Real-time simulation of arm and hand dynamics using ANN

Mykhailo Manukian[1[0000−0003−0752−3712]] and Sergiy Yakovenko[2[0000−0002−5946−6409]]

[1] Ukrainian Catholic University, Ilariona Svjentsits'koho Street, 17, Lviv, 79000, Ukraine
manukian@ucu.edu.ua
[2] West Virginia University, Morgantown, WV 26506, United States
seyakovenko@mix.wvu.edu

Abstract. The physics of body dynamics is a complex problem solved by the nervous system in real-time during the planning and execution of movements. The human hand has 27 degrees of freedom (DOF), and the arm has 4 DOF for elbow and shoulder joints. Due to the complexity of hand structure and functions, we need a complex biological "computer" in our head to control it. Neuroprosthetics require similar computations for neural decoding and sensory feedback tasks. Furthermore, since physical simulations are computationally complex, this research aims to approximate them using machine learning methods like ANN. For such a type of task, the most suitable network architecture is RNN or Transformer, which considers arm and hand motion's temporal dynamics. This study will validate different RNN - shallow recurrent ANN, LSTM - and Transformers architectures as candidates for the hand and arm motion control model. The input data for ANN is joint angles for an inverse-dynamic problem or joint torques for a forward-dynamic problem. Physical model of arm and hand, which operate in the MATLAB Simulink environment, will provide data to train our ANN. Lastly, the resulted model should work in real-time and have a latency of less than 4 ms to interpret torques into limb position coordinates to allow further usage of such a model in real-life applications.

Keywords: joint torques · RNN · musculoskeletal model · hand dynamics · arm dynamics · locomotion · motion control · Simulink.

1 Introduction

Today, arm and hand dynamics simulation from control signals are not yet fully solved. There are multiple attempts to solve it, but none of them achieved complete success yet. The human hand has 27 degrees of freedom (DOF): 4 in each finger, 3 for extension and flexion, and one for abduction and adduction; the thumb is more complicated and has 5 DOF, leaving 6 DOF for the rotation and translation of the wrist [1]. In addition to this, the arm itself also has 4 DOF for elbow and shoulder joints. Due to the complexity of hand structure and functions, we need a complex biological "computer" in our head to control it. As a result, the most significant part of our motor cortex responsible for hand motion control. So, solving a task of controlling arm and hand dynamics from control signals to the limb's exact position in space amongst the most challenging tasks in human locomotion simulation.

Many pieces of research focus on pattern recognition of EMG signals to classify a limited amount of gestures. Usually, the number of gestures are not high and lies within interval 4-12 gestures. Larger data sets are rare. However, despite promising results with high accuracy reported, pattern recognition usage in real-life applications could be complicated due to the small number of gestures. When the number of gestures increases, the accuracy of pattern recognition decrease as highlighted in [2].

The mapping of control signals recorded from cortex, nerves, or muscles during contractions and relaxations to precise limb position in space is a non-linear one. Considering this, any model which tries to map control signal to arm and hand position in space directly needs to solve a very complicated relationship. Thus, it could not be solved as a time series problem and approached as a pattern recognition problem with a limited number of gestures. As a result, the practical application of such solutions is limited also. The required model should solve two parts of the system - musculoskeletal transformation (MT) and limb dynamics (LD). Existing ML approaches attempt to solve both MT and LD parts altogether - without breaking them apart. Instead, our idea is to use the approach described in [11]. In this research, the musculoskeletal transformation was solved already. Hence we will focus only on the second part - find a model based on ANN to approximate limb dynamics part and find a mapping to arm and hand position in space. At time step $t + 1$ position of the limb in space will depend on two factors: distance covered within a time step $t + 1$ and the limb's initial position at time step t. The proposed model should consider such a temporal relationship where the final result on step $t + 1$ also depends on the result of step t.

In the proposed research scope, it is also essential to have some baseline model used during ANN training. One possible option here is to use the musculoskeletal model described in [3] to provide baseline values for network training. Finally, to allow such a model application in some real-life scenarios, there is also an important constraint. The resulting model should work in real-time and have a latency of less than 4 ms for the forward propagation.

Solving the outlined problem will lay a foundation for future prosthetic limb improvements and enhance its range of supported movements. It, in turn, will improve the quality and terms of amputee rehabilitation. Also, high performance and minimal response time of prosthetics in daily activities will improve amputees' quality of life by reducing their disability.

This paper includes the following:

- review of related work in the area of continuous locomotion control of upper limb;

– gap analysis between the proposed research and reviewed ones;
– refined definition of the problem and proposed solution approach;
– current results and project plan;
– summary and future work details.

2 Related Work

2.1 Literature Search Method

For this research, we used constrained Snowball sampling and citation network analysis described in [5]. According to [9] citation analysis is conducted on a sample of publications, often collected from Google Scholar or Microsoft Academic or other similar sites with citation relationships between documents. Usually, literature reviews done via keyword searches, which produce a set of related publications. It is also possible to include publications that cite the seed publications in the keyword set.

Snowball Sampling technique differs concerning the collection process from usual techniques, which uses keyword search. Here a citation network is built through a snowball sampling technique that starts with seed articles as explained in [9]. In this method, on the first level, the algorithm collect articles that cite the seed article. At the second level, articles that cite the articles from the first level are collected, and so on. As a result, a network of relevant articles is built around the seed and gives a broad view of the research topic, while keyword search usually gives a narrow list of possible candidate papers. There is a possibility of setting a required number of data collection levels and sampling rate (the proportion of articles collected at each level). The further away the article location is from the seed article, the less likely it is related to the original topic. By default, three levels of data collection are generally enough. The sampling rate allows keeping the data collection at a manageable level. Since the snowball sampling method used here, the list of candidate papers could grow huge, literally like "a snowball rolling down the hill." For example, setting the sampling rate to 50% of the research papers means that the result set will be $\frac{1}{2}$ the size at level 1, $\frac{1}{4}$ the size at level 2, $\frac{1}{8}$ at level 3. Setting a sample rate to 10 or 20 percent of the data makes the final data set manageable.

We have used Python scripts for controlled snowboard sampling to create a collection of relevant papers for this research topic. Mentioned scripts are provided by author of paper [6] on GitHub. As a result of script execution, we got a list of 281 publications collected. We reviewed this list, and none of the selected papers was relevant to the research topic. It could happen because the initial list of seeds is already representative enough and relevant to the given research topic. Hence we review all papers in the seeds list and choose five of them that most relevant based on described data, environment, research goals, and applied solution.

2.2 Review of Related Work

Identification of the Human Arm Kinetics using Dynamic Recurrent Neural Networks [7] This work reviews relationship between the electromyographic (EMG) signals and the movement of the skeletal system. EMG signals reflect the command signals sent by the central nervous system to the muscles. Many researchers used several techniques to solve this problem. Among those are the theory of optimization, mathematical high-order functions, and statistical correlation between EMG and limb movements. However, such techniques require EMG signal approximations or provide poor simulation results. The authors propose an alternative approach in the paper based on a dynamic recurrent neural network (DRNN). DRNN will approximate and map EMG signals to limb position. This paper aims to identify the relationship between muscle activity and upper-limb dynamics when a subject draws complex movements. Data for the research was recorded when the right-handed subject was drawing a figure "eight" with the extended right arm as fast as possible. Feedforward networks do not fit this task well since those do not include temporal relationships. DRNN structure consists of an array of several neurons and interconnections between all elements. There are two types of parameters: the classic weights between the units and the time constants associated with each artificial neuron. The time constants are required to increase the dynamical features of the model. Such a network will treat temporal sequences, and learning equations will be continuous in time, and time will appear explicitly.

Deep Learning with Convolutional Neural Networks Applied to Electromyography Data: A Resource for the Classification of Movements for Prosthetic Hands [2] The authors claim that despite the availability of various advanced myoelectric prosthetic hands, the control methods are still falling behind and rely in most cases on specific movement triggers or sequential control strategies. Results of proportional, natural, and dexterous control methods are still not robust enough to be translated into real life. The majority of the methods use surface electromyographic (EMG) signals map pattern recognition or proportional control algorithms. Those methods usually show accuracy higher than 90-95% on less than ten classes (gestures), while average accuracy usually below 80-90%. Deep learning and neural networks recently revolutionized such fields of machine learning as speech recognition and computer vision. Thus, it seems reasonable to investigate its abilities in EMG control methods as well. In this article, the authors apply convolutional neural networks to the classification of 50 hand movements in 67 intact subjects and 11 transradial hand amputees and compare the results with those obtained with classical machine learning methods on three Ninapro datasets. The convolutional neural network consists of a modified version of a well known CNN LeNet. Authors choose a simple network architecture to accelerate the training phase and evaluate the effects of several pre-processing, architectural, and optimization parameters. More complex network architectures were avoided on purpose by the authors. To increase overall accuracy, the authors performed data augmentation by doubling the amount of data and adding white Gaussian noise. To accommodate time dependency sliding time window was used for input data.

Myoelectric control of a computer animated hand: A new concept based on the combined use of a tree-structured artificial neural network and a data glove [10] In the article, the authors increase the number of EMG channels to read input data. As a result, this increases the dimensionality of the problem, and, to cope with it, the authors propose to use tree-structured ANN for EMG pattern recognition. The authors propose using a data glove located on a healthy hand contralateral to the amputated limb as baseline data. Also, using tree-structured, self-organized neural networks, the authors were able to achieve faster training cycles. Their manage to train the tree-structured network in about 30 seconds, which is 3-4 times faster than usual NN architecture. ANN training took place offline. Using the mentioned network, the authors demonstrated a successful 18 DOF control of an artificial hand (computer-animated model). The authors did not provide any technical details about the implemented tree-structured ANN.

A Human–Machine Interface Using Electrical Impedance Tomography for Hand Prosthesis Control [12] This paper proposes a different human-machine interface (HMI) approach to the EMG one. Instead, they propose to use bio-impedance imaging technology or electrical impedance tomography (EIT), which will react to the physical movement of muscles and bones in the forearm. During experiments, 11 gestures were recorded, and later neural networks were used to classify those gestures into respective categories. ANN was a feed-forward network with 10 or 100 neurons in the hidden layer. Due to the nature of the reading method, NN requires initial training each time the wristband is worn.

A Five-fingered Underactuated Prosthetic Hand Control Scheme [13] This paper proposes recognizing EMG motion patterns using the Autoregressive (AR) model, wavelet transforms, and Variable Learning Rate (VLR) based neural network. VLR based NN is applied to discriminate EMG motion patterns. VLR NN is feed-forward NN with three layers and no more than 30 neurons in the hidden layer determined via experiments. During experiments, a higher number of neurons in the hidden layer decrease NN recognition ability. Using VLR NN, authors could achieve fast learning speed, even for several samples of each motion. Force information extracted from the EMG signal control the driving speed of the particular finger. NN could predict one finger's movement at the given moment in time. Hence, a complex action consisting of multiple fingers' movement at once needs to be performed in a sequence (not continuously). The research's final goal was to control different fingers of the prosthetic hand based on EMG signals.

2.3 Gap analysis

The main difference between [7,2,10,13] and proposed research is that mapping from EMG signals to joint angles, velocities, and accelerations is considered a solved task in the scope of proposed research [11]. Instead, the proposed research aims at finding a mapping between joint dynamics and joint torques of all parts of the simulated hand and arm. While in [13] input data is EIT data, which is entirely different from EMG signals in nature.

The goal set in [7] is to draw an "eight" figure based on recorded EMG signals which does nothing with hand movement itself. Only arm dynamics down to the wrist was considered, and the most complicated part - hand - was left out of the scope of this research. However, there are many similarities between the research described in [7] with the proposed research. Speaking about [2], the research aims to classify 50 different gestures using CNN and compare results with classical classification techniques. A similar goal also set in research [12] - to correctly recognize and classify 11 gestures split into three groups, making real-life usage limited to the list of available gestures. In [10] goal is to map EMG signals with the precise position of the simulated hand, which should match to data collected from data-glove on a healthy hand. In the proposed research, we have a similar goal - to match joint angles, considering joint velocities and accelerations, to precise joint torques, which should match the result produced by the musculoskeletal model used as the baseline and described in [3]. Furthermore, in [13] goal of the research is to control different fingers based on EMG signals of the prosthetic hand. Nevertheless, even if each goal by itself is an important milestone and outstanding achievement, none of them allow controlling artificial limb in every-day life in real-time with similar precision and accuracy to a real one.

Table 1. Summary of reviewed research papers together with proposed research

Research	**Input Data**	**NN Architecture**	**Baseline Model or Method**	**Goal**
Proposed research	Joint angles, velocity and acceleration determined by [11]	RNN or Transformer inspired by NLP domain	Musculoskeletal model in SimuLink	Solve inverse dynamic problem and find joint torques in real-time
[7]	EMG signals	Dynamic RNN	Data points in space collected from subject	Approximate figure "8" drawn by straight hand
[2]	EMG signals	CNN	Three dataset of average 50 gestures	Classify 50 gestures and compare with classical classification methods
[10]	EMG signals	Tree-structured ANN	Data collected with data glove on healthy hand	To match position of simulated hand with positions from data glove
[12]	EIT data	Feed-forward shallow ANN	Dataset of 11 labeled gestures	To create HMI for hand prosthesis control able to recognize 11 gestures
[13]	EMG signals	VLR shallow NN	Six motions of thumb, index and middle fingers (2 per each finger)	Control different fingers and reproduce complex actions

In [2] CNN architecture was used with a sliding window to work effectively with time-series data. However, we propose to use RNN architecture instead because recurrent NN is naturally built for sequenced data, as was confirmed in [7]. Authors in [10] use tree-structured ANN, their colleagues in [12] use shallow NN architecture, which is enough to achieve good classification results, and in [13] authors use another shallow NN with variable LR. Despite being useful in pattern recognition of a limited list of gestures, such networks would not successfully learn the high-dimensional non-linear mapping between joint angles, velocity, acceleration, and joint torques to determine limb position in space. Since the publishing of [7] RNN algorithms were significantly improved and achieved much better results in such areas as NLP. It makes sense to apply more advanced RNN architectures like LSTM to solve hand dynamics approximation challenges. As an alternative to RNN, we propose considering the transformer model based on its recent advancements in the NLP domain.

Also, in [10] usage of data-glove has a critical limitation - such a method could not measure an amount of force applied. It could be relevant for the "fist" gesture so that the baseline value could be misleading in such situations. The proposed research musculoskeletal model [3] will provide a more precise baseline so that the model could react accordingly to various force applications. All reviewed publications summarized in Table 1 for easy consumption.

3 Research Setting and Approach to Solution

3.1 Research Structure

Based on the reviewed literature, we narrow down the scope of the proposed research. The main proposal is to design suitable recurrent neural networks (RNN) or Transformers models designed to work with sequential data in Natural Language Processing (NLP) domain. Next, we suggest applying the resulted model to find a robust mapping between joint dynamics and torques to identify a simulated limb's precise positions in space. Such RNN or Transformer based model, in addition to joint dynamics-torques mapping described above, should be able to learn temporal dependency between resulted trajectory on each time step from a previous one.

Baseline musculoskeletal model described in [3] implemented in MATLAB Simulink, and we will implement a proposed RNN or Transformer model in MATLAB as a final result of research. MATLAB is an industry-standard tool for R&D in engineering and robotics; hence future practical application is possible for the MATLAB model. We have added a proposed research summary to Table 1 for easy comparison with reviewed research.

As a result, we could outline the following research steps and goals:

- Find a suitable RNN or Transformer architecture to approximate mapping between joint dynamics and torques to calculate the precise trajectory of simulated arm and hand;

– Since a wide variety of RNN and Transformer architectures are already available in Python. We could expedite model architecture search and evaluation by using Python. Baseline musculoskeletal model in MATLAB Simulink environment will provide data for training;
– When a suitable model is found and demonstrate acceptable performance, assess its response time. If it is higher than 4 ms - improve the response time of the model to achieve real-time results with latency no more than 4 ms;
– As the last step, re-implement identified architecture as MATLAB model.

3.2 Neural Network Architecture

To solve the challenge of joint dynamics mapping to joint torques of a simulated artificial limb, we limit the list of suitable neural network architectures to the "vanilla" RNN as baseline one, LSTM and Universal Transformer, described below in detail.

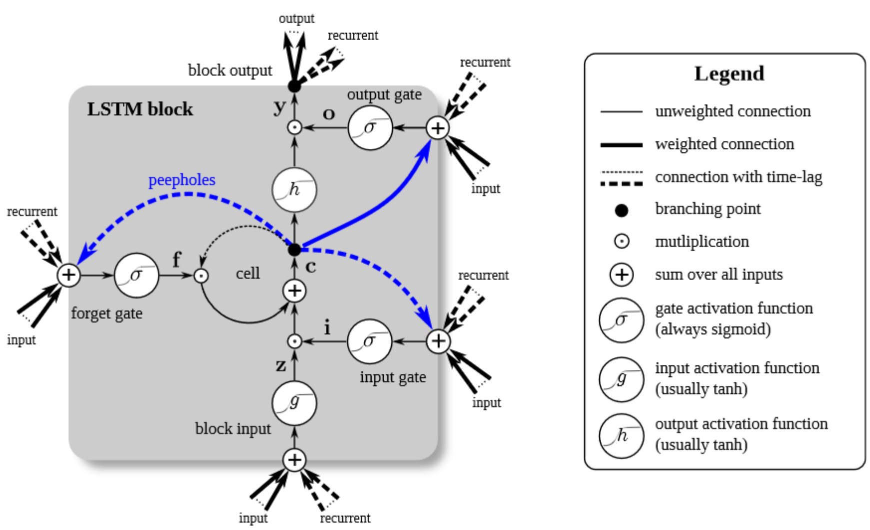

Fig. 1. "Vanilla" LSTM block setup as described in [8]

Long Short-Term Memory (LSTM) Based on results obtained in [8] it make sense to start with what was called "vanilla" LSTM block setup in that paper (see Figure 1). Such commonly used LSTM block architecture shows good performance on various data sets, and none of the improvements tested in [8] significantly improved this result. However, considering the real-time model response requirement in less than 4 ms, it makes sense to simplify the LSTM block to make it less computationally intensive. Following experiments from the same article [8] most significant simplifications to try are the coupling of input and forget gates and removing peephole connections because both of these changes does not decrease performance significantly. Hence, the final LSTM block setup we propose to start with is a common LSTM cell with coupled input and forget gates and removed peephole connections.

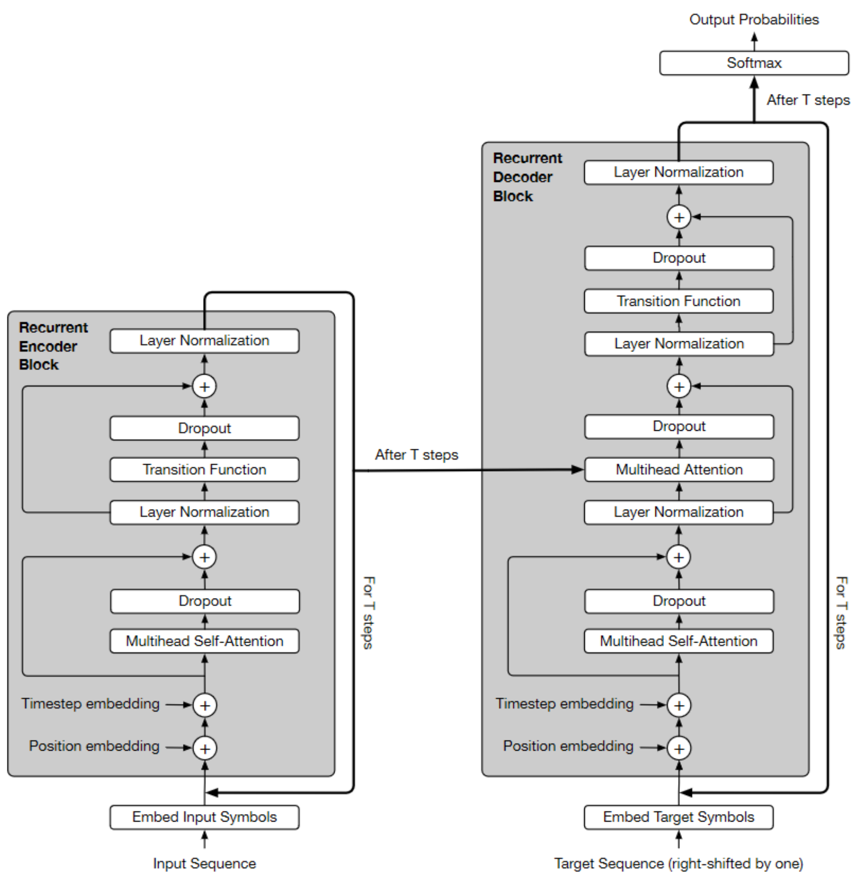

Fig. 2. Universal Transformer schema as defined in [4]

The most relevant hyperparameters are the learning rate and size of hidden layers. It confirmed by analysis done in [8] with the help of fANOVA framework for assessing hyperparameter importance. So, it makes sense to tune only these two hyperparameters while searching for the best performing model. Lastly, authors in [8] prove that hyperparameters could be treated as independent for the sake of tuning simplification. Measured interaction between two mentioned hyperparameters is insignificant. In [8] it also suggested tuning learning rate with the help of a smaller network to save time.

However, exact LSTM network architecture remains an open question, since even in [8] different network architectures were used for different datasets. As a result, we plan to search for exact network architecture through the experiments.

Transformer As an alternative to LSTM, if its performance would not be sufficient, we propose to adapt Universal Transformer (UT), which is introduced and fully described in [4]. As highlighted in [4] UT is "a parallel-in-time recurrent self-attentive sequence model which can be cast as a generalization of the Transformer model" and combine benefits of both feed-forward and RNN models. Authors claim that UT outperforms Transformers and LSTM models on various sequence-to-sequence tasks and confirmed this with conducted experiments in the scope of [4]. Also, authors of paper [4] provide a detailed schema of UT (see Figure 2) together with code to train and evaluate it, which could serve as a starting point to construct a model for arm and hand dynamics approximation based on UT.

Considering UT models' complexity, we assume that an overall model response time could be higher than the required 4 ms. The resulted model could require fine-tuning with regards to response time. Since UT models are easy to parallelize, as mentioned in [4], one possible measure could be to run many parallel models. Concrete decisions about UT model fine-tuning will be taken based on the results of experiments.

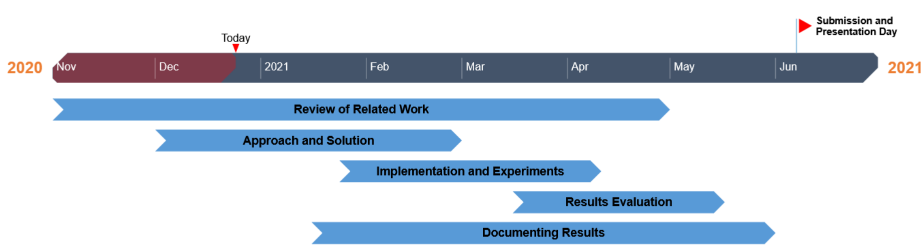

Fig. 3. Gantt chart of proposed research project

4 Current Results and Discussion

Execution plan of the proposed research project shown in Figure 3 with the indication of current progress. Currently, we have done an initial review of related researches and captured the result of gap analysis, which are described in section 2.3. However, as highlighted on the execution plan, this task is not entirely done yet, and we plan to keep reviewing related work in iterations.

We have also started to search for a suitable technical solution for the research scope described in section 3.1. Based on reviewed papers [8], and [4] we identified suitable candidates for neural networks architectures that should solve and approximate complex non-linear mapping between joint dynamics and torques to identify precise positions of the simulated artificial limb. Candidates architectures described in details in section 3.2. However, this list of possible models is not final and subject to changes during the project's later stages.

5 Summary and Future Work

This paper described a project proposal that aims to solve real-time simulation of arm and hand dynamics using ANN from joint angles, velocities, and accelerations as input data. Since mapping control signals directly to the simulated limb's position coordinates is complex and non-linear task, we propose instead finding a suitable model based on ANN that approximates mapping between joint dynamics and torques in the scope of this research. The musculoskeletal transformation from control signals to joint angles, velocities, and accelerations we consider as solved and its detailed description available in [11]. Based on reviewed papers, we identified that much research focuses on pattern recognition of the limited number of gestures with feed-forward shallow ANN or CNN, while continuous control of artificial limb is not in focus. Since the continuous control task of simulated limb based on joint dynamics is the sequential task, commonly-used feed-forward NN is not well suited. Instead, we propose approaching and solving it with neural network models explicitly developed for sequential data like in the NLP domain. We reviewed possible candidate architectures and outlined three of them to start with: the baseline RNN model, the LSTM model and the Universal Transformer model. The proposed LSTM model is a "vanilla" LSTM block setup described in [8] with coupled input and forget gates and without peephole connections. The Universal Transformer model is defined and fully described in [4]. To train the chosen ANN model, we decided to use a baseline musculoskeletal model described in [3].

As the next step, we plan to train chosen ANN models on given joint dynamics data from the baseline musculoskeletal model. After successful training, we will assess its performance and response time. If performance or response time is not acceptable, we will perform model fine-tuning to achieve the desired results. Suppose model fine-tuning will not help to achieve desired performance or response time levels. In that case, we plan to go back and re-iterate on possible solution approaches and ANN model selection activities until a suitable model is found. Under a suitable model, we meant a model that will achieve performance and response times on desired levels. For response time threshold defined as less than 4 ms, while for performance, no threshold defined yet, and it will be defined at later stages of the project. We target to implement the final ANN model architecture in MATLAB environment, while for model architecture search, we could use Python. Using Python, we will experiment and iterate faster by utilizing freely available out-of-the-box models and frameworks like TensorFlow or PyTorch. However, we will need to address connectivity between the Python candidate model and the baseline model implemented in the MATLAB Simulink environment. Any activities beyond the current project plan are not in focus, and we do not plan those at the current moment. However, this statement is subject to changes and could be adjusted within the subsequent project phases.

References

1. Agur, A.M.R.: Grant's Atlas of Anatomy, 10th Edition (1999)
2. Atzori, M., Cognolato, M., Müller, H.: Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. Frontiers in Neurorobotics **10**, 9–9 (2016). https://doi.org/10.3389/FNBOT.2016.00009
3. Boots, M.T., Hardesty, R., Sobinov, A., Gritsenko, V., Collinger, J.L., Fisher, L.E., Gaunt, R., Yakovenko, S.: Functional and structural moment arm validation for musculoskeletal models: A study of the human forearm and hand. bioRxiv (2020). https://doi.org/10.1101/2020.05.29.124644
4. Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., Łukasz Kaiser: Universal transformers. arXiv: Computation and Language (2018)
5. Dobrovolskyi, H., Keberle, N.: Collecting the seminal scientific abstracts with topic modelling, snowball sampling and citation analysis. ICTERI pp. 179–192 (2018)
6. Dobrovolskyi, H., Keberle, N.: On convergence of controlled snowball sampling for scientific abstracts collection. In: International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications. pp. 18–42 (2018)
7. Draye, J.P., Cheron, G., Bourgeois, M., Pavisic, D., Libert, G.: Identification of the human arm kinetics using dynamic recurrent neural networks. In: ESANN (1995)
8. Greff, K., Srivastava, R.K., Koutnik, J., Steunebrink, B.R., Schmidhuber, J.: Lstm: A search space odyssey. IEEE Transactions on Neural Networks **28**(10), 2222–2232 (2017)
9. Lecy, J.D., Beatty, K.E.: Representative literature reviews using constrained snowball sampling and citation network analysis (2012). https://doi.org/10.2139/SSRN.1992601
10. Sebelius, F., Eriksson, L., Balkenius, C., Laurell, T.: Myoelectric control of a computer animated hand: a new concept based on the combined use of a tree-structured artificial neural network and a data glove. Journal of Medical Engineering & Technology **30**(1), 2–10 (2006). https://doi.org/10.1080/03091900512331332546
11. Sobinov, A., Boots, M., Gritsenko, V., Fisher, L.E., Gaunt, R.A., Yakovenko, S.: Approximating complex musculoskeletal biomechanics using multidimensional autogenerating polynomials. bioRxiv p. 759878 (2019). https://doi.org/10.1101/759878
12. Wu, Y., Jiang, D., Liu, X., Bayford, R., Demosthenous, A.: A human–machine interface using electrical impedance tomography for hand prosthesis control. IEEE Transactions on Biomedical Circuits and Systems **12**(6), 1322–1333 (2018). https://doi.org/10.1109/TBCAS.2018.2878395
13. Zhao, J., Xie, Z., Jiang, L., Cai, H., Liu, H., Hirzinger, G.: A five-fingered underactuated prosthetic hand control scheme. In: The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. vol. 14, pp. 995–1000 (2006). https://doi.org/10.1109/BIOROB.2006.1639221