Modelling the behavior of a soft material-actuator

Aslan Miriyev
aslan.miriyev@columbia.edu

Sam Cohen
slc2206@columbia.edu

Neil Chen
neil.chen@columbia.edu

***Abstract*— Self-contained electrically-driven soft actuators with high strain density, low cost, simple fabrication methods, and low current draw represent a key challenge in soft robotics. We present several methods for predictive analysis of the behavior of such an actuator, with a focus on time-series analysis of actuator load and wear.**

I. INTRODUCTION

A soft, robust, self-contained composite material-actuator exhibiting high actuation stress along with very high actuation strain resolves many of the issues confronted by traditional soft actuation solutions (FEAs, PAMs, DEAs, etc.). The complex electrical, chemical, and physical behavior of these actuators, however, presents multiple steady-state and transient response controls problems.

Here we approach one of these problems: we attempt to evaluate the load characteristic, and in turn the wear, of a single soft material-actuator sample over time.

We evaluate the performance of three separate approaches to modelling the behavior of this soft material-actuator:

1) Function approximation of actuator load cycle behavior via deep learning.
2) Regression analysis on derived features of actuator load cycle characteristic.
3) Time-series autoregression methods.

Ultimately we find that more input features and/or samples are necessary to usefully model the behavior of such an actuator.

II. DATA

A. Experimental Setup

The experimental setup and data generation process consists of a new (or rejuvenated) actuator sample enclosed in an Instron load cell. The actuator is sufficiently constrained such that it can neither extend nor contract.

During measurement, a cycle begins whenever `Load` reaches $0N$. At the peak of a cycle, a `Load` threshold of roughly $135N$ has been reached, and current is no longer supplied to the actuator. Thus cycle `Load` minima and maxima remain roughly fixed between cycles. This behavior is visible in **Figure II-B**.

B. Dataset

The dataset (`raw_cycles.csv`, which can be downloaded here) consists of 2 features and $170,103$ time-series samples measured at $0.100s$ intervals. Please preview the dataset online, or refer to **Table ??**, for more information.

The data represents 47 actuation *cycles*. A *cycle* consists of a *heating* phase and a cooling phase. During *heating*, the actuator exerts positive force, and thus `Load` is convex increasing with time. During *cooling*, the actuator gradually exerts less force, and thus `Load` is convex decreasing with time.

Feature	Unit	Format
`Time`	seconds	3 decimal places
`Load`	Newtons	5 decimal places

Table **??**. Unit and format of features in input dataset.

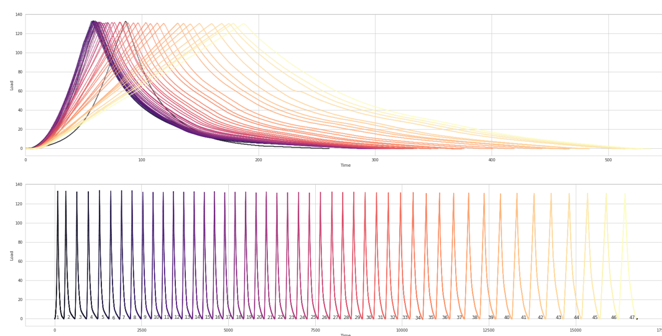

Fig. 1: Heating-cooling cycles over time.

Specified in this manner, the dataset consists of 47 complete cycles. Note that the local maximum and local minimum `Load` values of each cycle are roughly equal.

C. Exploratory Data Analysis

We begin by plotting and quantifying certain characteristics of the dataset:

1) Cycle period over cycle number: **Figure II-C**
2) Cycle heating time over cycle number: **Figure II-C**
3) Cycle cooling time over cycle number: **Figure II-C**

We find that the average cycle heating and cooling times are 91.29 seconds and 265.32 seconds, respectfully. We also note that cooling and heating durations appear to grow exponentially over the number of cycles. This trend is likely due to a variety of physical phenomena occurring within the actuator, such as gradual ethanol escape and internal heat build up.

III. DERIVED FEATURES

A. Motivation

We attempt to inject expert knowledge about the actuator system's behavior by constructing 11 derived features. These features encode approximately constant maximum/minimum-`Load` behavior, n^{th}-order moments of

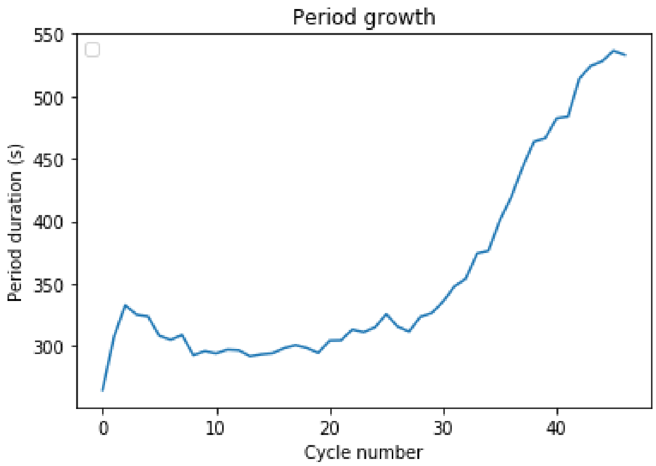

Fig. 2: **Figure II-C.** Period growth as a function of cycle number.

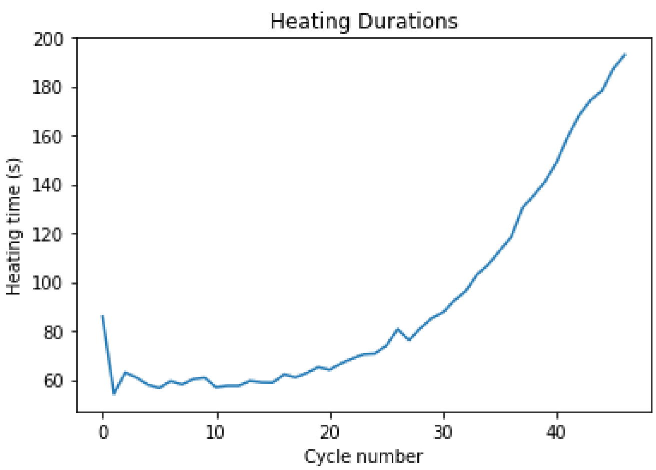

Fig. 3: **Figure II-C.** Heating duration as a function of cycle number.

each cycle, cycle index, and more. Each feature is manually computed over the entire dataset; derived features are excluded from validation/testing sets.

B. Computation

Table **??** enumerates all derived features.

IV. Learning Periodic Functions

We begin by fine-tuning a model to learn simple periodic functions. As our eventual goal is to predict the behavior of a non-periodic Time series, we opt to similarly treat this task as a Time series forecasting problem in which our samples are not independent, but are rather related to one another across time.

A. Formating Data for Supervised Learning

To build our set of examples, we first calculated the period P of a cycle.

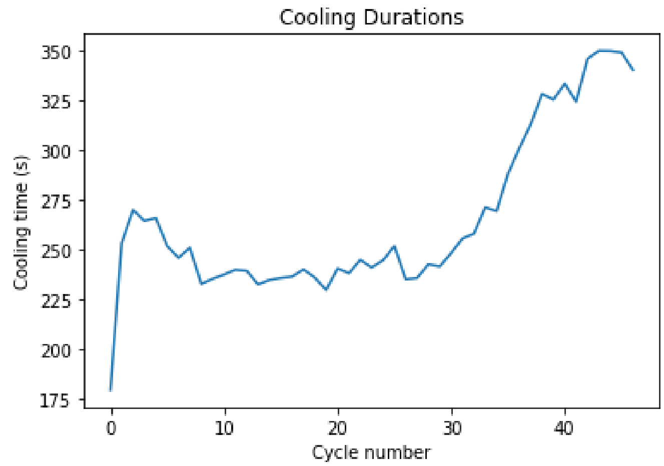

Fig. 4: **Figure II-C.** Cooling duration as a function of cycle number.

Feature	Meaning
Time	Absolute time since $0.00s$
Load	Absolute load from $0N$
Min	One-hot encoding of local minima
Max	One-hot encoding of local maxima
Cycle	Index of cycle from 0
Area	Area under cycle curve, calculated as $\int_{t_{min}}^{t_{max}} \texttt{Load}\ dt_i$
Heating	Time elapsed while heating, calculated as $t_{max} - t_{min}$
Cooling	Time elapsed while cooling, calculated as $t_{end} - t_{max}$
HCprop	Proportion $\frac{\texttt{Heating}}{\texttt{Cooling}}$ of heating time to cooling time
Period	Period of cycle, equivalent to $\texttt{Heating} + \texttt{Cooling}$
Tail	Time elapsed while $\texttt{Load} \leq (0.1 \times \texttt{Max Load})$
Belly	Multiple integral between $\texttt{Cooling}$ curve and line tangent to both $\texttt{Max Load}$ and $\texttt{Load}_{tend}$
Kurt	Kurtosis of curve (fourth standardized moment)
Skew	Skewness of curve

Table ??. All derived features and calculations involved.

A single example is formatted such that it contains three full cycles worth of *input* points (*Px3*) and one full cycle worth of *label* points (*P*).

We then utilize the "sliding window" method to step along our univariate time series one point at a time, collecting the proceeding *Px4* points to build an example. This process continues along the dataset until there aren't enough remaining points to produce a full *input* and *label* pairing.

The final result is stored in a Pandas Dataframe to be fed into the Neural Network.

B. Network Architecture

To best capture the time dependency of our data, we apply Long short-term memory (LSTM) neural networks to this task.

The network's architecture is as follows:

1) Input Layer (*Px3* data points)
2) LSTM Layer (120 memory gate neurons)
3) Dense Layer (100 neurons)
4) Output Layer (*P* predictions)

C. Prediction & Results

Below are our results on a sine wave, triangle wave, and square wave.

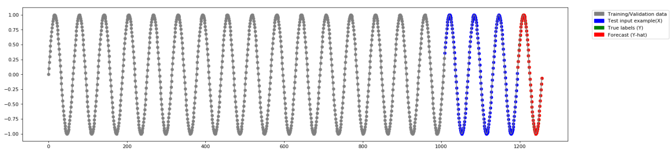

Fig. 5: **Figure IV-C.** Forecasting the sine wave with a LSTM network.

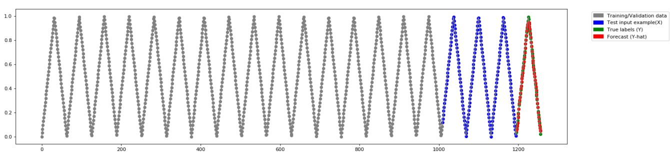

Fig. 6: **Figure IV-C.** Forecasting the Triangle wave with a LSTM network.

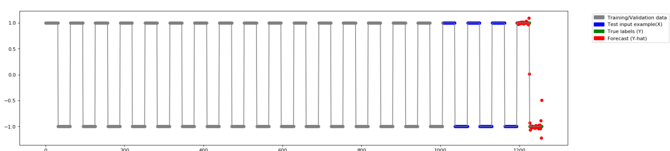

Fig. 7: **Figure IV-C.** Forecasting the Square wave with a LSTM network.

We quantify our prediction results by calculating the *Root Mean Squared Error* (RMSE) and the *Mean Absolute Error* (MAE) percentage for each wave forecast. To calculate the MAE %, we divide the MAE over all test data points by the difference between the cycle's peak and trough. By making the error relative to cycle height to produce a percentage, our results are more interpretable. **Table IV-C** contains these results across all three periodic functions. Note that each model trains for *1000 epochs* and uses a *150 batch size*.

Function type	MAE %	RMSE
`Sine`	0.269	0.006
`Triangle`	1.36	0.014
`Square`	5.34	0.193

TABLE I: **Table IV-C.** Periodic function forecasting results.

V. Learning Non-periodic Behavior

A. Data Formatting

We first modify the original data formatting phase to allow for a variable number of input and output points as cycle periods vary and require a variable number of points to capture.

To do so, we first perform a preprocessing step to calculate the period of each cycle based on it's peak and trough locations. We then use the same sliding window method that collects four cycles worth of points for each example. It's important to note that Keras LSTM networks must have a fixed size input and output, and so we store each example in a larger fixed-length array (a *container*). This length is preset and is always larger than an example worth of points, so as to not loose information. We represent *Null* values with -10. To help ensure that the network doesn't confuse these values with real data, we shift all data points until the smallest value is 0.

B. Network Architecture

Each layer in our network is much larger. Even after sub-sampling one data point for every fifty, the input data is far more dense than that of the periodic waves.

1) Input Layer (*Px3* data points)
2) LSTM Layer (500 memory gate neurons)
3) Dense Layer (300 neurons)
4) Dense Layer (150 neurons)
5) Output Layer (*P* predictions)

VI. Classical Approaches

A. Motivation

Simple linear/logistic regression methods, tuned properly, can outperform more complex models. We attempt to develop such methods with the goal of visualizing and predicting trends in actuator behavior.

B. Regression analysis

We use the following canonical approach to generate models and evaluate their performance on derived feature prediction:

1) Generate time-series cross-validation training-validation splits, delimiting data by each cycle.
2) Perform grid-search on regression parameters to identify models which best predict derived features

We then develop a regression model from these derived features:

1) Split cycles into four piecewise curves, as enumerated in VI-C.
2) Perform grid-search on regression parameters, minimizing loss (per piecewise curve) between predicted `Load` and ground truth `Load` as a function of `Time`.

C. Piecewise cycle splitting

We split curves into continuous piecewise sections using two simple methods: fixed boundaries and K-means clustering.

1) Fixed boundaries: We delimit curves at four points. Let $t_0 = 0$ denote the time at which a cycle begins; $t_{max} =$ the time at which a cycle reaches its maximum `Load`; and t_f denote the time at which a cycle finishes, such that $t_f =$ the `Period` of a cycle. Then:

$$t_1 = \frac{t_{max} - t_0}{2}$$
$$t_2 = t_{max}$$
$$t_3 = \frac{t_f - t_{max}}{2}$$

and we regress separately on the following subsections of a cycle's `Period`:

$$p_1 = t_1 - t_0$$
$$p_2 = t_2 - t_1$$
$$p_3 = t_3 - t_2$$
$$p_4 = t_f - t_3$$

2) K-means clustering: We delimit curves based on the clusterings generated by $K - means$ with $K = 2$. We run K-means, preserving no parameters, independently on the heating and cooling curves of each cycle. New (validation) points can be classified into cluster via the $1-$nearest-neighbors algorithm. K-means (as used here) computes the centroids of observations as

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

where $S_i =$ the set of observations x_j in cluster i. Since the load characteristic of the heating curve of each cycle is convex increasing, and we force $K = 2$, we can identify the elbow of each heating curve in an unsupervised manner. Similar reasoning applies to the cooling curve of each cycle. Then observations are once again clustered into one of p_1, p_2, p_3, p_4.

D. Piecewise regression performance

We identify a model which reports optimal validation accuracy via the following algorithm, using the definitions of t_i expressed in VI-C. r_i denotes a regression function using ridge (Tikhonov) regularization.

TRAIN(X, y)
$P_1, \ldots, P_4 = \{\}$
$\Gamma = [3.40, 10.0, 3.40, 5.60]$ **//** regularization penalties
$X = X \cup \bigcup_{i=2}^{5} i^{\text{th}}$-order features of X
for $c \in X$ **//** $c =$ a discrete cycle in X
    for $i = 1$ **to** 4
        $p_i = c[t_{i-1} : t_i]$ **//** $t_2 = t_{max}$ and $t_4 = t_f$
        $P_i = P_i \cup p_i$
for $i = 1$ **to** 4
    fit r_i on $(P_i, y[P_i])$, $\alpha_{r_i} = \Gamma_i$

Then new observations, with input features `Time` and `Cycle` (cycle index) only, map to predicted `Load` as follows. Let $X =$ the matrix of `Time` and `Cycle` observations for a single cycle.

PRED(X)
for $i = 1$ **to** 4
    predict $y_i = r_i(X)$
return $y_1 + \cdots + y_4$

E. Time-series analysis

F. Headings, etc

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named Heading 1, Heading 2, Heading 3, and Heading 4 are prescribed.

G. Figures and Tables

Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation Fig. 1, even at the beginning of a sentence.

TABLE II: An Example of a Table

One	Two
Three	Four

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an document, this method is somewhat more stable than directly inserting a picture.

Fig. 8: Inductance of oscillation winding on amorphous magnetic core versus DC bias magnetic field

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity Magnetization, or Magnetization, M, not just M. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write Magnetization (A/m) or Magnetization A[m(1)], not just A/m. Do not label axes with a ratio of quantities and units. For example, write Temperature (K), not Temperature/K.

VII. CONCLUSIONS

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

APPENDIX

Appendixes should appear before the acknowledgment.

ACKNOWLEDGMENT

The preferred spelling of the word acknowledgment in America is without an e after the g. Avoid the stilted expression, One of us (R. B. G.) thanks . . . Instead, try R. B. G. thanks. Put sponsor acknowledgments in the unnumbered footnote on the first page.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

REFERENCES

[1] G. O. Young, Synthetic structure of industrial plastics (Book style with paper title and editor), in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 1564.

[2] W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123135.

[3] H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.

[4] B. Smith, An approach to graphs of linear forms (Unpublished work style), unpublished.

[5] E. H. Miller, A note on reflector arrays (Periodical styleAccepted for publication), IEEE Trans. Antennas Propagat., to be publised.

[6] J. Wang, Fundamentals of erbium-doped fiber amplifiers arrays (Periodical styleSubmitted for publication), IEEE J. Quantum Electron., submitted for publication.

[7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style), IEEE Transl. J. Magn.Jpn., vol. 2, Aug. 1987, pp. 740741 [Dig. 9th Annu. Conf. Magnetics Japan, 1982, p. 301].

[9] M. Young, The Techincal Writers Handbook. Mill Valley, CA: University Science, 1989.

[10] J. U. Duncombe, Infrared navigationPart I: An assessment of feasibility (Periodical style), IEEE Trans. Electron Devices, vol. ED-11, pp. 3439, Jan. 1959.

[11] S. Chen, B. Mulgrew, and P. M. Grant, A clustering technique for digital communications channel equalization using radial basis function networks, IEEE Trans. Neural Networks, vol. 4, pp. 570578, July 1993.

[12] R. W. Lucky, Automatic equalization for digital communication, Bell Syst. Tech. J., vol. 44, no. 4, pp. 547588, Apr. 1965.

[13] S. P. Bingulac, On the compatibility of adaptive controllers (Published Conference Proceedings style), in Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory, New York, 1994, pp. 816.

[14] G. R. Faulhaber, Design of service systems with priority reservation, in Conf. Rec. 1995 IEEE Int. Conf. Communications, pp. 38.

[15] W. D. Doyle, Magnetization reversal in films with biaxial anisotropy, in 1987 Proc. INTERMAG Conf., pp. 2.2-12.2-6.

[16] G. W. Juette and L. E. Zeffanella, Radio noise currents n short sections on bundle conductors (Presented Conference Paper style), presented at the IEEE Summer power Meeting, Dallas, TX, June 2227, 1990, Paper 90 SM 690-0 PWRS.

[17] J. G. Kreifeldt, An analysis of surface-detected EMG as an amplitude-modulated noise, presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.

[18] J. Williams, Narrow-band analyzer (Thesis or Dissertation style), Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

[19] N. Kawasaki, Parametric study of thermal and chemical nonequilibrium nozzle flow, M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.

[20] J. P. Wilkinson, Nonlinear resonant circuit devices (Patent style), U.S. Patent 3 624 12, July 16, 1990.