2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

Comparing Reinforcement Learning Agents and Supervised Learning Neural Networks for EMG-Based Decoding of Continuous Movements

Joseph Berman, Robert Hinson, and He Huang, *Senior Member, IEEE*

Abstract— **Recent work on electromyography (EMG)-based decoding of continuous joint kinematics has included model-based approaches, such as musculoskeletal modeling, as well as model-free approaches such as supervised learning neural networks (SLNN). This study aimed to present a new kinematics decoding framework based on reinforcement learning (RL), which combines machine learning and model-based approaches together. We compared the performance and robustness of our new method with those of the SLNN approach. EMG and kinematic data were collected from 5 able-bodied subjects while they performed flexion and extension of the metacarpophalangeal (MCP) and wrist joints simultaneously at both a slow and fast tempo. The data were used to train an RL agent and a SLNN for each of the 2 tempos. All the trained agents and SLNNs were tested with both fast and slow kinematic data. Pearson's correlation coefficient (r) and normalized root mean square error (NRMSE) between measured and estimated joint angles were used to determine performance. Our results suggest that the RL-based kinematics decoder is more robust to changes in movement speeds between training and testing data and has better performance than the SLNN.**

I. Introduction

Human neuromuscular signals, such as electromyograms (EMG), have been used as inputs to human-machine interface (HMI) systems for many applications in rehabilitation engineering, such as exoskeleton [1] and prosthesis control [2-5]. Recent work has focused on using EMG to continuously estimate the motion of multiple joints simultaneously. Much of this work can be divided into two approaches: model-based and model-free prediction. Model-based approaches include musculoskeletal models that use the empirically determined Hill-type muscle model [2-4], as well as state space kinematic models [6]. An example of a model-free approach is training a supervised learning neural network (SLNN) to map EMG data to joint torque or motions [7,8].

Each of these approaches has their advantages and disadvantages. Model-based approaches make use of empirically derived and studied systems, such as the Hill-type muscle model [9], making them more robust to previously unseen inputs [10]. However, they often require many simplifying assumptions and require the optimization of many musculotendon parameters. On the other hand, model-free approaches can be quickly trained or optimized and can recreate complex behaviors that are difficult to explicitly model. However, model-free approaches make use of black box functions and are heavily dependent on the amount and quality of training data provided.

In this paper, we use reinforcement learning (RL), an advanced machine learning method, which allows actions taken by an agent to be used as inputs to an environment that calculates the final output [11]. This method seeks to maximize a reward calculated by a predefined function. We trained RL agents to map processed EMG signals and previous kinematic states to wrist and metacarpophalangeal (MCP) joint torques which were then input to a forward dynamics model to calculate joint angles. Because EMG signals have a closer relationship to forces exerted by muscles than joint angles, this method has an advantage over the SLNNs which were trained to map processed EMG signals directly to joint angles. In this study, EMG and kinematic data of the wrist and MCP joints were recorded and used as training and testing data. The offline performances of the trained RL agents and SLNNs were compared to determine if it is feasible to use our RL-based framework for continuous joint kinematics decoding. We hypothesized that the RL-based decoder would be more robust and better able to predict kinematics than the SLNN.

II. Methods

A. Subjects

Experiments were approved by the Institutional Review Board of the University of North Carolina at Chapel Hill. Five able-bodied subjects (3 male, 2 female, age range 22-31, right-hand dominant) provided informed consent to participate.

B. Experiment Protocol

Kinematic and EMG data were recorded simultaneously from subjects with their right upper limb in a static posture in which the shoulder was relaxed, arm and forearm were in neutral posture, and elbow was at 90̊ of flexion. First, EMG data were collected while subjects performed the maximum voluntary contraction (MVC) for the flexion and extension of the wrist and MCP joints. Then, to cover a wide range of common movements, data were collected from subjects for 3 different movement patterns: isolated wrist flexion/extension, isolated MCP flexion/extension, and simultaneous wrist and

This work was supported by NSF under Grant 1856441.

J. Berman is with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695- 7115 USA, and also with the UNC/NC State Joint Department of Biomedical Engineering, North Carolina State University, Raleigh, NC 27695-7115 USA, (e-mail: jmberman@ncsu.edu).

R. Hinson and H. Huang are with the UNC/NC State Joint Department of Biomedical Engineering, North Carolina State University, Raleigh, NC 27695- 7115 USA, and also with the University of North Carolina at Chapel Hill, Chapel Hill, NC 27599 USA (e-mail: rmhinson@ncsu.edu; hhuang11@ncsu.edu).

978-1-7281-1178-0/21/$31.00 ©2021 IEEE

MCP flexion/extension. Each movement pattern was performed for 10 s for one slow and one fast fixed tempo (0.25 Hz and 0.5 Hz, respectively). Each movement pattern was repeated 3 times for each tempo, for a total of 18 trials for each subject. Subjects rested between each trial.

C. Data Acquisition

The right forearm of each subject was wiped with an alcohol pad and a bipolar electrode (Motion Lab Systems, Inc., USA) was placed over each of the 4 muscles in the forearm identified by palpation as shown in Fig. 1: flexor digitorum superficialis (FDS), extensor digitorum (ED), flexor carpi radialis (FCR), and extensor carpi radialis longus (ECRL). Electrodes were connected to an EMG system (MA300 DTU, Motion Lab Systems, USA) and signals were recorded at 1000 Hz. To record motion in the hand, a Leap Motion Controller (Leap Motion, Inc., USA) was chosen for its ability to capture hand and wrist positions [12]. The positions of the palm and metacarpal bone segments in the hand were recorded at 120 Hz simultaneously with EMG signals using the Leap Motion Controller placed on a table approximately 4” below the subject’s hand.

D. Data Processing

The envelopes of the EMG signals were obtained by calculating the mean absolute value (MAV) of a 200 ms sliding window, adjusted in 10 ms increments resulting in 100 Hz processed EMG data. The maximum value of each processed EMG signal from the MVC trials was used to normalize the corresponding processed EMG signal in each of the remaining trials.

The wrist angles during each trial were computed by finding the angle between the palm segment and the axis pointing directly away from the subject with the origin in the center of the Leap Motion Controller. The MCP angles were computed by finding the angle between the phalangeal segment and the palm segment. The kinematic data were downsampled to 100 Hz to match the processed EMG signals.

E. Reinforcement Learning Agent and Environment

The RL agent was implemented using the Deep Deterministic Policy Gradient (DDPG) algorithm with a neural network-based actor-critic structure for a continuous action space [13] with the Reinforcement Learning Toolbox in MATLAB 2021a (Mathworks, MA). The actor network μ received a state *s* and output an action *a*. The critic network *Q* received both *s* and *a* and output the expected long-term reward. The critic network was created using an addition layer to combine a neural network with an input layer for *s* and 2 hidden layers and a neural network with an input layer for *a* and 1 hidden layer. The actor network was created with an input layer for *s* and 2 hidden layers. Each hidden layer contained 128 neurons. The rectified linear unit activation function was used for each hidden layer and a tanh activation function was used for the output of the actor network. No activation function was used for the output of the critic network. The output of μ was scaled to the range [-3, 3] selected to provide a full range of feasible values of wrist and MCP joint torque. Temporally correlated noise sampled from an Ornstein-Uhlenbeck process [14] $\mathcal{N}$ was then added to this result during training to encourage action exploration. Target critic and actor networks Q' and μ' were created with the same structures as *Q* and μ. The target networks were used in the calculation of a target *y* used to train *Q*. The weights of the target networks were updated using a smoothing method to slowly track the weights of *Q* and μ to help avoid the divergence of *Q* [13].

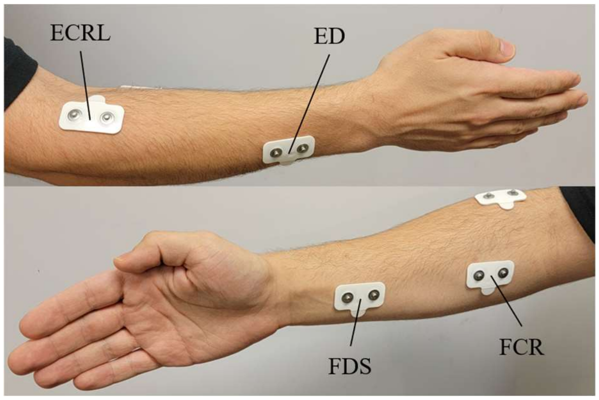

Fig. 1. Approximate locations of the 4 electrodes used and their corresponding targeted muscle.

Eight observations were used to define s_k. The observations included the 4 processed EMG signals and the estimated position and velocity of the wrist and MCP joints at the current timestep *k*: $\hat{\theta}_{w,k}$, $\hat{\dot{\theta}}_{w,k}$, $\hat{\theta}_{m,k}$, and $\hat{\dot{\theta}}_{m,k}$. The actions output from μ given s_k were the estimated torque values for the wrist and MCP joints: $\tau_{w,k}$, $\tau_{m,k}$. All estimated position, velocity, and torque values were used as inputs to a planar link-segment model of the hand and wrist [2], which was used

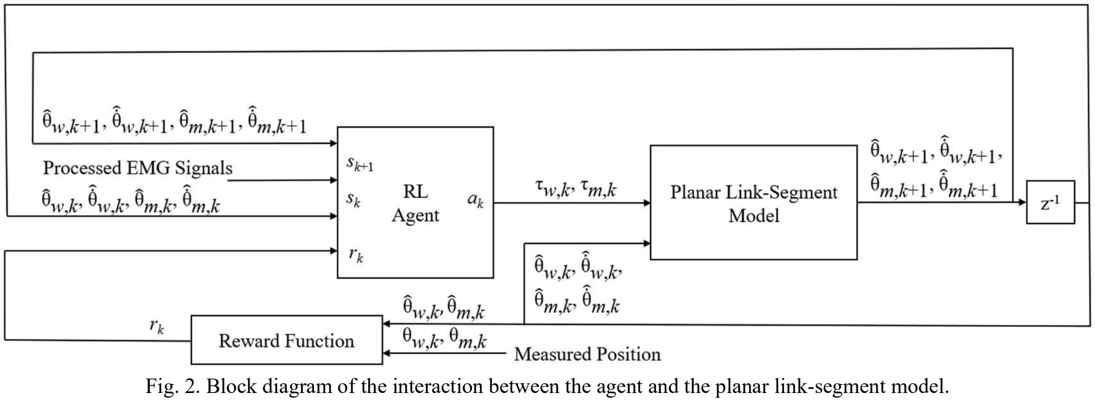
$\hat{\theta}_{w,k+1}, \hat{\dot{\theta}}_{w,k+1}, \hat{\theta}_{m,k+1}, \hat{\dot{\theta}}_{m,k+1}$
Processed EMG Signals
$\hat{\theta}_{w,k}, \hat{\dot{\theta}}_{w,k}, \hat{\theta}_{m,k}, \hat{\dot{\theta}}_{m,k}$
s_{k+1}
s_k
r_k
RL Agent
a_k
$\tau_{w,k}, \tau_{m,k}$
Planar Link-Segment Model
$\hat{\theta}_{w,k+1}, \hat{\dot{\theta}}_{w,k+1}, \hat{\theta}_{m,k+1}, \hat{\dot{\theta}}_{m,k+1}$
z^{-1}
$\hat{\theta}_{w,k}, \hat{\dot{\theta}}_{w,k}, \hat{\theta}_{m,k}, \hat{\dot{\theta}}_{m,k}$
$\hat{\theta}_{w,k}, \hat{\theta}_{m,k}$
r_k
Reward Function
$\theta_{w,k}, \theta_{m,k}$
Measured Position

Fig. 2. Block diagram of the interaction between the agent and the planar link-segment model.

```
Algorithm 1 DDPG algorithm
Initialize $Q(s,a|W^Q)$ and $\mu(s|W^\mu)$ with random weights $W^Q$ and $W^\mu$.
Initialize target networks $Q'$ and $\mu'$ with weights $W^Q$ and $W^\mu$.
Initialize replay buffer $D$.
Initialize $M = 50$, $\sigma^2 = 0.5$, $\beta = 10^{-5}$, $N = 1000$, $\gamma = 0.99$, $\alpha = 0.001$.
Initialize $K$ as the total number of samples in the training data.
for episode = 1, $M$ do
    Initialize $\mathcal{N}$ with variance $\sigma^2$ and variance decay rate $\beta$.
    Get initial state $s_1$.
    for $k = 1, K$ do
        Take action $a_k = [\tau_{w,k}, \tau_{m,k}] = \mu(s_k|W^\mu) + \mathcal{N}_k$.
        Get reward $r_k$ and new state $s_{k+1}$.
        Store transition $(s_k, a_k, r_k, s_{k+1})$ in $D$.
        Sample $N$ random transitions from $D$.
        Set $y_i = r_i + \gamma Q'(s_{i+1}, a'_{i+1} |W^{Q'})|_{a'_{i+1} = \mu'(s_{i+1}|W^{\mu'})}$
        Update the critic network by minimizing the loss:
            $L = \frac{1}{N}\Sigma_{i=1}^{N}(y_i - Q(s_i, a_i|W^Q))^2|_{a_i = \mu(s_i|W^\mu)}$
        Update the actor network with the sampled policy gradient:
            $\nabla_{W^\mu} J \approx \frac{1}{N}\Sigma_{i=1}^{N}\nabla_{a_i} Q(s_i, a_i|W^Q)\nabla_{W^\mu} a_i|_{a_i = \mu(s_i|W^\mu)}$
        Update the target networks:
            $W^{Q'} \leftarrow \alpha W^Q + (1 - \alpha)W^{Q'}$, $W^{\mu'} \leftarrow \alpha W^\mu + (1 - \alpha)W^{\mu'}$
    end for
end for
```

to compute the estimated kinematics at the next timestep $k+1$. The reward function at the current timestep was calculated as:

$$r_k = \frac{0.3}{0.3 + |\theta_{w,k} - \hat{\theta}_{w,k}|} + \frac{0.3}{0.3 + |\theta_{m,k} - \hat{\theta}_{m,k}|}$$

to allow higher reward values for smaller differences in estimated and measured positions for each joint, similar to a previously used reward function for joint position decoding using RL [15]. The DDPG algorithm used, adapted from [13], is shown as Algorithm 1. The interaction between the agent and the model is outlined in Fig. 2.

F. Supervised Learning Neural Network

A SLNN was created with the Deep Learning Toolbox in MATLAB 2021a (Mathworks, MA). The SLNN took the 4 processed EMG signals as inputs and directly output the estimated position of the wrist and MCP joints for the current timestep. The SLNN was tested while the number of neurons per hidden layer and then the number of hidden layers were incremented starting from 1 each until performance no longer significantly increased. One hidden layer with 5 neurons was chosen for maximum performance.

G. Training and Testing

The data from 6 of the 9 trials (2 randomly selected from each movement type) were used to train a SLNN and an RL-based decoder for each of the two fixed tempos. The data from the remaining three trials for each tempo were reserved as testing data. The RL-based decoder was allowed to train for a total of 50 episodes and the SLNN was allowed to train until the gradient of the mean square error (MSE) fell below 10^{-7}. All trained algorithms were used to predict joint angles for both the slow and fast tempo testing data.

H. Evaluation Metrics

For each trained SLNN and RL-based decoder, the kinematic predictions of the testing data were evaluated using Pearson's correlation coefficient (r) between measured and estimated angles of each joint. The normalized root mean square error (NRMSE) between measured and estimated angles of each joint was calculated by dividing the root mean square error (RMSE) by the difference of the maximum and minimum measured joint angles. The values of r and NRMSE were averaged across all subjects and both joints for each testing case.

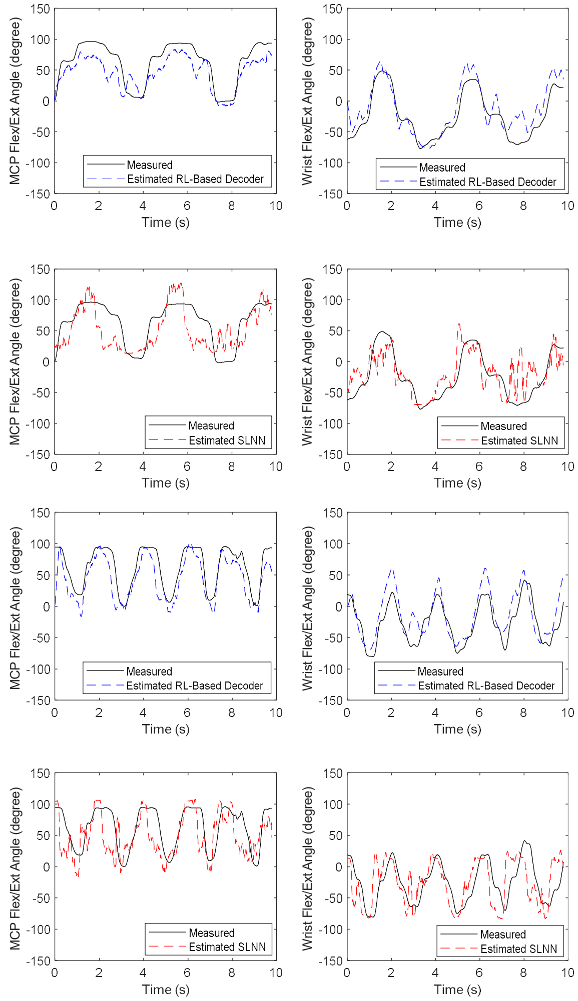

Fig. 3. Representative plots of simultaneously predicted MCP (left) and wrist (right) joint angles by an RL-based decoder (blue) and a SLNN (red) compared to measured joint angles (black) for one subject. Both algorithms were trained with slow kinematic data and plots were generated for predictions on slow (top 2 rows) and fast (bottom 2 rows) kinematic testing data.

I. Statistical Analysis

A student's t-test was conducted to compare correlation and NRMSE between the two algorithms (RL and SLNN) for each testing case. Differences were considered statistically significant for $p<0.05$. All results are represented as mean ± standard deviation unless specified otherwise.

III. Results and Discussion

Trained RL-based decoders and SLNNs were able to provide reasonable predictions of able-bodied subjects' wrist and MCP joint kinematics (Fig. 3). Evaluation of the RL-based decoders and SLNNs trained with slow kinematic data showed slow kinematic testing data was predicted with similar accuracy, both in terms of correlation (RL: 0.67 ±

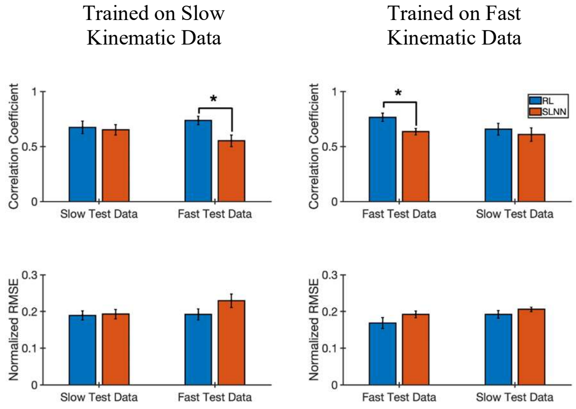

Fig. 4. Summary of RL-based decoder and SLNN kinematic prediction accuracy. Correlation coefficient (top) and NRMSE (bottom) are shown for RL agents and SLNNs trained on slow kinematic data (left) and fast kinematic data (right). Error bars represent standard error (N = 5) and stars indicate significance at the p < 0.05 level.

0.06; SLNN: 0.65 ± 0.05; p = 0.78) and NRMSE (RL: 0.19 ± 0.01; SLNN: 0.19 ± 0.01; p = 0.84). However, the RL-based decoders predicted fast kinematic data with significantly higher correlation (RL: 0.74 ± 0.04; SLNN: 0.55 ± 0.05; p = 0.02) and lower NRMSE (RL: 0.19 ± 0.01; SLNN: 0.23 ± 0.02; p = 0.15) than the SLNNs.

When RL-based decoders and SLNNs were trained with fast kinematic data, the predictions of the RL-based decoders showed significantly higher correlation (RL: 0.77 ± 0.04; SLNN: 0.64 ± 0.03; p = 0.03) as well as lower NRMSE (RL0.17 ± 0.01; SLNN: 0.19 ± 0.01; p = 0.21) when tested on fast kinematic data in comparison to the SLNN decoders. When tested on slow kinematic data, the RL-based decoder and SLNN predictions had similar correlation (RL: 0.66 ± 0.05; SLNN: 0.61 ± 0.06; p = 0.56) and NRMSE (RL: 0.19 ± 0.01; SLNN: 0.21 ± 0.01; p = 0.28). Performance metrics are summarized in Fig. 4.

In all test conditions, the RL-based decoders demonstrated either similar performance to the SLNNs or significantly better performance, as demonstrated by correlation and NRMSE. Of particular note, the RL-based decoder trained with slow kinematic data predicted fast kinematic test data significantly better than the SLNNs trained on the slow kinematic data. These results indicate the RL-based decoders are more robust to inputs that differ from the provided training data than the SLNNs.

A potential reason for this increased robustness is that the RL-based decoders were trained to predict joint torques, which have a stronger relationship with the EMG inputs than the joint angles the SLNNs were trained to predict. By using RL to train these agents, we were able to explicitly define a forward dynamics model for the joints of interest to relate the predicted torques of the agent to the measured kinematics. This model could not be implemented with the SLNNs, as traditional NN optimization using backpropagation requires the gradients of all functions in the system to be explicitly known, demonstrating the flexibility afforded by using RL.

IV. Conclusion

This study compared our RL-based decoder with a SLNN for predicting continuous joint angles when trained and tested using different combinations of fast and slow tempo kinematic data. For all testing cases, the RL-based decoders performed similar to or better than the SLNNs. In addition, the predictions of the RL-based decoder achieved significantly higher correlation values than the SLNN when trained using slow kinematic training data and tested using fast kinematic data. This suggests the RL-based decoder is more robust to differences in training and testing data than a SLNN. Our results show that it is feasible to use the RL-based framework presented in this paper for continuous joint kinematics decoding. Future studies will evaluate the performance of the RL-based decoder when trained and tested on data from amputees.

References

[1] K. Kiguchi and Y. Hayashi, “An EMG-based control for an upper-limb power-assist exoskeleton robot,” IEEE Trans. Syst. Man, Cybern. Part B Cybern., vol. 42, no. 4, Aug. 2012.

[2] D. L. Crouch and H. Huang, “Lumped-parameter electromyogram-driven musculoskeletal hand model: A potential platform for real-time prosthesis control,” J. Biomech., vol. 49, no. 16, Dec. 2016.

[3] L. Pan, D. L. Crouch, and H. Huang, “Myoelectric Control Based on a Generic Musculoskeletal Model: Toward a Multi-User Neural-Machine Interface,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 26, no. 7, Jul. 2018.

[4] M. Sartori, G. Durandau, S. Došen, and D. Farina, “Robust simultaneous myoelectric control of multiple degrees of freedom in wrist-hand prostheses by real-time neuromusculoskeletal modeling,” J. Neural Eng., vol. 15, no. 6, 2018.

[5] T. A. Kuiken, G. Li, B. A. Lock, et al., “Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms,” JAMA - J. Am. Med. Assoc., vol. 301, no. 6, 2009.

[6] Q. Ding, J. Han, and X. Zhao, “Continuous Estimation of Human Multi-Joint Angles from sEMG Using a State-Space Model,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 25, no. 9, Sep. 2017.

[7] C. Loconsole, S. Dettori, A. Frisoli, C. A. Avizzano, and M. Bergamasco, “An EMG-based approach for on-line predicted torque control in robotic-assisted rehabilitation,” in IEEE Haptics Symposium, HAPTICS, Feb. 2014, pp. 181–186.

[8] S. Muceli and D. Farina, “Simultaneous and proportional estimation of hand kinematics from EMG during mirrored movements at multiple degrees-of-freedom,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 20, no. 3, pp. 371–378, May 2012.

[9] F. E. Zajac, “Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control.,” Critical reviews in biomedical engineering, vol. 17, no. 4. 1989.

[10] L. Pan, D. L. Crouch and H. Huang, "Comparing EMG-Based Human-Machine Interfaces for Estimating Continuous, Coordinated Movements," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 27, no. 10, pp. 2145-2154, Oct. 2019.

[11] Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine.

[12] A. H. Butt, E. Rovini, C. Dolciotti, G. De Petris, P. Bongioanni, M. C. Carboncini, and F. Cavallo, “Objective and automatic classification of Parkinson disease with Leap motion controller,” BioMedical Engineering OnLine, vol. 17, no. 1, 2018.

[13] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, et al., "Continuous control with deep reinforcement learning", ICLR, 2016.

[14] G. E. Uhlenbeck and L. S. Ornstein, “On the theory of the Brownian motion,” Phys. Rev., vol. 36, no. 5, 1930.

[15] W. Wu, K. R. Saul, and H. (Helen) Huang, “Using Reinforcement Learning to Estimate Human Joint Moments From Electromyography or Joint Kinematics: An Alternative Solution to Musculoskeletal-Based Biomechanics,” J. Biomech. Eng., vol. 143, no. 4, Jul. 2021.