15th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering
and
3rd Conference on Imaging and Visualization
CMBBE 2018
P. R. Fernandes and J. M. Tavares (Editors)

TOWARDS FINITE-ELEMENT SIMULATION USING DEEP LEARNING

Francois Roewer-Despres*, Najeeb Khan†, Ian Stavness‡

Department of Computer Science, University of Saskatchewan
*francois.roewerdespres@usask.ca, †najeeb.khan@usask.ca, ‡ian.stavness@usask.ca

Keywords: Deep Learning, Finite-Element Modeling, Autoencoder, Principal Component Analysis, Biomechanics Simulation, Dimensionality Reduction.

Abstract: *Finite-element modeling is commonly used to simulate soft-tissue biomechanics, but is too computationally burdensome for use in real-time applications. Various forms of dimensionality reduction have been investigated to reduce the computational cost of finite-element simulation, such as surrogate models, principal-component analysis, and model-order reduction, however linear dimensionality reduction techniques may be insufficient to capture the high degree of non-linearity in biological soft-tissue materials. Recent advances in deep learning have the potential to represent a highly complex and non-linear model deformation space in a compact form. In this paper, we use a deep-autoencoder to approximate the large deformations of a non-linear, muscle actuated beam. We found that the autoencoder consistently produced lower reconstruction error than the equivalently sized principal-component analysis model. These results are a preliminary step towards modeling more fulsome biomechanical soft-tissue models with deep learning approaches.*

1. INTRODUCTION

Finite-element (FE) modeling is the standard approach for representing soft-tissue structures in biomechanical simulations. FE models, however, incur a substantial computational cost due to having thousands of degrees-of-freedom. The slow simulation speeds of FE models make them difficult to use in a clinical context where quick, or even real-time, feedback of simulation results are necessary. Dimensionality reduction approaches, such as condensation [1], surrogate models [2, 3], model-order reduction [4, 5], and principal-component analysis (PCA) [6, 7] have been investigated in various biomechanical simulation contexts in order to speed up FE simulation times. Many of these dimensionality reduction approaches are intrinsically linear transformations. Large deformation, non-linear FE models, which are common for muscle tissue modeling such as tongue modeling [8], present a significant challenge for traditional FE dimensionality reduction.

Deep autoencoders (DAEs) have recently been applied to rigid bodies, but not FE models [9], yet DAEs have the potential to complement FE simulations by learning an approximation of the FE model's dynamics that can be computed in real-time. However, evaluating the accuracy of learned FE models is an essential first step for their use in biomechanical modeling. In this paper, we compare the reconstruction accuracy of DAEs of a non-linear FE beam model to that of traditional PCA [10]. To improve our analysis, we construct DAEs with either 3D or 6D latent spaces, with a matching number of principal components (PCs) in the PCA method. A fundamental assumption in machine learning is that models are trained using data drawn from the same distribution as the eventual use cases. However, because our DAEs are trained to approximate a specific FE model's dynamics, it is interesting to consider whether modifying the simulation inputs' distribution, while keeping model dynamics constant, greatly affects reconstruction accuracy, especially in comparison to PCA. To assess this, we evaluate our models using testing data drawn from either the same or a different distribution as the training data. Finally, we compare the dimensionality reduction capabilities of the DAEs to that of the PCA models.

2. METHODS

FE Model We implemented a simple FE beam model in ArtiSynth, an open-source biomechanical simulation toolkit [11]. The beam consisted of 16 hexahedral elements (45 nodes) arranged in a rectangular grid with dimensions of $0.04 \times 0.04 \times 0.10$ meters (see Figure 1). The 9 nodes on the rightmost face of the beam were set non-dynamic as a fixed boundary condition. The 3D position of the remaining 36 nodes represented the FE model state as a 108 dimensional vector (36×3). Deformations of the beam were effected by muscle elements, which generate compressive stress when activated. Superior longitudinal (SL), inferior longitudinal (IL), vertical (V), and transverse (T) muscle elements were added along the top, bottom, vertical, and horizontal element edges, respectively.

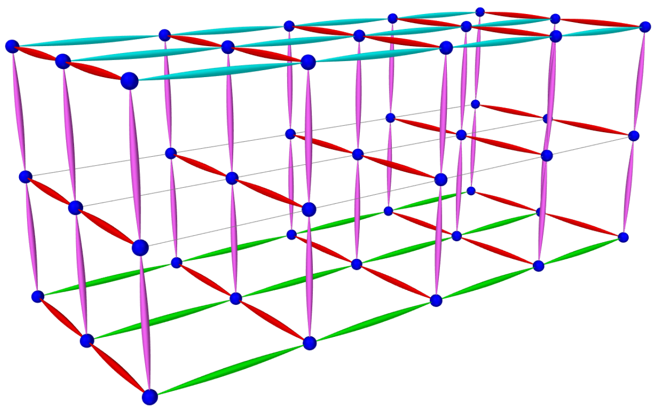

Figure 1. Simple FE beam model, with nodes in blue and SL, IL, V, and T muscle elements in cyan, green, purple, and red, respectively.

Data Gathering. Training and testing data were generated using the BatchSim feature of ArtiSynth, which permits flexible and autonomous execution of simulations with combinatorial or probabilistic input parameter (SL, IL, V, and T muscle activations) variation, and produces structured outputs (FE model state) that can be directly used with deep learning and PCA.

We assumed that muscle activations are coordinated with independently controllable synergies [12, 13]. For the beam, these synergies were the four muscle groups built into the model, but for future anatomical models, the synergies could be derived from data [14, 15]. Training data were generated to create sparse activation patterns for the muscle synergies, i.e. to reduce co-contraction of antagonist synergies in order to generate large beam deformations. For all possible combinations of 1 or 2 of the 4 muscles, we simulated all possible combinations of activations from 0% to 100% (in increments of 10%). To introduce noise, each simulation was repeated multiple times, with the activation of all remaining muscles being drawn from independent uniform distributions over [0%, 5%] in each repetition. In total, 11750 samples were drawn. For each sample, ArtiSynth's quasi-static solver computed the final deformation of the beam that results from the muscle activations, and BatchSim recorded the resulting model state (see Figure 2). In addition, for each sample, we categorized its deformation class (Neutral, Bend Up, Bend Down, Elongate, or Contract) and computed its deformation strength (e.g. slightly elongated vs. very elongated) relative to the neutral position of the central node of the front face (larger values indicate greater strength). These were recorded for investigating dimensionality reduction capabilities.

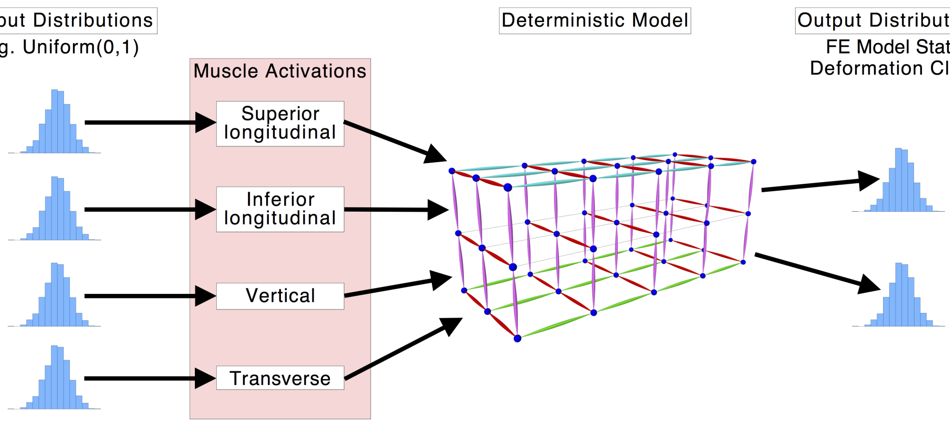

Figure 2. Data gathering procedure using the BatchSim feature of ArtiSynth.

Testing data were generated similarly, but split into two sets. Muscle activations were drawn, in the first (Test Set A), from the same distribution as that of the training data (2938 samples), and in the second (Test Set B), from independent uniform distributions over [0%, 100%] (3000 samples).

Deep Autoencoders. We constructed feedforward DAEs with fully-connected layers consisting of Parametric Rectified Linear Units (P-ReLU) [16] (see Figure 3). These differ from regular ReLU in that the activation function

$$\phi(x) = \max(\gamma x, x) \tag{1}$$

includes a learnable parameter γ, whereas $\gamma \equiv 0$ in ReLU. γ was initialized to 0 for all layers except the output layer, where it was initialized to 1. As such, all layers except the last were initially ReLU, whereas the last layer initially used a linear activation function, which is common practice when using DAEs for regression [17]. The architecture in terms of number of units per layer was as follows: 108, 64, 32, 16, X, 16, 32, 64, 108, where $X \in \{3, 6\}$ is the number of units in the code layer, representing the dimension of the latent space.

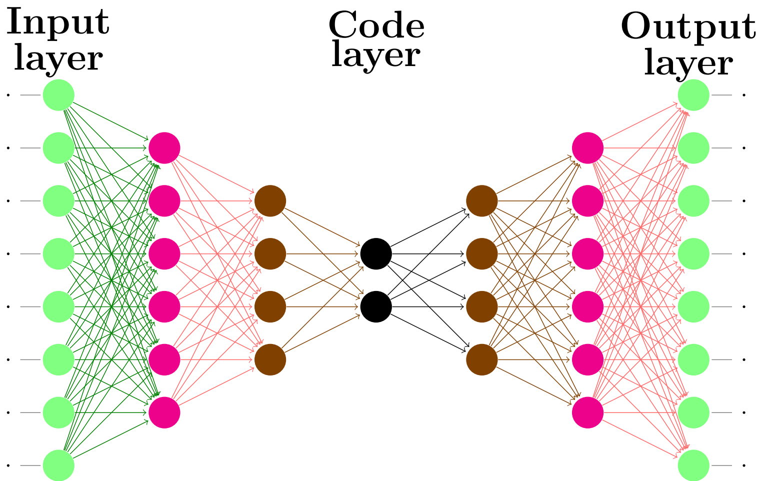

Figure 3. General architecture of a deep autoencoder.

Regularization is common practice in artificial neural networks to prevent overfitting to training data and improve generalizability to unseen examples, but at the cost of some loss in precision. However, since we were training DAEs to capture the deformation behaviour of a specific object, if the training data sufficiently cover the space of possible deformations, then overfitting would be minimal, and regularization, unnecessary. If overfitting occurred, validation loss should be significantly larger than training loss. In addition, Test Set B was designed to assess the generalizability of the model to examples drawn from a distribution different than the training set.

Each DAE was trained for 1000 epochs in mini-batches of size 128 using the mean squared error (MSE) loss function and the Adam optimizer with

$$\begin{aligned} \alpha &= 0.0001 \\ \beta_1 &= 0.9 \\ \beta_2 &= 0.999 \\ \epsilon &= 1\text{e}{-8} \end{aligned} \tag{2}$$

and α decay rate of 0.000001 [18]. The training data were split into training and validation sets in a 90-10 ratio. All data were normalized using the training set mean and standard deviation.

Principal Component Analysis. PCA can be interpreted as a machine learning model. If only the k most important PCs are kept, PCA is equivalent to a DAE with k units in the code layer and using only linear activation functions [19]. As such, DAEs are a non-linear generalization of PCA, and the space of the k most important PCs can be considered the k-dimensional latent space of a PCA model. To match our DAEs, we used $k \in \{3, 6\}$. The "training" of a PCA model consists in factoring the training data matrix to obtain the k most important PCs.

Reconstruction Accuracy Measurement. We quantified reconstruction accuracy using MSE. Each sample in a test set was fed as input to a DAE or PCA model, which attempted to reconstruct the input as its output. Given that the input and output vectors corresponded to the 3D position of the FE beam nodes, computing the MSE between them amounts to computing the average squared distance between each node's true and reconstructed position. Given that the 3D positions were in units of meters, the MSE was in units of square meters.

Dimensionality Reduction. PCA is used as a dimensionality reduction technique whereby the input vector is mapped into the latent space [19]. DAEs can serve the same purpose by interpreting the output of the code layer units as a latent space vector. For both DAE and PCA models, we mapped each input vector of a test set into the latent space. To visualize the structure of the latent space, we plotted the point corresponding to each latent space vector, then coloured and sized each point according to the deformation class and strength, respectively, of the corresponding input vector. Since 6D latent spaces cannot be directly visualized, we embedded the 6D vectors in 3D space using the t-SNE method, which maintains the relative distance of the 6D points in the 3D space [20]. This is important, as qualitatively determining whether or not dimensionality reduction preserves the beam's structural information, relies upon observing that deformation classes cluster together and that deformation strengths vary smoothly within each cluster.

3. RESULTS

Figure 4 shows the loss as a function of epoch for all trained DAEs. The near equivalence between training and validation losses indicates overfitting is minimal.

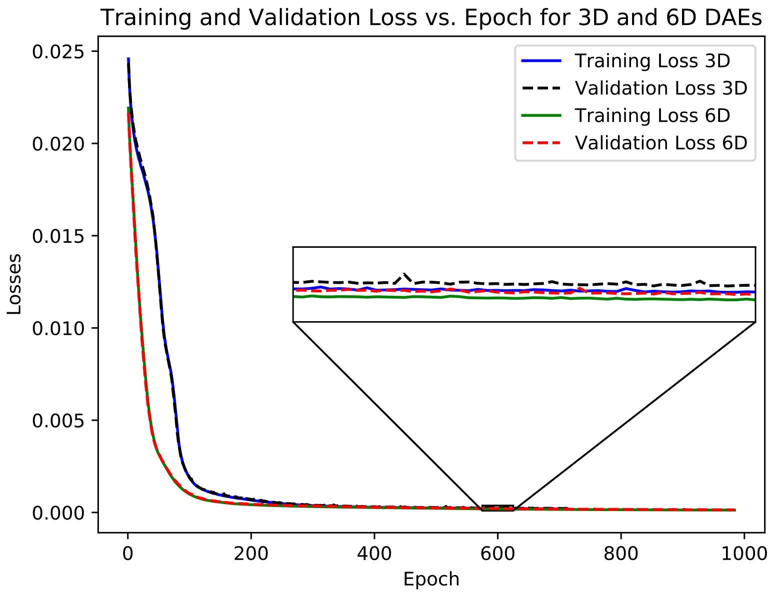

Figure 4. Training and validation loss vs. epoch for DAEs with a 3D or 6D latent space.

An example reconstruction of the beam model deformation by DAE and PCA is shown in

Figure 5, where the true node positions are in blue, DAE-reconstructed node positions in red, and PCA-reconstructed node positions in green. Qualitatively, the DAE reconstruction appears to better fit the true deformation as compared to the PCA reconstruction.

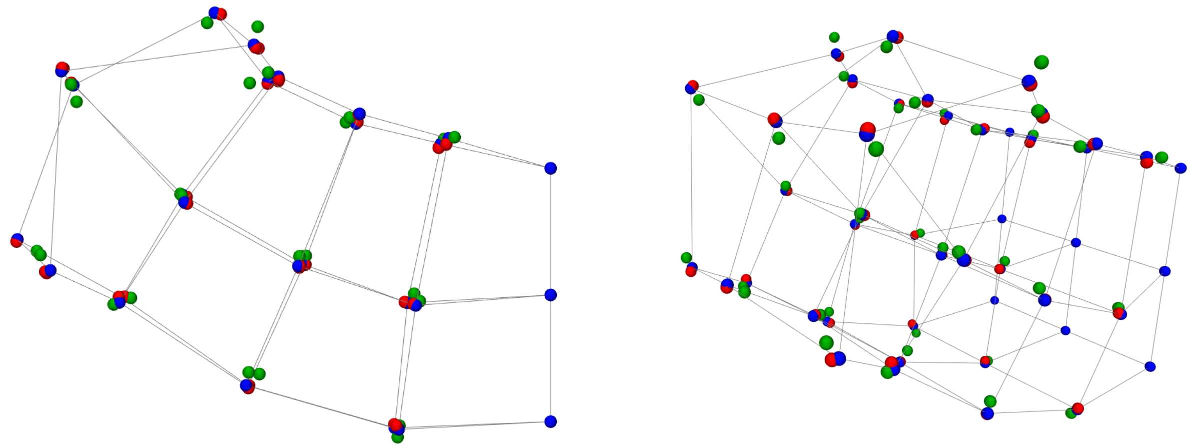

Figure 5. Side view (left) and oblique view (right) of a reconstruction example, where IL and SL have 50% and 100% activation, respectively, showing DAE (red points) and PCA (green points) reconstructions with 3D latent spaces.

Table 1 summarizes the testing results for both the DAE and PCA models, using both test sets, and with both 3D and 6D latent spaces. The entries are the mean accuracy (MSE) for each case, with the accuracy standard deviation in parentheses.

Test Set	Model	Latent Space Dimension 3	Latent Space Dimension 6
A	DAE	1.175e−07[a] (2.587e−07)	7.912e−08[b] (1.776e−07)
A	PCA	4.341e−06[c] (3.667e−06)	4.590e−07[d] (3.713e−07)
B	DAE	1.588e−06[e] (1.155e−06)	1.071e−06[f] (8.058e−07)
B	PCA	5.003e−06[g] (3.210e−06)	1.274e−06[h] (7.671e−07)

Table 1. Reconstruction accuracy results for all configurations.

Table 2 summarizes hypothesis test results. Each test is a two-tailed, two-sample t-test (paired whenever possible), with the null hypothesis that the two population means (of which the sample means in Table 1 are estimates) involved in the test are equal. The "model" tests examine whether DAEs outperform PCA models, with the null hypothesis that the mean accuracy of DAE and PCA models with equivalent test set and latent space dimension are equal. The "dimension" tests examine whether 6D latent spaces outperform their 3D counterpart, with the null hypothesis that the mean accuracy of two models of the same type (DAE or PCA) and

Test Type	Null Hypothesis	Sample Means Involved	p-value
Model	Model choice has no effect on mean accuracy with 3D latent space and using Test Set A	a, c	$p < 0.00001$
Model	Model choice has no effect on mean accuracy with 3D latent space and using Test Set B	e, g	$p < 0.00001$
Model	Model choice has no effect on mean accuracy with 6D latent space and using Test Set A	b, d	$p < 0.00001$
Model	Model choice has no effect on mean accuracy with 6D latent space and using Test Set B	f, h	$p < 0.00001$
Dimension	Latent space dimension has no effect on mean accuracy of DAEs on Test Set A	a, b	$p < 0.00001$
Dimension	Latent space dimension has no effect on mean accuracy of DAEs on Test Set B	e, f	$p < 0.00001$
Dimension	Latent space dimension has no effect on mean accuracy of PCA on Test Set A	c, d	$p < 0.00001$
Dimension	Latent space dimension has no effect on mean accuracy of PCA on Test Set B	g, h	$p < 0.00001$
Test Set	Test set has no effect on mean accuracy of DAEs with 3D latent space	a, e	$p < 0.00001$
Test Set	Latent space dimension has no effect on mean accuracy of DAEs with 6D latent space	b, f	$p < 0.00001$
Test Set	Latent space dimension has no effect on mean accuracy of PCA with 3D latent space	c, g	$p < 0.00001$
Test Set	Latent space dimension has no effect on mean accuracy of PCA with 6D latent space	d, h	$p < 0.00001$

Table 2. Hypothesis test results. Sample means refer to entries in Table 1.

equivalent test set, but differing latent space dimension are equal. The "test set" tests examine whether Test Set A outperforms Test Set B, with the null hypothesis that the mean accuracy of two models of the same type (DAE or PCA) and equivalent latent space dimension, but differing test set are equal. We see that, in all cases, the null hypothesis is rejected in favour of the alternative hypothesis that the accuracy means are not equal. Comparing mean values (see Table 1), DAEs consistently outperform PCA models, 6D consistently outperform 3D latent spaces, and Test Set A consistently outperforms Test Set B.

The latent space visualizations for DAE and PCA models were qualitatively similar (see Figure 6). In all cases, the clusters are well defined, with few outliers. In addition, deformation strengths vary smoothly within each cluster. The t-SNE embedding of 6D latent spaces (Figure 6, right column) resulted in occasional bimodal clustering observed within each deformation class, though these may, in fact, form a contiguous cluster in the higher-dimensional space.

4. CONCLUSIONS

Finite-element (FE) models face a number of computational challenges compared to multibody musculoskeletal models. DAEs have the potential to compute an approximation of FE model dynamics in real-time, while providing greater accuracy than PCA. Evaluating the accu-

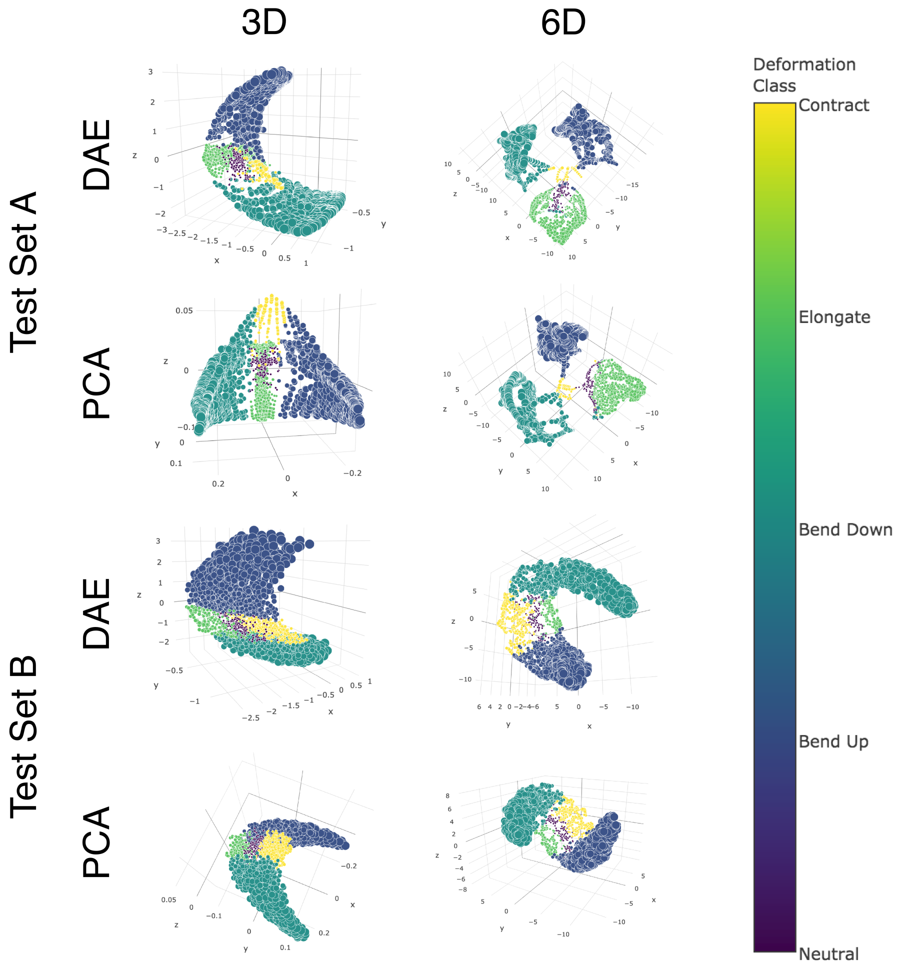

Figure 6. Latent space visualizations for all configurations.

racy of these models represents a critical first step.

Results show DAEs outperform equivalent PCA models. Although the difference is statistically significant because of the large sample sizes, the result may not always be practically significant. However, an order of magnitude better performance is observed in the case of Test Set A with 3D latent spaces. This fits with the qualitative assessment that the DAE-reconstructed red points fit better than the PCA-reconstructed green points. Comparing across latent space dimensions, results show 6D latent spaces outperform their 3D counterpart. This agrees with intuition, since doubling the number of dimensions allows the models to capture more variable information. Similarly, comparing across test sets, we see that Test Set A outperforms Test Set B. This was expected, since samples drawn from the same distribution as the training data should be reconstructed more accurately. However, the DAEs, despite having no regularization, do not drastically overfit the training data, since they still outperform equivalent PCA models across test sets.

Although DAE and PCA models structure their latent spaces differently (linearly vs. nonlinearly), neither appears superior to the other at this stage. As future work, we plan to extend this analysis to larger FE models that involve more substantial non-linearity (e.g. FE models with contact, and a 3D FE tongue model) to investigate whether the non-linear dimensionality reduction of the DAEs produces a more effective low-dimensional representation than PCA. If so, the encoders and decoders resulting from the trained DAEs may serve as effective pre-trained layers to separately learn predictive forward (muscle activation to FE state) and inverse (FE state to muscle activation) models.

As future work, we plan to assess the computational cost and numerical stability of both DAE and PCA models compared to Artisynth's quasi-static incremental solver. We also plan to test additional measures of accuracy besides MSE and alternative loss functions for training the DAEs. We are particularly interested in assessing how these factors change as the complexity and size of the FE model increases.

5. REFERENCES

[1] Simon P DiMaio and Septimiu E Salcudean. Interactive simulation of needle insertion models. *IEEE transactions on biomedical engineering*, 52(7):1167–1179, 2005.

[2] Ilan Eskinazi and Benjamin J Fregly. An open-source toolbox for surrogate modeling of joint contact mechanics. *IEEE Transactions on Biomedical Engineering*, 63(2):269–277, 2016.

[3] Jason P Halloran, Ahmet Erdemir, and Antonie J Van Den Bogert. Adaptive surrogate modeling for efficient coupling of musculoskeletal control and tissue deformation models. *Journal of biomechanical engineering*, 131(1):011014, 2009.

[4] Jernej Barbič and Doug L James. Real-time subspace integration for st. venant-kirchhoff deformable models. *ACM transactions on graphics (TOG)*, 24(3):982–990, 2005.

[5] Steven S An, Theodore Kim, and Doug L James. Optimizing cubature for efficient integration of subspace deformations. *ACM transactions on graphics (TOG)*, 27(5):165, 2008.

[6] Takaaki Kuratate, Hani Yehia, and Eric Vatikiotis-Bateson. Kinematics-based synthesis of realistic talking faces. In *AVSP'98 International Conference on Auditory-Visual Speech Processing*, 1998.

[7] Pierre Badin, Gerard Bailly, Lionel Reveret, Monica Baciu, Christoph Segebarth, and Christophe Savariaux. Three-dimensional linear articulatory modeling of tongue, lips and face, based on mri and video images. *Journal of Phonetics*, 30(3):533–553, 2002.

[8] Ian Stavness, John E Lloyd, and Sidney Fels. Automatic prediction of tongue muscle activations using a finite element model. *Journal of biomechanics*, 45(16):2841–2848, 2012.

[9] Najeeb Khan and Ian Stavness. Prediction of muscle activations for reaching movements using deep neural networks. In *The American Society of Biomechanics (ASB)*, 2017.

[10] Alex Pentland and John Williams. Good vibrations: Modal dynamics for graphics and animation. *ACM*, 23(3), 1989.

[11] John E Lloyd, Ian Stavness, and Sidney Fels. Artisynth: A fast interactive biomechanical modeling toolkit combining multibody and finite element simulation. In *Soft tissue biomechanical modeling for computer assisted surgery*, pages 355–394. Springer, 2012.

[12] Emilio Bizzi and Vincent CK Cheung. The neural origin of muscle synergies. *Frontiers in computational neuroscience*, 7:51, 2013.

[13] Bryan Gick and Ian Stavness. Modularizing speech. *Frontiers in psychology*, 4:977, 2013.

[14] Lena H Ting and Jane M Macpherson. A limited set of muscle synergies for force control during a postural task. *Journal of neurophysiology*, 93(1):609–613, 2005.

[15] Jonghye Woo, Maureen Stone, Fangxu Xing, Jordan Green, Arnold Gomez, Van Wedeen, Jerry L Prince, and Georges El Fakhri. Discovering functional units of the human tongue during speech from cine-and tagged-mri. *The Journal of the Acoustical Society of America*, 141(5):3647–3647, 2017.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[17] Alan Julian Izenman. *Modern Multivariate Statistical Techniques: Regression, Classification and Manifold Learning*. Springer, 2008.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[20] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.