RNN-BASED IDENTIFICATION OF SOFT CONTINUUM ARM DYNAMICS USING CURRICULUM LEARNING

PAVLO POMIN[1], TOMAS CAKURDA[1], ALEXANDER HOSOVSKY[1]

[1]Technical University of Kosice, Faculty of Manufacturing Technologies with the Seat in Presov, Slovakia

DOI: 10.17973/MMSJ.2025_09_2025044

pavlo.pomin@tuke.sk

In recent years, soft robotics has emerged as a key area of research in modern robotics. Among structural solutions, robotic arms play a key role. From the perspective of actuation, soft arms can be classified into two main types: cable-driven and pressure-driven arms. The latter often utilize pneumatic artificial muscles, whose characteristics — such as hysteresis and high degrees of freedom—make traditional dynamic modeling methods ineffective. This work focuses on the application of recurrent neural networks (LSTM, GRU), which are well-suited for processing time-series data. To improve model accuracy, a Curriculum Learning strategy with varying training block sizes is applied, enabling gradual learning.

The proposed methods are evaluated on the task of predicting the X-coordinate of the soft arm's endpoint. Results confirm the effectiveness of the selected approach and demonstrate the positive impact of Curriculum Learning on the ability of RNNs to model the dynamic behavior of soft robotic arms.

KEYWORDS

Curriculum learning, dynamic modelling, GRU, LSTM, pneumatic artificial muscles, soft robotic arm

1 INTRODUCTION

The use of new types of actuators in robot design opens up possibilities for expanding their functionality and mimicking nature-inspired movements. Pneumatic artificial muscles (PAMs) enable the conversion of air pressure into segment bending [Nguyen 2019, Yasa 2023]. A significant portion of soft fluidic robotic arm designs is based on a circular arrangement of PAMs around the longitudinal axis of the segment. Rigid mount frames are also used in the arm structure to help maintain the precise shape of the segment and enhance its stability during deformation. Both ends of the segment are fitted with end caps, to which the muscles and the axial element are attached. Individual segments can be interconnected, creating multi-segment arms with a greater range of motion [Godage 2016, Olson 2020, Peng 2019].

The outer layer of a pneumatic muscle consists of helically wound fibers interlaced at a defined angle. When compressed air is supplied, the muscle contracts in length and increases in diameter. At the maximum allowable pressure, it reaches its largest diameter and shortest length. Conversely, at minimal internal pressure, it has the smallest diameter and greatest length. The combination of the muscle's structure, the materials used, and the method of activation results in the nonlinear dynamics of its motion [Trojanova 2021, Kalita 2022].

The use of non-electric actuation combined with a soft structure enables the deployment of these robotic arms in a variety of tasks. Compared to traditional rigid arms, their design is significantly lighter and allows deformation at any point along the structure, increasing overall flexibility. These characteristics make them particularly well-suited for object manipulation [Fantoni 2014], general-purpose applications in robotics [Deaconescu, 2008], and operations in unstructured environments [Della Santina 2020].

However, practical applications of these arms require precise control and modeling methods. Alongside the development of soft robotics, modeling approaches have also evolved, with the choice of a suitable model for a given task depending primarily on the required accuracy and computational complexity [Yasa, 2023]. One of the most commonly used methods is the Piecewise Constant Curvature (PCC) model. Its fundamental principle is that, based on known pressure values in individual muscles, it is possible to determine the changes in their length and the forces acting on the segment. These values are then used to calculate the shape and spatial position of the segment [Sokolov 2024]. To determine the final configuration of a single segment, three main parameters are typically used: arc length (l), bending angle (θ), and the angle of the plane containing the arc (α) [Sokolov 2024, Yasa 2023].

Another effective method for modeling soft arms is the Cosserat rod theory. The previously mentioned PCC model is based on this theory. To reduce computational complexity, the PCC model simplifies several parameters such as bending, torsion, shear, and elongation. As a result, the Cosserat rod theory is not only a more accurate method for modeling soft arms, but also more computationally demanding. In applications where real-time performance is required, the PCC model is typically used instead [Armanini 2023, Gilbert 2019].

However, Cosserat rod theory is particularly important in the context of soft fluidic arms, especially where precise prediction of motion and deformation is required. These arms, actuated by pneumatic muscles, exhibit strongly nonlinear behavior resulting from the combination of pressure forces, material elasticity, and geometric nonlinearities. The Cosserat approach enables detailed modeling of continuous deformation along the entire segment, capturing more realistic curvatures and torsional effects than the simplified PCC model. As a result, Cosserat rod theory is primarily used in simulations and analyses where accuracy is crucial — for example, in the design of control algorithms or the validation of prototypes [Alessi 2024, Xun 2023].

In contrast to physics-based approaches such as Cosserat rod theory or the PCC model—which require detailed information about the segment's geometry and material properties (e.g., length, diameter, elasticity modulus, pressure, etc.)—there are also machine learning-based approaches [Yasa 2023]. These methods do not require a detailed description of the system's dynamics but instead learn the behavior of the arm directly from data. Among the most commonly used are multilayer perceptrons (MLPs) [Ogunmolu 2016], recurrent neural networks (RNNs), and their modern variants such as Long Short-Term Memory (LSTM) [George Thuruthel 2022, Thuruthel 2019] and Gated Recurrent Unit (GRU) [Bonassi 2021]. These models can be effective in approximating the nonlinear behavior of soft arms, especially when sufficient amounts of training data collected from the real system are available.

Architectures such as LSTM, GRU, and recurrent neural networks (RNNs) in general are well-suited tools for solving time-dependent tasks. Unlike standard neural networks, these models process not only the current input data but also information from previous time steps in the form of hidden states [Salem 2020]. This mechanism allows them to retain relevant information over multiple iterations, which is crucial for tasks involving sequential data.

The main difference between LSTM and GRU architectures lies in how they update the hidden state. GRU is a simplified

version of LSTM, making it less computationally demanding and generally faster to train. The resulting efficiency of an LSTM or GRU network depends on the complexity of the specific task [da Silva, 2024].

When working with nonlinear systems, it is often effective to use deep neural networks (DNNs) with an LSTM or GRU structure consisting of more than one layer. In practice, the so-called "black-box" modeling approach is commonly applied, where the model relies solely on input-output data and does not require knowledge of the system's exact dynamics [Bruder, 2019]. In contrast, methods such as PCC or Cosserat rod theory are considered "white-box" approaches, as they are based on physical principles and their results can be verified through analytical calculations. In black-box models, the internal structure of the model is unknown—the network learns the system's behavior exclusively from data. The accuracy of such models is evaluated by comparing their outputs with reference data and subsequently calculating statistical accuracy metrics [da Silva, 2024, Le 2021].

When training deep neural networks, it may happen that the model either fails to learn the relationship between inputs and outputs or learns it with insufficient accuracy. To minimize this issue, [Bengio 2009] introduced an approach for training models according to a predefined schedule, which the authors called Curriculum Learning (CL).

This approach is inspired by the principle of gradual learning—the model is first trained on simpler examples and only later on more complex ones. Such organization of the data can lead to faster convergence and higher accuracy of the resulting model. The effectiveness of this method is also supported by the study [Chakraverty 2022], in which Curriculum Learning (CL) was applied to a feedforward neural network for solving differential equations. The approach significantly improved the accuracy and stability of the computations. Further evidence of CL's effectiveness is provided by [Hacohen 2019], which demonstrates systematic performance improvements across different architectures and datasets compared to conventional training.

The article is further structured into four main chapters. Chapter 2 describes the experimental system and the data processing approach. Chapter 3 focuses on the methodology used, specifically the neural network architectures (LSTM, GRU), their parameters, and the training process, including the implementation of the Curriculum Learning method. Chapter 4 presents the results obtained, along with their analysis and a comparison of the individual models. The final chapter summarizes the main findings, supplemented by a brief discussion of observed trends and recommendations for future research.

2 EXPERIMENTAL SYSTEM AND DATA COLECTION

The data for training the neural networks were obtained from a real soft-fluidic robotic arm powered by compressed air. The arm's structure, shown in Fig. 1, consists of upper and lower plastic end caps produced using 3D printing, with a diameter of 100 mm (1), three pneumatic muscles of the type FESTO DMSP-10-300N-RM-CM (2), and a central axis. The axis is made of a latex tube filled with plastic granulate (3). All components are assembled into a single rigid structure using screw joints.

The pneumatic muscles have a diameter of 10 mm, a tube length of 300 mm, and a total length of 360 mm. They are symmetrically arranged around the circumference of a pitch circle with a diameter of 80 mm, with an angular spacing of 120° between each muscle.

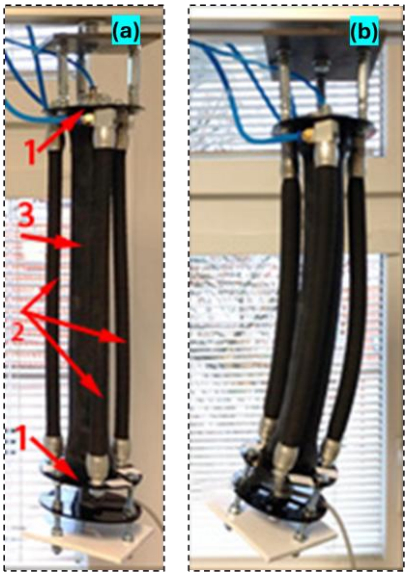

Figure 1. Soft-fluidic segment used for data collection for neural network training: segment in the non-activated state (a) with labeled components – plastic end caps (1), FESTO fluidic muscles (2), and central latex element with filler (3); segment in the activated state (b)

These muscles enable active movement of the segment depending on the pressure inputs. By changing the pressure in individual muscles, their length changes, which causes the entire segment to bend. The direction of bending depends on the ratio of forces acting in each muscle. To regulate the pressure in the actuators, SMC ITV0050-3BS regulators were used—one for each muscle. The source of compressed air was a TAGRED PROFESSIONAL compressor.

During the experiments, the position (X, Y, Z coordinates) and orientation (Roll, Pitch, Yaw rotations) of the segment's endpoint in space were monitored. A POLHEMUS PATRIOT position tracker was used to record this data. Power supply for the components was provided by a KORAD KD3005D power source, and signal processing was handled using a Humusoft MF644 I/O card.

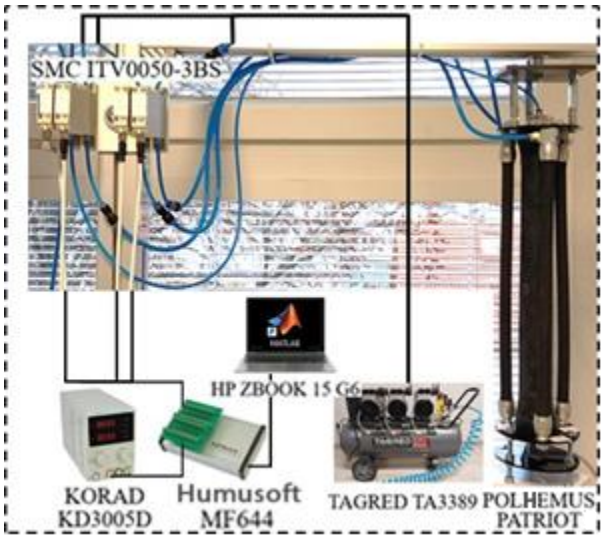

Figure 2. Experimental system with labeled components

The entire control system was designed in the MATLAB Simulink environment, which enabled segment control and measurement synchronization. The complete setup, including all components used, is shown in Fig. 2.

For the purpose of training data collection, three independent control signals were generated to regulate the pressure in each muscle. Each signal corresponded to one muscle and had

pressure values ranging from 1 to 7.5 bar. The duration of individual pressure pulses varied between 1 and 10 seconds.

During experiments on the real system, data were recorded at a frequency of 60 Hz. The actual pressure values in all three muscles (P1, P2, P3) were monitored, as well as the position (coordinates X, Y, Z) and orientation (angles Roll, Pitch, Yaw) of the segment's endpoint in space. The result was a dataset containing 216,001 input-output samples. The pressure values were further smoothed using a Savitzky–Golay filter with a first-degree polynomial and a window length of 23, in order to eliminate measurement noise. The origin of the coordinate system was placed at the tip of the arm in its non-activated state (P1 = P2 = P3 = 0 bar). All recorded endpoint positions were determined relative to this reference point.

3 METHODOLOGY

The training of neural networks (NNs) was carried out in the MATLAB environment using the Deep Learning Toolbox. To process time-dependent data, such as changes in pressure and position, recurrent neural networks with LSTM and GRU architectures were selected.

Before training, the input data (actual pressure values) were normalized to the range [0, 1], and the output data (coordinates and angles) were normalized to the range [-1, 1]. Training was performed using the Adam optimization algorithm, with the number of epochs set to 50 and the mini-batch size set to 32. The data were not shuffled during training (Shuffle = never), which preserved the temporal sequence of the inputs. The dataset was split into 70% for training, 15% for validation, and 15% for testing. The learning rate schedule was initially set to constant, with a starting learning rate of 0.001.

To evaluate prediction accuracy, three metrics were used: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). RMSE was chosen as the primary indicator, as it quadratically emphasizes larger errors, providing a more sensitive and rigorous assessment of model performance.

Before calculating the metric values, the neural network outputs were denormalized to ensure that the comparison with real data was relevant and accurate.

First, models with different architectures (LSTM, GRU), varying numbers of hidden layers (1 to 3), and different numbers of neurons in those layers (50 or 100) were compared. Within a single model, all hidden layers had the same number of neurons. All tested neural network models had the same inputs—actual pressure values in the individual muscles. The output from the neural network was then compared with data obtained from real-world experiments.

Additionally, both single-output models (X-coordinate only) and multi-output models (coordinates X, Y, Z and rotations Roll, Pitch, Yaw) were tested. Accuracy was evaluated exclusively based on the RMSE metric for the X-coordinate. The purpose of including multi-output models was to determine to what extent information about other coordinates and rotations influences the model's ability to accurately predict the X-coordinate.

The trained neural networks were divided into four groups based on their architecture (LSTM or GRU) and output type (X or XYZ_all). In the first training round, a total of 24 different models were created, combining various numbers of layers and neurons.

In the second round of training, different initial learning rates (InitialLearnRate = 0.01, 0.001, 0.0001) were tested for the best model from each group (based on the RMSE metric). The change applied only to new models, and the learning rate scheduling scheme was enabled (LearnRateSchedule = piecewise) with parameters LearnRateDropFactor = 0.5 and LearnRateDropPeriod = 10.

To explore the impact of training data organization on prediction accuracy, the Curriculum Learning (CL) method was applied to the original models. This approach was chosen as an alternative to random shuffling, which is unsuitable when working with time-dependent data. Instead, CL was implemented in the form of block-wise training, where the network gradually learned from sequentially ordered data segments based on their complexity.

In this phase, three models were created in each group (LSTM_X, LSTM_XYZ_all, GRU_X, GRU_XYZ_all), with training data divided into blocks of different sizes: 20,000, 15,000, and 5,000 samples per block. Each model was trained on a different number of blocks, allowing the effect of block size and segmentation on network performance to be evaluated. The Initial Learn Rate parameter was kept the same as in the first training round.

In the third round of training, a combination of the Curriculum Learning approach and scheduled learning rate reduction (LearnRateSchedule = 'piecewise') was tested. In each of the four groups, the best model from the second round—trained using Curriculum Learning and selected based on the RMSE value for the X-coordinate—was chosen as the baseline model.

The same Curriculum Learning method (with the same training data segmentation) was then applied to these models, but this time using the piecewise learning rate schedule with the same parameters as in the second round (LearnRateDropFactor and LearnRateDropPeriod).

As a result, 15 models were trained in each of the four groups (combinations of LSTM/GRU architecture and X/XYZ_all outputs), totaling 60 experimental models. Within the proposed framework, the following factors were tested for their impact:

- two recurrent neural network architectures (LSTM and GRU),
- two learning strategies (constant vs. scheduled learning rate),
- additional output information (coordinates + orientations),
- the Curriculum Learning method,
- and its combination with scheduled learning rate reduction.

This comprehensive approach established a robust foundation for analyzing the impact of individual parameters on the accuracy of modeling the motion of the soft-fluidic arm's endpoint.

4 RESULTS

The results are presented in the form of tables and graphs. The names of individual models contain key information about the training settings used. Specifically, the name indicates the output type (X or XYZ_all), the initial learning rate in the case of using a piecewise LearnRateSchedule (e.g., LR_01), the application of Curriculum Learning along with the block size (e.g., cur_20), and the number of hidden layers and neurons per layer (e.g., 1x50). If a parameter is not explicitly stated in the model name, it is assumed that the default setting was used. In this research, the default setting is defined as the LSTM architecture, a constant learning rate of 0.001, and no application of the Curriculum Learning method.

In the first round, a total of 24 models were trained - 12 with the LSTM architecture and 12 with the GRU architecture, each tested with X or XYZ_all outputs. The results are presented in Tables 1 and 2. The metric values were determined using the test dataset, which represented 15% of the original dataset. Based on these results, the best-performing models in terms of prediction accuracy were selected for the second round of training.

Table 1. Values of RMSE, MAE, and R^2 metrics for the training of LSTM models in the first round with X and XYZ_all outputs

Model name	RMSE [cm]	MAE [cm]	R^2 [-]
X_1x50	1.3408	1.0619	0.8677
X_1x100	1.1586	0.8873	0.9012
X_2x50	1.1029	0.8358	0.9105
X_2x100	1.0594	0.8073	0.9174
X_3x50	1.0937	0.8169	0.9120
X_3x100	1.1882	0.8921	0.8961
XYZ_all_1x50	1.3492	1.0362	0.8661
XYZ_all_1x100	1.2461	0.9381	0.8858
XYZ_all_2x50	1.2769	0.9663	0.8800
XYZ_all_2x100	1.1875	0.8803	0.8963
XYZ_all_3x50	1.2773	0.9453	0.8800
XYZ_all_3x100	1.1465	0.8347	0.9033

For single-output models (X), the top results were achieved by the X_2x100 models, with RMSE values of 1.0594 cm for LSTM and 1.0455 cm for GRU. For models with six outputs (XYZ_all), the highest accuracy was shown by the XYZ_all_3x100 models, with RMSE = 1.1465 cm (LSTM) and 1.0705 cm (GRU).

Table 2. Values of RMSE, MAE, and R^2 metrics for the training of GRU models in the first round with X and XYZ_all outputs

Model name	RMSE [cm]	MAE [cm]	R^2 [-]
X_1x50	1.1263	0.8924	0.9067
X_1x100	1.1333	0.8904	0.9055
X_2x50	1.1365	0.8824	0.9050
X_2x100	1.0455	0.8223	0.9196
X_3x50	1.0686	0.8287	0.9160
X_3x100	1.0649	0.8315	0.9166
XYZ_all_1x50	1.6746	1.3872	0.7937
XYZ_all_1x100	1.0904	0.8541	0.9125
XYZ_all_2x50	1.1759	0.9209	0.8983
XYZ_all_2x100	1.0831	0.8537	0.9137
XYZ_all_3x50	1.0979	0.8518	0.9113
XYZ_all_3x100	1.0705	0.8381	0.9157

These results are consistent with the other metrics (MAE and R^2), which are visualized in Fig. 3.

Among the 24 models, the best results were achieved by the X_2x100 model with GRU architecture, which recorded RMSE = 1.0455 cm, MAE = 0.8223, and R^2 = 0.9196.

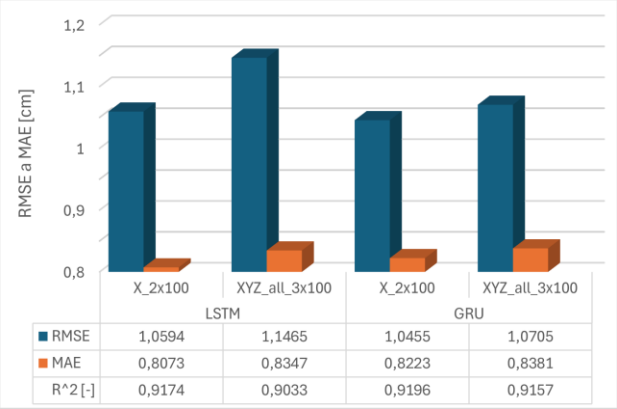

Figure 3. Comparison of RMSE, MAE, and R^2 metrics for the 4 best models from the first round of training

In the second round of training, the selected models were updated by changing the LearnRateSchedule parameter to "piecewise." The number of training epochs remained the same—50. In total, 12 models were trained in this way—3 for each group based on architecture (LSTM/GRU) and output type (X / XYZ_all).

Independently of this branch, the second round also tested the impact of the Curriculum Learning (CL) method, which was applied to 4 selected models from the first round. For each of these models, three versions were trained using different training block sizes—and thus a different number of blocks—resulting in 12 additional models. Each block was trained for the same number of epochs—50.

Table 3. Values of RMSE, MAE, and R^2 metrics for the training of LSTM models in the second round with X and XYZ_all outputs

Model name	RMSE [cm]	MAE [cm]	R^2 [-]
X_LR_01_2x100	1.1515	0.8777	0.9024
X_LR_001_2x100	1.2259	0.9446	0.8894
X_LR_0001_2x100	2.5952	2.0715	0.5045
X_cur_20_2x100	0.7461	0.5035	0.9590
X_cur_15_2x100	0.6907	0.4560	0.9649
X_cur_5_2x100	0.5858	0.4040	0.9748
XYZ_all_LR_01_3x100	1.2170	0.8876	0.8910
XYZ_all_LR_001_3x100	1.3022	0.9836	0.8752
XYZ_all_LR_0001_3x100	3.4193	2.7898	0.1399
XYZ_all_cur_20_3x100	0.7853	0.5049	0.9546
XYZ_all_cur_15_3x100	0.7256	0.4811	0.9613
XYZ_all_cur_5_3x100	0.6159	0.4182	0.9721

In total, 24 new models were created in the second round—half with a modified learning rate and half using the Curriculum Learning method.

The results from Table 3 and Table 4 show that changing the LearnRateSchedule parameter from a constant value to piecewise had no significant impact on model accuracy. This conclusion applies to both architectures (LSTM and GRU) and both output types (X and XYZ_all). In contrast, the application of the Curriculum Learning method demonstrated improved prediction accuracy of the X-coordinate across all four model groups. Notable improvements were observed particularly in the X_cur_5_2x100 and XYZ_all_cur_5_3x100 models for both LSTM and GRU architectures.

Table 4. Values of RMSE, MAE, and R^2 metrics for the training of GRU models in the second round with X and XYZ_all outputs

Model name	RMSE [cm]	MAE [cm]	R^2 [-]
X_LR_01_2x100	1.0592	0.8291	0.9175
X_LR_001_2x100	1.0770	0.8461	0.9147
X_LR_0001_2x100	3.1086	2.5323	0.2891
X_cur_20_2x100	0.6505	0.4365	0.9689
X_cur_15_2x100	0.6184	0.4061	0.9719
X_cur_5_2x100	0.5903	0.4008	0.9744
XYZ_all_LR_01_3x100	1.0924	0.8542	0.9122
XYZ_all_LR_001_3x100	1.1088	0.8687	0.9095
XYZ_all_LR_0001_3x100	3.2759	2.6689	0.2105
XYZ_all_cur_20_3x100	0.7049	0.4716	0.9634
XYZ_all_cur_15_3x100	0.7083	0.4791	0.9631
XYZ_all_cur_5_3x100	0.6307	0.4152	0.9707

For both architectures and output types, the best results were achieved by models trained with the smallest blockSize value (5000), i.e., with the highest number of training blocks.
For single-output models (X-coordinate), the best results achieved were as follows:
- LSTM: RMSE = 0.5858 cm, MAE = 0.4040 cm, R^2 = 0.9748
- GRU: RMSE = 0.5903 cm, MAE = 0.4008 cm, R^2 = 0.9744
Models with six outputs (XYZ_all) achieved slightly lower accuracy:
- LSTM: RMSE = 0.6169 cm, MAE = 0.4182 cm, R^2 = 0.9721
- GRU: RMSE = 0.6307 cm, MAE = 0.4152 cm, R^2 = 0.9707
A comparison of the model results is shown in Fig. 4. The percentage improvement of these models relative to the original models from the first round is presented in Table 5. Based on these results, the selected models were chosen for the third round of training.

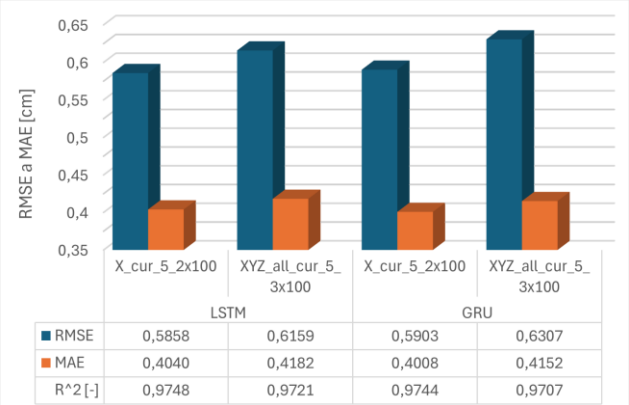

Figure 4. Comparison of RMSE, MAE, and R^2 metrics for the 4 best models from the second round of training

Among the 24 models in the second round, the best results were achieved by the X_cur_5_2x100 model with LSTM architecture.

Table 5. Percentage improvement in accuracy of the models X_cur_5_2x100 and XYZ_all_cur_5_3x100 (based on the RMSE metric) compared to the original models without the application of Curriculum Learning (X_2x100 and XYZ_all_3x100)

Architecture	LSTM	GRU
Model name	X_2x100	
X_cur_5_2x100	44.7045%	46.2800%
	XYZ_all_3x100	
XYZ_all_cur_5_3x100	43.5490%	41.0836%

In the third round of training, the same principle of scheduled learning rate reduction (LearnRateSchedule = 'piecewise') was applied as in the previous round. Within this approach, an additional 12 models were trained using the Curriculum Learning method.

Table 6. Values of RMSE, MAE, and R^2 metrics for the training of LSTM models in the third round with X and XYZ_all outputs

Model name	RMSE [cm]	MAE [cm]	R^2 [-]
X_LR_01_cur_5 _2x100	0.9458	0.7019	0.9342
X_LR_001_cur_5 _2x100	0.8199	0.5685	0.9505
X_LR_0001_cur_5 _2x100	1.0219	0.7377	0.9232
XYZ_all_LR_01_ cur_5_3x100	0.7108	0.4595	0.9628
XYZ_all_LR_001_ cur_5_3x100	0.716	0.4729	0.9623
XYZ_all_LR_0001_ cur_5 _3x100	0.8641	0.5544	0.9451

The results, shown in Table 6 and Table 7, indicate that for single-output models, the best performance was achieved by the X_LR_001_cur_5_2x100 model in both architectures (LSTM and GRU). For models with six outputs, the best results were obtained by the XYZ_all_LR_01_cur_5_3x100 model for the LSTM architecture, and by the XYZ_all_LR_001_cur_5_3x100 model for the GRU architecture.

Table 7. Values of RMSE, MAE, and R^2 metrics for the training of GRU models in the third round with X and XYZ_all outputs

Model name	RMSE [cm]	MAE [cm]	R^2 [-]
X_LR_01_cur_5 _2x100	0.7976	0.5784	0.9532
X_LR_001_cur_5 _2x100	0.6846	0.4697	0.9655
X_LR_0001_cur_5 _2x100	0.7966	0.5536	0.9533
XYZ_all_LR_01_ cur_5_3x100	0.7463	0.5338	0.959
XYZ_all_LR_001_ cur_5_3x100	0.6942	0.4693	0.9645
XYZ_all_LR_0001_ cur_ 5_3x100	0.7685	0.5228	0.9565

Despite the combination of approaches (CL + LR), a slight decrease in prediction accuracy for the X-coordinate was observed compared to the second-round results, as indicated by the values of the RMSE, MAE, and R^2 metrics.
Figure 5 shows a comparison of the four best models from the third round. Despite a decrease in the effectiveness of these models compared to the second round, the results are still better than those achieved by the models in the first round.

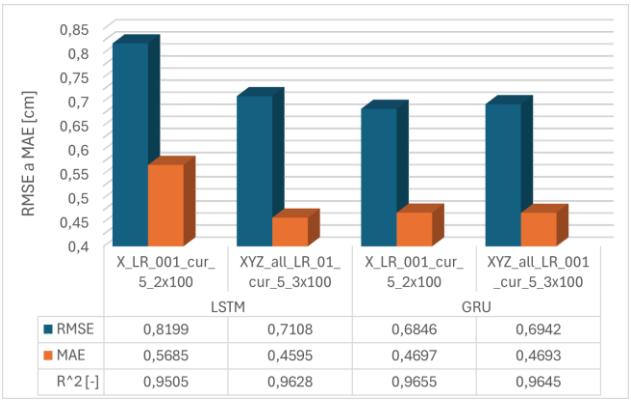

Figure 5. Comparison of RMSE, MAE, and R^2 metrics for the 4 best models from the third round of training

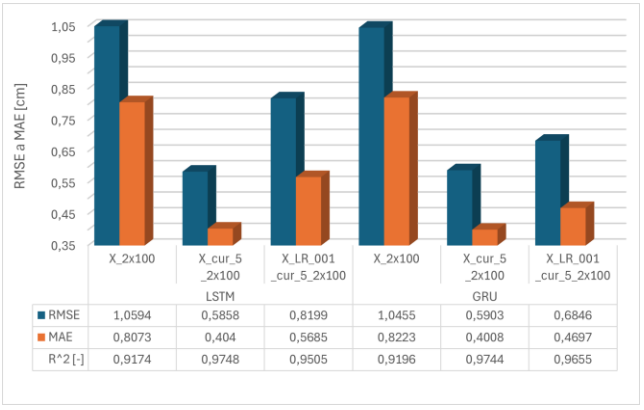

Figure 6. Comparison of RMSE, MAE, and R^2 metrics for the best models with X and XYZ_all outputs from each training round

Figure 6 summarizes the comparison of the six most successful models from the entire study based on three metrics—RMSE, MAE, and R^2. It is evident that the best results were achieved by the X_cur_5_2x100 model with LSTM architecture.

This comprehensive view confirms that the combination of a single-coordinate output (X), the application of Curriculum Learning with a small block size (blockSize = 5000), and a well-chosen architecture (2 hidden layers with 100 neurons) leads to the most effective model in the entire experiment.

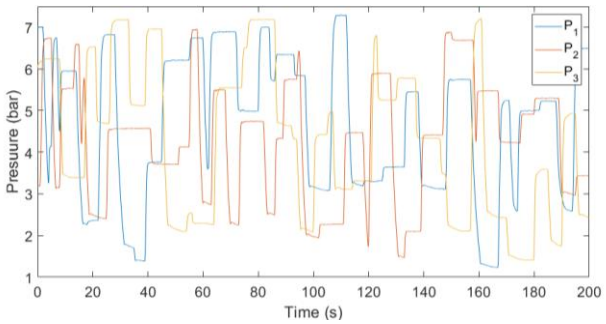

Figure 7. Pressure variation in individual muscles. Values used as input to the neural network for predicting the X-coordinate

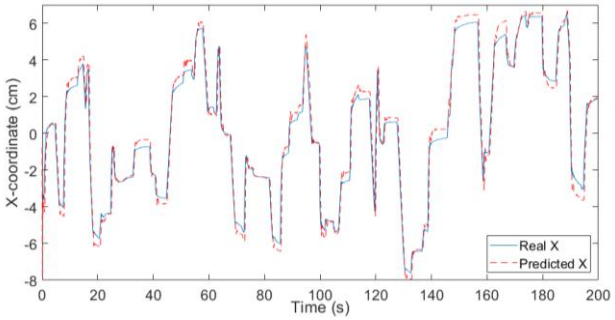

Figure 8. X-coordinate trajectory on test data. Prediction by model X_cur_5_2x100 with LSTM architecture

The input data consisted of actual pressure values are shown in Figure 7. Figure 8 illustrates the prediction accuracy of the X-coordinate on test data achieved by the most successful model (X_cur_5_2x100 with LSTM architecture). The model closely follows the real trajectory, especially in dynamic sections with position changes. Larger deviations occur mainly in stable segments where movement is minimal. Since dynamic sections last longer than static ones, the model has access to more training data for transitional states. This may lead to reduced accuracy in stable sections. Therefore, it is advisable to increase the proportion of stable segments in the training data to help the model better learn how to handle these scenarios.

5 DISCUSSION AND CONCLUSION

The achieved results confirm the ability of the proposed neural networks to effectively model the dynamics of soft-fluidic arms. The models were tested on predicting a single coordinate (X), which serves as a foundation for future extension to predicting additional coordinates and rotational angles. The calculated metrics demonstrate the impact of learning rate changes and data organization according to the Curriculum Learning approach on the final model accuracy. The results confirm that, with the number of epochs held constant, changing the learning rate did not lead to a significant improvement in model performance. The findings suggest that to enable effective learning at lower learning rates, the number of training epochs should be increased so that the network can better learn complex temporal dependencies.

The differences in performance between models with one output and those with six outputs clearly show higher accuracy in models working with a smaller number of output parameters. This may suggest that the dependencies between coordinates and angles do not have a significant positive impact on network accuracy. This research demonstrated that increasing the number of outputs without adjusting the architecture and hyperparameters leads to reduced neural network performance. On the other hand, the results indicate the potential of increasing the number of hidden layers and neurons as a solution for models with a larger number of output parameters.

The results obtained from the third round of training confirm the impact of the tested approaches on model performance. The combination of scheduled learning rate reduction and the Curriculum Learning (CL) method led to improved accuracy in all four models compared to their counterparts from the first round. Furthermore, the positive effect of CL alone on enhancing the prediction of the X-coordinate was also confirmed.

The results also suggest that the performance of the models in the third round was slightly lower than in the second round, highlighting the significant impact of learning rate settings. With the number of epochs (50) kept constant, models with a lower learning rate were unable to sufficiently adapt to the training data. Therefore, it becomes evident that when the learning rate is reduced, the number of epochs must be proportionally increased to ensure sufficient time for learning complex nonlinear dependencies.

It was also reaffirmed that models with a smaller number of outputs (e.g., only the X-coordinate) achieve better accuracy. In the case of models with six outputs, the unchanged network architecture limits performance, indicating the need to expand the network (increased number of layers and neurons) when working with a larger number of output parameters. When comparing the effectiveness of LSTM and GRU architectures throughout the entire experiment, it is not possible to definitively determine which architecture is superior. The results show that the choice of a suitable architecture (LSTM or GRU) depends on the specific task and context. This conclusion also applies to other tested approaches and hyperparameters—their effectiveness is conditioned by the nature of the data and the modeling objective.

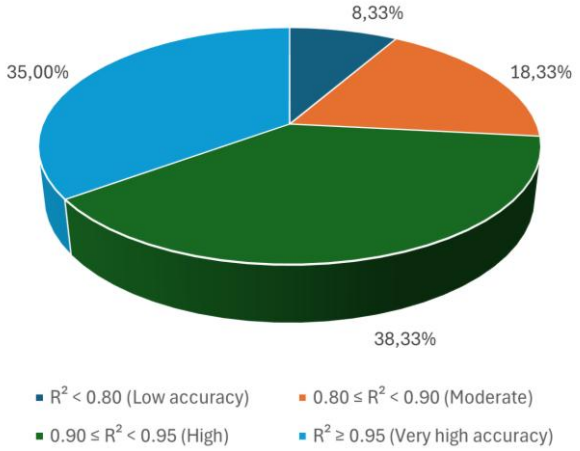

Figure 9. Distribution of Models by Prediction Accuracy (R^2)

To better evaluate overall model accuracy, a percentage-based classification of models was conducted according to their R^2 values, following commonly accepted thresholds [Chicco 2021]. Models with $R^2 \geq 0.95$ were considered highly accurate, while those below 0.80 were classified as insufficient. As shown in Fig. 9, this distribution clearly reflects the impact of training strategies and architectural choices on model performance.

The achieved results, along with the evaluation of models based on various parameters, demonstrate the high effectiveness of recurrent neural networks (RNNs) with LSTM and GRU architectures in modeling the dynamics of continuous (soft-fluidic) arms. The examined approaches, including

learning rate adjustment and the Curriculum Learning method, significantly contributed to improving model accuracy.

ACKNOWLEDGMENTS

This research was funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I03-03-V03-00075 and Scientific grant agency of the Ministry of Education, Research, Development and Youth of the Slovak Republic and the Slovak Academy of Sciences under the project VEGA 1/0061/23.

REFERENCES

[Alessi 2024] Alessi, C., et al. Rod models in continuum and soft robot control: a review. ArXiv, 2024. DOI: 10.48550/arXiv.2407.05886.

[Armanini 2023] Armanini, C., Boyer, F., Mathew, A.T., Duriez, C. and Renda, F. Soft Robots Modeling: A Structured Overview. IEEE Transactions on Robotics, 2023, Vol. 39, No. 3, pp. 1728-1748. DOI: 10.1109/TRO.2022.3231360.

[Bengio 2009] Bengio, Y., Louradour, J., Collobert, R. and Weston, J. Curriculum learning. In: Proc. of the 26th Annual Int. Conf. on Machine Learning (ICML '09); Montreal, Quebec, Canada; ACM Press. 2009, Vol. 60, pp 41-48.

[Bonassi 2021] Bonassi, F., Farina, M. and Scattolini, R. On the stability properties of Gated Recurrent Units neural networks. Systems & Control Letters, 2021, Vol. 157, 105049. DOI: 10.1016/j.sysconle.2021.105049.

[Bruder 2019] Bruder, D., Gillespie, B., Remy, C.D. and Vasudevan, R. Modeling and Control of Soft Robots Using the Koopman Operator and Model Predictive Control. In: Proceedings of Robotics: Science and Systems XV. Freiburg, Germany, 2019.

[Chicco 2021] Chicco, D., Warrens, M.J. and Jurman, G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Computer Science, 2021, Vol. 7, Article e623. DOI: 10.7717/peerj-cs.623.

[Danh Le 2021] Danh Le, T. A Hysteresis Black-Box Model of a Soft Robotic Arm Featuring Pneumatic Artificial Muscle. IAEME Publication, 2021, Vol. 12, No. 1, pp. 827-836. DOI: 10.34218/IJARET.12.1.2021.075.

[Santina 2020] Della Santina, C., Katzschmann, R.K., Bicchi, A. and Rus, D. Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment. International Journal of Robotics Research, 2020, Vol. 39, No. 4, pp. 490-513.

[Fantoni 2014] Fantoni, G., et al. Grasping devices and methods in automated production processes. CIRP Annals, 2014, Vol. 63, No. 2, pp. 679-701.

[Gilbert 2019] Gilbert. H.B. and Godage. I.S. Validation of an Extensible Rod Model for Soft Continuum Manipulators. In: 2nd IEEE Int. Conf. on Soft Robotics (RoboSoft); Seoul, South Korea, 2019, pp. 711-716. DOI: 10.1109/ROBOSOFT.2019.8722721.

[Godage 2016] Godage, I.S., Medrano-Cerda, G.A., Branson, D.T., et al. Dynamics for variable length multisection continuum arms. International Journal of Robotics Research, 2016, Vol. 35, No. 6, pp. 695-722.

[Hacohen 2019] Hacohen, G. and Weinshall, D. On the Power of Curriculum Learning in Training Deep Networks. In: Proceedings ICML 2019. ArXiv:1904.03626. DOI: 10.48550/arXiv.1904.03626.

[Kalita 2022] Kalita, B., Leonessa, A. and Dwivedy, S.K. A Review on the Development of Pneumatic Artificial Muscle Actuators: Force Model and Application. Actuators, 2022, Vol. 11, No. 10. DOI: 10.3390/act11100288.

[Nguyen 2019] Nguyen, P.H., Sparks, C., Nuthi, S.G., et al. Soft Poly-Limbs: Toward a New Paradigm of Mobile Manipulation for Daily Living Tasks. Soft Robotics, 2019, Vol. 6, No. 1, pp. 38-53. DOI: 10.1089/soro.2018.0065.

[Ogunmolu 2016] Ogunmolu, O., Gu, X., Jiang, S. and Gans, N. Nonlinear Systems Identification Using Deep Dynamic Neural Networks. ArXiv, 2016. DOI: 10.48550/arXiv.1610.01439.

[Olson 2020] Olson, G., Hatton, R.L., Adams, J.A. and Menguc, Y. An Euler–Bernoulli Beam Model for Soft Robot Arms Bent through Self-Stress and External Loads. Int. J. of Solids and Structures, 2020, Vol. 207, pp. 113-131. DOI: 10.1016/j.ijsolstr.2020.09.015.

[Peng 2019] Peng, Y., et al. Development of continuum manipulator actuated by thin McKibben pneumatic artificial muscle. Mechatronics, 2019, Vol. 60, pp. 56-65.

[Sahoo 2022] Sahoo, A.K. and Chakraverty, S. Curriculum Learning-Based Artificial Neural Network Model for Solving Differential Equations. In: Chakraverty, S. (ed.); Soft Computing in Interdisciplinary Sciences. Singapore: Springer, 2022, pp. 129-145. ISBN 978-981-16-4713-0.

[Salem 2022] Salem, F.M. Recurrent Neural Networks: From Simple to Gated Architectures. Cham: Springer International Publishing, 2022. DOI: 10.1007/978-3-030-89929-5.

[Sokolov 2024] Sokolov, O., et al. Study on the Piecewise Constant Curvature Kinematic Model of a Two-Segment Robot. MM Science Journal, 2024, No. 6. DOI: 10.17973/MMSJ.2024_12_2024105.

[Thuruthel 2019] Thuruthel, T.G., Shih, B., Laschi, C. and Tolley, M.T. Soft robot perception using embedded soft sensors and recurrent neural networks. Science Robotics, 2019, Vol. 4, No. 26, p. eaav1488. DOI: 10.1126/scirobotics.aav1488.

[Thuruthel 2022] Thuruthel. T.G., Gardner. P. and Iida. F. Closing the Control Loop with Time-Variant Embedded Soft Sensors and Recurrent Neural Networks. Soft Robotics, 2022, Vol. 9, No. 6, pp. 1167-1176. DOI: 10.1089/soro.2021.0012.

[Trojanova 2021] Trojanova, M., Cakurda, T., Hosovsky, A., Krenicky, T. Estimation of Grey-Box Dynamic Model of 2-DOF Pneumatic Actuator Robotic Arm Using Gravity Tests. Applied Sciences, 2021, Vol. 11, No. 10, Art. No. 4490.

[Xun 2023] Xun, L., Zheng, G. and Kruszewski, A. Cosserat-Rod Based Dynamic Modeling of Soft Slender Robot Interacting with Environment. ArXiv, 2023. arXiv:2307.06261.

[Yasa 2023] Yasa, O., et al. An Overview of Soft Robotics. Annual Review of Control, Robotics and Autonomous Systems, 2023, Vol. 6, No. 1, pp.1-29. DOI: 10.1146/annurev-control-062322-100607.

CONTACTS:

Ing. Pavlo Pomin
Ing. Tomas Cakurda, PhD.
Prof. Ing. Alexander Hosovsky, PhD.
Technical University of Kosice
Faculty of Manufacturing Technologies with the Seat in Presov
Bayerova 1, 080 01 Presov, Slovakia
+421 055 602 6420
e-mail: pavlo.pomin@tuke.sk, tomas.cakurda@tuke.sk, alexander.hosovsky@tuke.sk