despite the promising results, the definition of the *difficulty* of tasks and the shaping procedure (i.e. the pace of advancing to increased difficulty) need to be carefully designed. While there are some coherent approaches to do that (Bengio et al., 2009), more innovative *curriculum* strategies may render even higher advantages depending on the application. For instance, an example of a Teacher-Learner pair of Perceptrons (Derényi, Geszti and Györgyi, 1994) contributed towards that trajectory.

In a similar fashion, Teacher–Student Curriculum Learning (TSCL) framework (Matiisen et al., 2019) was proposed to automatically perform *curriculum learning*. The Student can be any target machine learning model, while the Teacher automatically selects appropriate subtasks for the Student to train on, with regards to its fast progression and avoid forgetting. As a result, the *curricula* does not require to be manually engineered, but, can be automatically discovered by a variety of available algorithms (Matiisen et al., 2019) that can be tested against each specific application to evaluate their suitability.

A similar concept of starting small is proposed in different Machine Learning methods, such as Neuroevolution (NE). NE is referring to the methodology of applying principles of Evolutionary Algorithms (EAs) to the process of training and optimizing ANNs (Stanley et al., 2019) and have been used in the implementation of the **3D evolvable simulator (Task 2.4)**. Some NE algorithms, i.e., NeuroEvolution of Augmenting Topologies (NEAT) (Stanley and Miikkulainen, 2002), have been proved to be more efficient than others, because of some critical factors. One of them is that they encourage the incremental growth of the complexity in solutions, on account of the initial populations being structures of minimal complexity, termed as *complexification*. Namely, starting simple before attempting more complex solutions. We tested this concept within the **3D evolvable simulator (Task 2.4)** and a total acceleration of the whole computation by 30% was achieved (Tsompanas, 2024).

3 Results

Extensive results of the application of statistical measures (described in Section 2.1) to compare experimental and simulation data will be presented in the second version of this Deliverable (due on M36).

Nonetheless, here the initial results of comparing two different simulators with different complexities will be presented. Namely, the outputs of the COMSOL Multiphysics v5.6 (COMSOL Inc., Sweden) structural finite element model (FEM) simulation of the catheter tip deflection will be compared with the Voxelyze[11] framework simulating the catheter tip under the same conditions/parameters. The configuration of simulating the catheter tip was defined by SSSA team and the results from COMSOL were already published (Salvatori et al., 2023).

In specific, module "solid mechanical" was used, and the mesh was set to normal (physics-controlled mesh, number of elements: 6065, minimum elements quality: 0.2046) after the evaluation of the mesh convergence. A model was defined consisting of a hollow cylinder with an outer radius of 1 mm (*D* = 2 mm), an inner radius of 0.75 mm (*d* = 1.5 mm) and a catheter tip length (*lc*) of 15 mm. Two axial load points were applied to the outer wall of the catheter (*p1* and *p2*), with the same magnitude and opposite direction, simulating the biohybrid actuator contraction behaviour as a strip of muscle tissue, with a length defined as *x1* (as illustrated in Fig. 1).

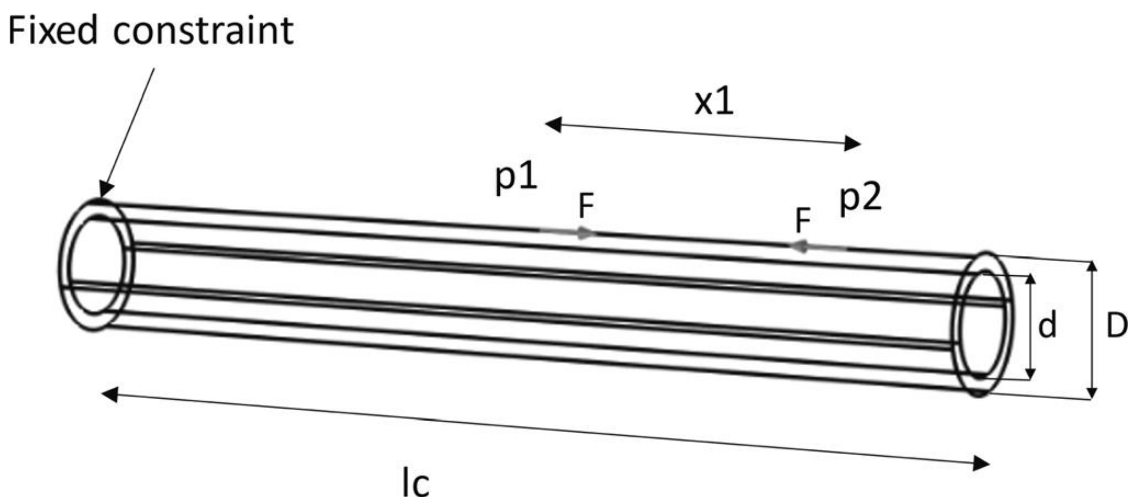


Figure 1 Depiction of the model implemented in COMSOL and in Voxelyze, and of the variables considered in the analysis. Adopted from Salvatori et al. (2023).

Table II. Parameters used in both simulators.

Parameter	Value in COMSOL	Value in Voxelyze
Lattice dimensions (voxel size)	N/A	0.2 mm
D (diameter)	2 mm	10 voxels
t (wall thickness)	0.2 mm	1 voxel
lc (catheter tip length)	15 mm	75 voxels
E (elastic modulus)	0.34 MPa	0.34 MPa
Density	1280 kg/m^3	1280 kg/m^3
Poisson's ratio	0.48	0.48
p1 (load points)	5 mm (varying)	33 % of *lc* (varying)
p2	10 mm (varying)	66 % of *lc* (varying)
x1	5 mm (varying)	33 % of *lc* (varying)
F (force)	100 µN (varying)	100 µN (varying)

The parameters used for the initial configuration for both COMSOL and Voxelyze are provided in **Table II**. The last four lines are mentioning the varying parameters to find the best configuration of the catheter tip. The configuration of the catheter tip in Voxelyze is depicted in Fig. 2.

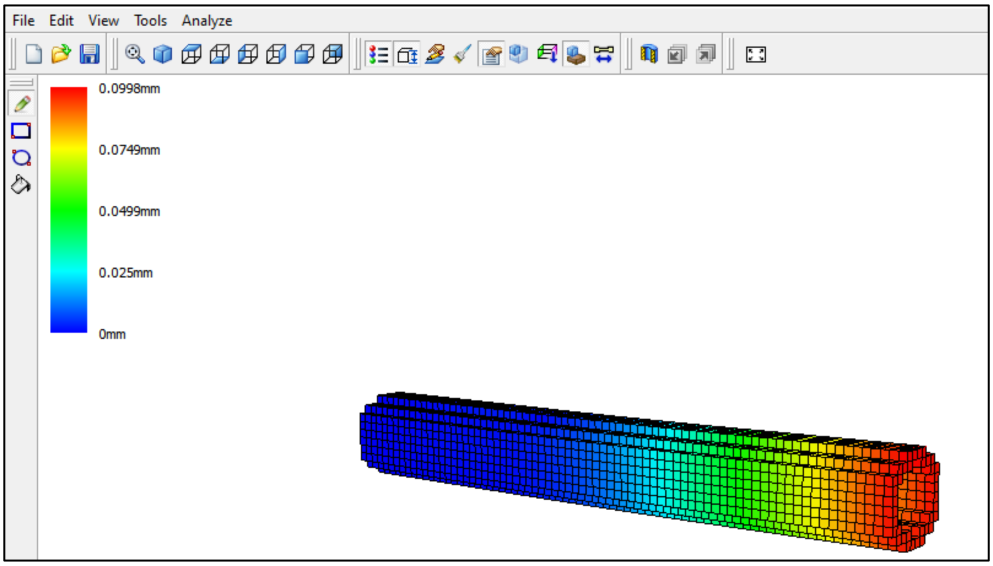
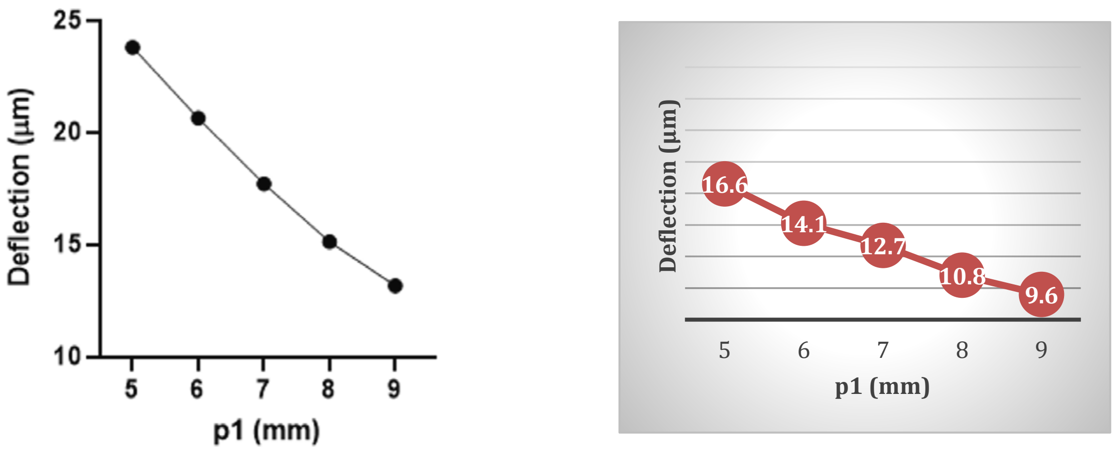


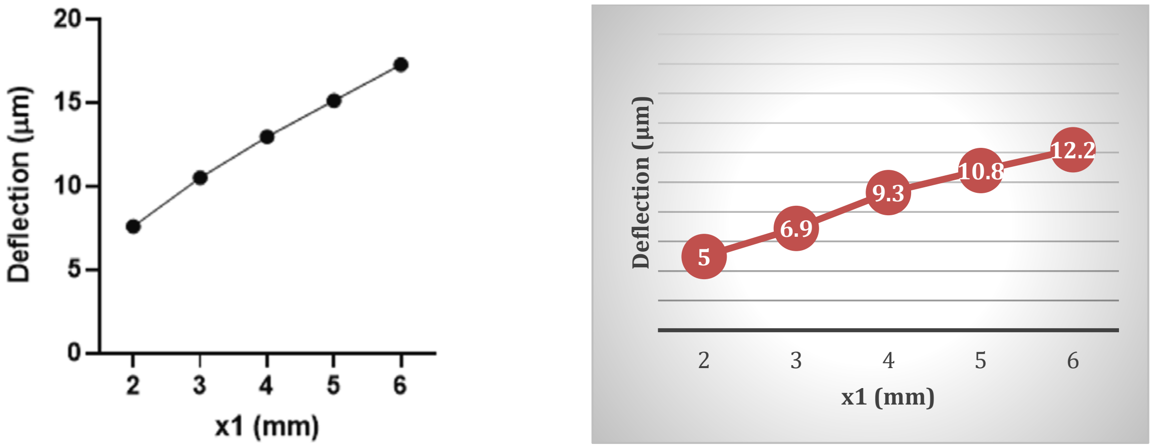
Figure 2. Configuration of the catheter tip in Voxelyze with the contour representing the total displacement at the end of the simulation.

Figure 3 depicts the maximum deflection when varying the distance between the application of the first load point and the fixed constraint (*p1*), Fig. 4 when varying the length of the biohybrid actuator (*x1*) and Fig. 5 when varying the applied force (*F*). All the aforementioned figures illustrate the outputs of COMSOL (as reported in by Salvatori et al. (2023)) and Voxelyze.



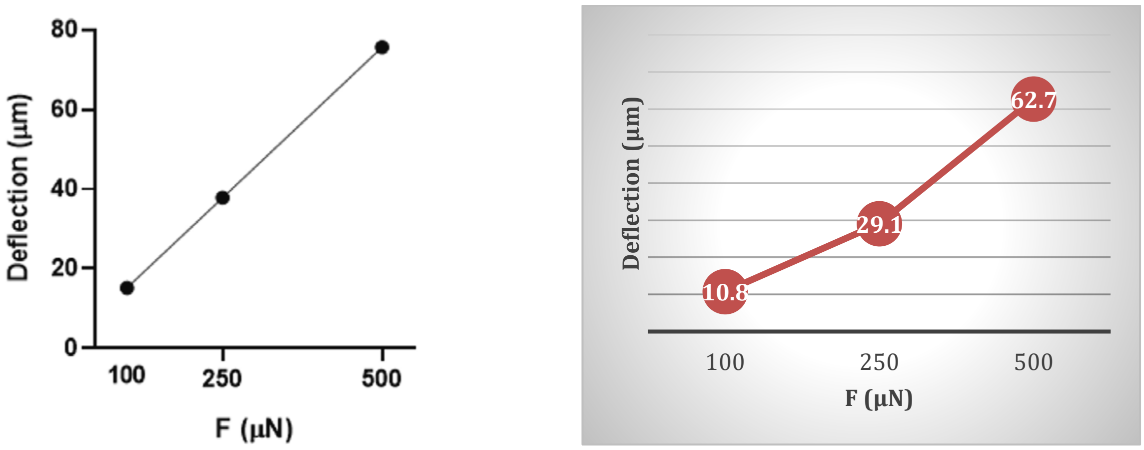
(A) (B)

Figure 3. Maximum deflection with varying distance between the application of the first load point and the fixed constraint (*p1*) for (**A**) COMSOL and (**B**) Voxelyze. Subfigure (**A**) is adopted from Salvatori et al. (2023).



(A) (B)

Figure 4. Maximum deflection with varying the length of the biohybrid actuator (*x1*) for (**A**) COMSOL and (**B**) Voxelyze. Subfigure (**A**) is adopted from Salvatori et al. (2023).



(A) (B)

Figure 5. Maximum deflection with varying the applied force (*F*) for (**A**) COMSOL and (**B**) Voxelyze. Subfigure (**A**) is adopted from Salvatori et al. (2023).

Comparing the outputs for each of the model configurations, it can be noticed that the same trend (i.e. increasing outputs/displacements with increasing parameters) with similar rates are provided by both COMSOL and Voxelyze. Despite the fact that Voxelyze produces significantly different values and is considered more abstract and, thus, less accurate than COMSOL, it can be utilized for Machine Learning applications that will lead to decision making and optimization designing tools. Due to the fact that Voxelyze is more simplistic, it will require less computational time to execute and will allow the training of algorithms in a trivial way.

4 Deviations from the Workplan / Future Steps

No deviations from the initial workplan.

5 Conclusions

Conclusions on the appropriateness of the selected measures to compare experimental and simulation data will be presented in the second version of this Deliverable (due on M36), since large datasets need to be assessed in order to derive meaningful outcomes.

6 References

Allen, D. M. (1974). The relationship between variable selection and data agumentation and a method for prediction. *technometrics*, *16*(1), 125-127.

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009, June). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning* (pp. 41-48).

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, *13*(2).

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly weather review*, *78*(1), 1-3.

Derényi, I., Geszti, T., & Györgyi, G. (1994). Generalization in the programed teaching of a perceptron. *Physical Review E*, *50*(4), 3192.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, *48*(1), 71-99.

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, *80*(1), 27-38.

Harrell Jr, F. E., Lee, K. L., & Mark, D. B. (1996). Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine*, *15*(4), 361-387.

Matiisen, T., Oliver, A., Cohen, T., & Schulman, J. (2019). Teacher–student curriculum learning. *IEEE transactions on neural networks and learning systems*, *31*(9), 3732-3740.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, *12*(2), 153-157.

Mordecai, E. (1930). Methods of correlation analysis. Wiley

Salvato, E., Fenu, G., Medvet, E., & Pellegrino, F. A. (2021). Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning. *IEEE Access*, *9*, 153171-153187.

Salvatori, C., Trucco, D., Niosi, I., Ricotti, L., & Vannozzi, L. (2023, July). A Novel Steerable Catheter Controlled with a Biohybrid Actuator: A Feasibility Study. In *Conference on Biomimetic and Biohybrid Systems* (pp. 378-393). Cham: Springer Nature Switzerland.

Sanger, T. D. (1994). Neural network learning control of robot manipulators using gradually increasing task difficulty. *IEEE transactions on Robotics and Automation*, *10*(3), 323-333.

Spearman, C. (2010). The proof and measurement of association between two things. *International journal of epidemiology*, *39*(5), 1137-1150.

Stanley, K. O., Clune, J., Lehman, J., & Miikkulainen, R. (2019). Designing neural networks through neuroevolution. *Nature Machine Intelligence*, *1*(1), 24-35.

Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, *10*(2), 99-127.

Steel, R. G. D., & Torrie, J. H. (1960). Principles and procedures of statistics. Principles and procedures of statistics.

Tsompanas, M. A. (2024, March). Incremental Growth on Compositional Pattern Producing Networks Based Optimization of Biohybrid Actuators. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)* (pp. 275-289). Cham: Springer Nature Switzerland.

Wilks, D. S. (1990). On the combination of forecast probabilities for consecutive precipitation periods. *Weather and forecasting*, *5*(4), 640-650.

Zou, G. Y. (2007). Toward using confidence intervals to compare correlations. *Psychological methods*, *12*(4), 399.

7 Appendix

Code availability:

1. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html
2. https://github.com/psinger/CorrelationStats/blob/master/corrstats.py
3. https://www.statsmodels.org/stable/generated/statsmodels.stats.contingency_tables.mcnemar.html
4. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
5. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

6. https://scikit-learn.org/stable/auto_examples/model_selection/plot_nested_cross_validation_iris.html
7. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.brier_score_loss.html
8. https://scikit-learn.org/stable/modules/generated/sklearn.calibration.calibration_curve.html
9. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html
10. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html
11. https://github.com/voxcraft/voxcraft-sim