Deep Learning Based Upper-limb Motion Estimation Using Surface Electromyography

Tianzhe Bao

University of Leeds

School of Electronic and Electrical Engineering

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

November, 2021

This thesis is dedicated to those who have been inspirational and supportive to me in the past thirty years.

Intellectual Property Statement

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

The right of Tianzhe Bao to be identified as Author of this work has been asserted by him in accordance with the Copyright, Designs and Patents Act 1988.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to Dr. Zhi-Qiang Zhang for all of his support in my PhD journey from the beginning. I am more than grateful to him for providing me with constructive feedback on my work and daily life. I would also like to thank my cosupervisors Prof. Sheng Quan Xie, Dr. Syed Ali Raza Zaidi, and Prof. Ian Robertson for the academic inspiration, technical guidance, financial assistance, as well the encouragement, patience and opportunities presented to me during my PhD period.

I would also like to express my gratitude to Dr. Pengfei Yang from Xidian University and Prof. Ping Liu from Shandong Agricultural University for their assistance and support.

I wish to acknowledge the help from my colleagues. They are Dr. Emmanuel Ayodele, Dr. Tuan Anh Tang, Dr. Miaomiao Liu, Dr. You You, Dr. Zhenhong Li, Dr. Wei Meng, Mr. Qingxiang Kong, Mr. Chao Wang, Mr. Lin Wu, Mr. Yihui Zhao, Mr. Kun Qian, Miss Yue Zhang, and Miss Orla Gilson. Thanks for all the co-operations we had and for all the fun we shared.

Finally, I would like to thank my family members who have given me their unwavering love and support throughout the journey of study. Thank you very much for being with me through all the ups and downs.

Abstract

To advance human-machine interfaces (HMI) that can help disabled people reconstruct lost functions of upper-limbs, machine learning (ML) techniques, particularly classification-based pattern recognition (PR), have been extensively implemented to decode human movement intentions from surface electromyography (sEMG) signals. However, performances of ML can be substantially affected, or even limited, by feature engineering that requires expertise in both domain knowledge and experimental experience. To overcome this limitation, researchers are now focusing on deep learning (DL) techniques to derive informative, representative, and transferable features from raw data automatically. Despite some progress reported in recent literature, it is still very challenging to achieve reliable and robust interpretation of user intentions in practical scenarios. This is mainly because of the high complexity of upper-limb motions and the non-stable characteristics of sEMG signals. Besides, the PR scheme only identifies discrete states of motion. To complete coordinated tasks such as grasping, users have to rely on a sequential on/off control of each individual function, which is inherently different from the simultaneous and proportional control (SPC) strategy adopted by the natural motions of upper-limbs.

The aim of this thesis is to develop and advance several DL techniques for the estimation of upper-limb motions from sEMG, and the work is centred on three themes: 1) to improve the reliability of gesture recognition by rejecting uncertain classification outcomes; 2) to build regression frameworks for joint kinematics estimation that enables SPC; and 3) to reduce the degradation of estimation performances when DL model is applied to a new individual. In order to achieve these objectives, the following efforts were made: 1) a confidence model was designed to predict the possibility of correctness with regard to each classification of convolutional neural networks (CNN), such that the uncertain recognition can be identified and rejected;

2) a hybrid framework using CNN for deep feature extraction and long short-term memory neural network (LSTM) was constructed to conduct sequence regression, which could simultaneously exploit the temporal and spatial information in sEMG data; 3) the hybrid framework was further extended by integrating Kalman filter with LSTM units in the recursive learning process, obtaining a deep Kalman filter network (DKFN) to perform kinematics estimation more effectively; and 4) a novel regression scheme was proposed for supervised domain adaptation (SDA), based on which the model generalisation among subjects can be substantially enhanced.

Contents

List of Figures

List of Tables

Abbreviations

Acc	Accuracy
ADAM	Adaptive Moment Estimation
AE	Autoencoder
AI	Artificial Intelligence
ANOVA	One-way Analysis of Variance
ANN	Artificial Neural Network
BMEC	Balanced Mean Effective Confidence
ConfScore	Confidence Score
CNN	Convolutional Neural Network
DA	Domain Adaptation
DL	Deep Learning
DOF	Degree of Freedom
Err	Error Rate
FA	False Acceptance
FC	Fully Connected Layer
FFT	Fast Fourier Transform
FR	False Rejection
FT	Fine-tuning
GRU	Gated Recurrent Unit
HMI	Human-Machine Interface
LDA	Linear Discriminant Analysis
LSTM	Long Short-Term Memory
LSTM-KF	LSTM-based Kalman Filter
MEC	Mean Effective Confidence
ML	Machine Learning

MLP	Multilayer Perceptron
MMD	Maximum Mean Discrepancy
MSE	Mean Square Error
PCA	Principal Component Analysis
PR	Pattern Recognition
RBF	Radial Basis Function
RF	Random Forest
RMS	Root Mean Square
RNN	Recurrent Neural Network
SDGM	Stochastic Gradient Descent with Momentum
sEMG	Surface Electromyography
SDA	Supervised Domain Adaptation
SPC	Simultaneous and Proportional Control
STL	Supervised Transfer Learning
SVM	Support Vector Machine
SVR	Support Vector Regression
TA	True Acceptance
TAR	True Acceptance Rate
TL	Transfer Learning
TR	True Rejection
TRR	True Rejection Rate
UDA	Unsupervised Domain Adaptation
UTL	Unsupervised Transfer Learning

Chapter 1

Introduction

1.1 Motivation and Objectives

From simple grasps to dexterous manipulations, the upper extremity performs a large number of essential tasks in human activities. Thereby the upper limb amputation, from which millions of individuals are suffering worldwide [7], can affect nearly all aspects of daily life. To help disabled people reconstruct lost functions of hands or arms, surface electromyography (sEMG), the bio-electricity detected from skeletal muscles non-invasively using surface electrodes attached on the skin, has been vastly applied in the control of human-machine interfaces (HMI) including intelligent prostheses and exoskeleton robotics.

Despite the advances in mechanical design of dexterous artificial hands, a vast majority of commercial products still use the threshold control strategy proposed several decades ago [8]. A pair of electrodes are placed on antagonist muscles, and the envelopes of sEMG are extracted for the actuation of prostheses in a single degree of freedom (DOF) such as hand opening/closing or wrist flexion/extension. Once a trigger signal, commonly the muscle co-contraction, is detected to be above the threshold, the control mode is switched to actuate movements in another DOF [9]. Although this strategy is easy to perform, only very limited DOFs can be involved for prosthesis control. Furthermore, since there is no one-to-one

relationship between muscle activities and controlled motions, the lack of intuitiveness can increase the cognitive burden of users, which also contributes to the dissatisfaction of sEMG-based prosthetic devices [10].

In order to increase the user acceptance of myoelectric systems, it is highly desirable to achieve intuitive and multi-functional control by decoding users' movement intentions from their sEMG patterns directly and correctly. To this end, machine learning (ML) techniques, particularly the pattern recognition (PR) scheme that identifies certain classes of upper-limb motions, have attracted significant research interests in both academia and industry [11–13]. By learning information from collected data, ML works to build a linear or non-linear relationship between sEMG signals and target motions. In general, the implementation of ML includes two core parts: feature engineering and classification. The former is composed of feature extraction, channel selection and dimension reduction. It intends to construct representative sEMG features that are associated with certain muscle activations. Meanwhile, the latter exploits algorithmic strategies to assign input features to one of pre-defined movements/gestures. Both processes can be remarkably influential to decoding performances.

However, feature engineering usually requires expertise in both domain knowledge and experimental experience. In this process, some useful temporal-spatial information, such as the stochastic nature of sEMG signals, may be easily buried [14, 15]. More recently, deep learning (DL) approaches, including convolutional neural network (CNN), recurrent neural network (RNN) and auto-encoder (AE), are providing a new perspective for feature learning. To be specific, DL exploits multiple layers of neural networks to extract representative features from sEMG. In particular, these layers of features are not designed by human engineers but learned from raw data using a general-purpose learning procedure [16]. By composing non-linear modules that transform the feature from a lower level (starting from the raw data) to a higher (usually more abstract) level, complex functions can be built to discover intricate structures in high-dimensional data. For instance, CNN utilises convolution layers with learnable filters to discover specific types of patterns in the input image. Higher layers of CNN can detect motifs that

correspond to some parts of target objects rather than just preliminary features of pixels [16]. In short, the main advantage of DL can be summarised as the capability to derive informative, representative, and transferable features from raw data automatically [17]. This process, in many cases, is verified to be more convenient and efficient than feature engineering.

Despite the potential advances of DL, related research on motion estimation using sEMG is still at the very beginning. More importantly, the usability of DL in practical scenarios can be greatly hindered by a number of technical challenges. Firstly, user-safety is critical in myoelectric control systems. However, the performance of a well-trained model can be influenced by various real-time disturbances, resulting in unintended estimations and even unacceptable risks to users [18]. Secondly, PR-scheme only identifies discrete states of gestures and thereby requires users to apply a sequential on/off control of each individual function to complete dexterous manipulation tasks. This is inherently different from the natural control strategy of upper-limbs [19, 20]. For instance, the reaching or grasping movements usually involve simultaneous actuation of wrist orientation and hand posture, rather than the sequential activations of these DOFs. In addition, as a type of bio-electricity, sEMG is user-specific and time-varying. This property often results in a poor generalisation of ML/DL models in the cross-user scenario and long-term utilisation [21, 22].

By addressing some of the aforementioned challenges, the aim of this research is to develop and advance several DL techniques to further enhance the usability of motion estimation in upper-limb myoelectric systems. This target can help to provide a powerful ”brain” for HMI to be operated following users' movement intentions smoothly and naturally. Considering that most upper-limb functions are conducted by hands, this thesis mainly focuses on the decoding of wrist and hand motions using DL techniques. These motions can include a variety of hand postures, the bending of fingers, and rotations of wrist in multiple DOFs. In particular, there are three main objectives in this thesis, including:

1) To investigate the confidence estimation of DL in wrist movement/gesture recognition, such that uncertain results of classifier can be accurately identified

to improve the reliability of myoelectric system.

2) To develop DL-based regression framework for multi-DOFs wrist/fingers kinematics estimation, thus enabling the simultaneous and proportional control (SPC) scheme that is adopted by the natural movements of upper-limbs.

3) To investigate the shift of sEMG patterns in practical scenarios and associated effects on DL, based on which effective learning method can be proposed to improve model robustness and generalisation.

1.2 Contributions of This Research

The original **technical contributions** of this research include:

1) Design of a confidence estimation model and a rejection evaluation metric for CNN-based hand movement recognition using sEMG.

2) Development of an online platform to validate the effectiveness of confidence-based rejection in real-time experiments.

3) Investigation of CNN-based feature extraction in multi-DOF joint kinematics estimation.

4) Exploration of the hybrid framework using CNN and LSTM to fully explore the temporal-spatial information in sEMG.

5) Incorporation of LSTM units into the computational graph of Kalman filter to exploit the advantages of two methods in sequence learning.

6) Study on domain shift impacts in the inter-subject scenario and development of a novel regression scheme to support supervised domain adaptation by extracting domain-invariant features among subjects.

1.3 Thesis Outline

This thesis consists of 7 chapters. Besides the abstract and introduction, a brief overview of the remaining chapters and the corresponding contributions is presented below.

Chapter 2 reviews related efforts in upper-limb motion estimation using ML/DL. It mainly focuses on the comparison of feature engineering and feature learning, the post-processing techniques to enhance model reliability, the regression frameworks to enable SPC, and the transfer learning approaches for efficient model recalibration.

Chapter 3 presents the research on confidence estimation in CNN-based hand gesture classification. In this chapter, a novel approach is proposed to estimate the probability of correctness for each classification. Specifically, a confidence estimation model is established to generate confidence scores (ConfScore) based on posterior probabilities of CNN, and an objective function is specially designed to train the parameters of this model. In addition, a comprehensive metric that combines the true acceptance rate and the true rejection rate is proposed to evaluate the rejection performance of ConfScore, such that a better balance between the system security and control lag could be achieved in the design/assessment of a rejection option.

Chapter 4 proposes a hybrid DL model for multi-DOF joint kinematics estimation to achieve SPC in myoelectric systems. The main concern is that CNN can only exploit spatial correlations but ignores the temporal dependencies during continuous muscle contractions. In contrast, long short-term memory neural network (LSTM) is able to capture long-term and non-linear dynamics of time-series data, but it usually works with hand-crafted features that are not sufficiently informative. Inspired by the advantages of CNN and LSTM, a CNN-LSTM hybrid framework is proposed to fully explore the temporal-spatial information in sEMG. More specifically, a CNN model is firstly established to extract deep features from sEMG data, then these features are processed via LSTM-based sequence regression to estimate wrist kinematics more accurately.

Chapter 5 presents an extended work of Chapter 4. The consideration is that DL techniques only intend to capture the relationship between sEMG and target kinematics, but ignore the prior knowledge of the system. Therefore, estimation performances can be easily impacted by measurement noise that are quite common in the detection of sEMG signals. By contrast, Kalman filter applies

Kalman gain to combine the internal transition model and the observation model effectively, thereby it can be more robust to the noisy measurements. To this end, a novel architecture named LSTM-based Kalman filter process (LSTM-KF) is established to further enhance the sequence regression on high-level features extracted by CNN. Specifically, LSTM-KF adopts the computational graph of KF but estimates parameters of the transition/observation model and the Kalman gain from data using LSTM modules. With this process, the advantages of KF and LSTM can be exploited jointly.

Chapter 6 attempts to address the domain shift impacts on DL in the inter-subject scenario. The motivation is that a pre-trained CNN model usually suffers from severe degradation when testing on a new individual, whereas training a new model from scratch is burdensome. To this end, a novel regression scheme is presented for supervised domain adaptation, based on which the model generalisation can be effectively enhanced. Specifically, a two-stream CNN with shared weights is established to exploit source and target sEMG data simultaneously, such that domain-invariant features can be extracted. To tune CNN weights, both regression losses and a domain discrepancy loss are employed, where the former enable supervised learning and the latter minimises distribution divergences between two domains.

Chapter 7 concludes the efforts of this thesis. It also discusses some potential research directions that could be derived from current work.

1.4 Publication List

The works presented in this thesis have been published in peer-reviewed journals and conferences, and the contents are taken from some of the following articles. The candidate worked as the first author, and the co-authors acted in an advisory capacity, providing feedback, general guidance, and comments.

Chapter 3

Tianzhe Bao, Syed Ali Raza Zaidi, Sheng Quan Xie, Pengfei Yang and Zhi-Qiang Zhang. "CNN Confidence Estimation for Rejection-based Hand Gesture

Classification in Myoelectric Control". IEEE Transactions on Human Machine System (in press).

Chapter 4

Tianzhe Bao, Syed Ali Raza Zaidi, Sheng Quan Xie and Zhi-Qiang Zhang. "Surface-EMG based Wrist Kinematics Estimation using Convolutional Neural Network." In IEEE 16th International Conference on Wearable and Implantable Body Sensor Networks (BSN), pp. 1-4. IEEE, 2019.

Tianzhe Bao, Syed Ali Raza Zaidi, Sheng Quan Xie, Pengfei Yang and Zhi-Qiang Zhang. "A CNN-LSTM Hybrid Model for Wrist Kinematics Estimation Using Surface Electromyography." IEEE Transactions on Instrumentation and Measurement, 2020, 70: 1-9.

Chapter 5

Tianzhe Bao, Yihui Zhao, Syed Ali Raza Zaidi, Pengfei Yang and Zhi-Qiang Zhang. "A Deep Kalman Filter Network for Hand Kinematics Estimation using sEMG". Pattern Recognition Letters, 143 (2021): 88-94.

Chapter 6

Tianzhe Bao, Syed Ali Raza Zaidi, Sheng Quan Xie, Pengfei Yang and Zhi-Qiang Zhang. "Inter-subject Domain Adaptation for CNN-based Wrist Kinematics Estimation using sEMG". IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 29, pp. 1068-1078, 2021.

Chapter 2

ML and DL for Upper-limb Motion Estimation Using sEMG

2.1 Introduction

In the past decade, ML techniques have been extensively implemented to decode movement intentions from sEMG signals. By learning useful information from data, ML does not rely on the physiological theory to identify human motion intentions, has lower requirement on parameter determination [23], and can be insensitive to the electrode configuration [24]. To date, the classification-based PR scheme has gained most attentions in the research of ML in both literature and commercial products. It aims to identify certain classes of movements/gestures by assuming that sEMG patterns can be reproducible for the same motion but separable among the different [1]. With well-designed feature sets and classification algorithms, encouraging results ($> 95\%$ on a large repertoire of motions) have been reported in numerous studies [11]. To better exploit the temporal and spatial information of sEMG, many researchers are now focusing on feature learning that is achieved by DL approaches. Different from feature engineering exploited by ML, this process captures useful representations from raw data using algorithms rather than human expertise. In many cases, the complex mapping

between sEMG and target motions can be built more effectively and efficiently.

This chapter provides an overview of ML/DL techniques in upper-limb myoelectric control. Firstly, the process of ML/DL based motion estimation is formulated. The related efforts in feature engineering and feature learning is then introduced. In addition, recent advancements in post-processing, joint kinematics estimation, and transfer learning are reviewed. The problems of existing approaches will also be highlighted for the purpose of improvement.

2.2 From Feature Engineering Towards Feature Learning

2.2.1 Problem Statement

The ML/DL based motion estimation can be formulated as a function that maps sEMG signal to target motions:

$$\widehat{\boldsymbol{y}}_t = f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t\right), \tag{2.1}$$

where $\widehat{\boldsymbol{y}}_t$ denotes the estimation of t^{th} data segment, $\boldsymbol{x}_t$ represents sEMG segments that are obtained by dividing a stream of sEMG signals into overlapping windows of length L, and $f_{\boldsymbol{\theta}}(\bullet)$ denotes the algorithmic strategy. Parameters $\boldsymbol{\theta}$ can be optimised by minimising the loss function $\mathcal{L}\left(\boldsymbol{y}, \widehat{\boldsymbol{y}}, \boldsymbol{\theta}\right)$ that evaluates how far the distribution of model predictions is from that of measured movements, where $\widehat{\boldsymbol{y}}$ and $\boldsymbol{y}$ denote model predictions and corresponding measurements (ground-truth), respectively. It is noted that the formulation of $\mathcal{L}\left(\boldsymbol{y}, \widehat{\boldsymbol{y}}, \boldsymbol{\theta}\right)$ varies among different learning tasks. Take the binary classification as example, the dissimilarity between two probability distributions is usually calculated using the cross-entropy loss:

$$\mathcal{L}\left(\boldsymbol{y}, \widehat{\boldsymbol{y}}, \boldsymbol{\theta}\right) = \sum_{t=1}^{N} y_t \cdot \log \hat{y}_t + \left(1 - y_t\right) \cdot \log\left(1 - \hat{y}_t\right), \tag{2.2}$$

where N denotes the total number of training samples. Herein y_t is a scalar that takes the value of 0 or 1 to denote the true class of t^{th} training sample, $\hat{y}_t$ refers to the probability calculated by ML models

Since sEMG signals are non-stationary and random waves, information offered by raw data can only be utilised efficiently through feature construction with sliding windows. Therefore, a key procedure in Eq. (2.1) is to obtain informative $\boldsymbol{x}_t$ to preserve the separability of sEMG patterns. In fact, Scheme *et al.* [25] believes that the classification task is essentially a linear problem if feature representation is sufficient. In ML this target is achieved by feature engineering, including feature extraction, channel selection, and dimension reduction. By contrast, DL exploits feature learning that aims to create a better representation by extracting high-level features from input data using multiple layers of processing blocks. In the following part, a brief overview on related works of feature engineering and feature learning in upper-limb myoelectric control will be provided.

2.2.2 Feature Engineering

In ML techniques, $\boldsymbol{x}_t$ denotes hand-crafted features, including time domain features (TD), time-serial domain features (TSD), frequency domain features (FD), and time-scale or time-frequency domain features (TSCD/TFD) [26]. Specifically, TD features are calculated based on statistical or physical analysis of signal amplitudes, mainly designed to indicate signal energy, activation level of motor units, or duration of contraction [27]. They include mean absolute value (MAV), integrated absolute value (IAV), variance (VAR), mean absolute value slope (MAVS), Willison amplitude (WAMP), zero crossing (ZC), slope sign changes (SSC), waveform length (WL), root mean square (RMS), etc. TSD features, such as autoregressive coefficients (AR) and Cepstral coefficients (CC), are developed to examine the stability of signals in a time series. By contrast, FD features are statistical properties of power spectrum. They can help to analyse the rate and shape of motor unit action potentials (MUAP). This information is highly related with muscle fatigue. Typical FD features include power spectrum (PS), frequency ratio (FR), mean and median of signal frequencies (FMN, FMD). Moreover, TFD

features, which can be obtained via short-time Fourier transform (STFT), wavelet transform (WT) or wavelet packet transform (WPT), are proposed to integrate temporal and spectral information to identify transient/steady state patterns of muscle contractions. Table 2.1 summarises the mathematical definitions of some commonly used features [26, 28, 29].

Since one feature can only provide limited information, most practices adopt the idea of combing multi-features from different groups [30]. The representative works are the Hudgins' feature set [31] and the Phinyomark's feature set [26]. Apparently, the feature combination will result in a very high dimension space for $f_{\boldsymbol{\theta}}(\bullet)$ and information redundancy. In this context, dimension reduction can be performed to decrease the computation time while preserving the most discriminative information. Prevalent efforts include principal component analysis (PCA) [30], non-negative matrix factorisation (NMF) [32], uncorrelated linear discriminant analysis (ULDA) [33], and independent component analysis (ICA) [34], etc. Besides, decreasing the number of electrodes/channels can also help to reduce feature dimensions and the computational load. Effective approaches such as sequential forward [35] and backward selection [36] have also been proposed.

2.2.3 Feature Learning

In DL structures, both $\boldsymbol{x}_t$ and $f_{\boldsymbol{\theta}}(\bullet)$ can be learned from data simultaneously. Among all DL techniques, CNN is the earliest and also the mostly investigated one in myoelectric control. This is because CNN, a neural network originally designed to recognise visual patterns directly from pixel images, presents an impressive capability of feature learning on high-dimensional raw data. Therefore, CNN is able to exploit the spatial information of sEMG, such that the correlations of muscle groups can be fully considered. Typically, CNN consists of multiple convolution layers, pooling layers and fully-connected layers. In particular, convolution layers are applied to construct discriminative features by using a set of learnable filters/kernels. After training, filters are learned to detect specific types of features in the input, and those features can be mapped to target motions via fully connected layers. In fact, the convolutional layers normally work as feature

Table 2.1: Mathematical definitions of some commonly used hand-crafted features. Herein, $\boldsymbol{x}_t(l)$ denotes the l^{th} element of t^{th} sliding window, where $l \in [1, L]$ and L denotes the window length.

TD Features	
Integrated EMG	$IMEG_t = \sum_{l=1}^{L} \lvert\boldsymbol{x}_t(l)\rvert$
Mean absolute value	$MAV_t = \frac{1}{L}\sum_{l=1}^{L} \lvert\boldsymbol{x}_t(l)\rvert$
Variance	$VAR_t = \frac{1}{L}\sum_{l=1}^{L} (\boldsymbol{x}_t(l) - \bar{\boldsymbol{x}}_t)^2$
Mean absolute value slope	$MAVS_t = MAV_{t+1} - MAV_t$
Waveform length	$WL_t = \sum_{l=1}^{L-1} \lvert\boldsymbol{x}_t(l) - \boldsymbol{x}_t(l+1)\rvert$
Root mean square	$RMS_t = \sqrt{\frac{1}{L}\sum_{l=1}^{L} \boldsymbol{x}_t(l)^2}$
TSD Features	
Autoregressive coefficients	$\boldsymbol{x}_t(l) = \sum_{p=1}^{P} a_p \boldsymbol{x}_t(l-p)$, P is the order of AR model, a_p is the coefficient.
Cepstral coefficients	$c_1 = -a_1, c_p = -a_p - \sum_{l=1}^{p-1}\left(1 - \frac{l}{p}\right) a_p c_{p-l}$ $1 \leq l \leq p$, a_p are the AR coefficients
FD Features	
Mean of signal frequencies	$FMN_t = \sum_{j=1}^{M} f_j p_j / \sum_{j=1}^{M} p_j$ f_j is frequency of the spectrum at frequency bin j, p_j is the EMG power spectrum, m is the length of the bin.
Frequency ratio	$FR = \sum_{j=LLC}^{ULC} P_j / \sum_{j=LHC}^{UHC} P_j$ ULC and LLC are the upper and lower cut-off frequency of the low frequency band; UHC and LHC are those of the high frequency band, respectively.
TSCD/TFD Features	
Short-time Fourier transform	$STFT_t[l, m] = \sum_{n=1}^{L} \boldsymbol{x}_t[n] g[n-l] e^{-j2\pi mn/L}$ g is the window function and m denotes the frequency bins

extractor, and the extracted information are verified to be more representative than those obtained by hand-crafted features [14, 37]. In addition, it is observed that the first convolutional layer may serve for preprocessing to condition the signal for other layers [37].

To apply CNN for motion estimation, the sparse multi-channel sEMG in a segment should be reconstructed as image variable $\boldsymbol{X}_t$ whose size is $\mathrm{C} \times L \times 1$ or $1 \times L \times \mathrm{C}$, where C represents the number of sensor channels. In practice, these two formats represent the two-dimensional (2D) and one-dimensional (1D) sEMG images, respectively. Besides, when $\boldsymbol{X}_t$ is reconstructed from raw sEMG signals directly, it is named as the time-domain image. Otherwise, a spectrum image can be obtained by applying fast Fourier transform (FFT) on each channel of sEMG. In summary, four basic types of sEMG images can be utilised in myoelectric control, i.e. 2D time-domain image, 2D spectrum image, 1D time-domain image, and 1D spectrum image. Alternatively, sEMG can also be acquired from high-density matrix, and the 2D spatial format can be converted into instantaneous sEMG images that views like a heatmap [38]. The examples of some types of sEMG images are illustrated in Figure 2.1. Since $\boldsymbol{X}_t$ is normally a small-scale input for CNN, architectures based on the simple single-stream LeNet-5 [39] are mostly applied [40–42]. To further improve the learning capability, recent efforts have also been made to optimise architectures, such as a two-stage multi-stream CNN [38], an early-late fusion sub-networks [43], and a parallel multiple-scale convolution architecture [44], etc.

Apart from the spatial information, temporal dependencies of successive segments are also of significance in myoelectric control. This is because sEMG signals can be typically regarded as the time-series data during continuous muscle contractions. To this end, RNN and the variations, i.e. LSTM, gated recurrent units (GRU) and bidirectional RNN (Bi-RNN), have also been widely investigated in myoelectric control [45–48]. The structure of RNN is shown in Figure 2.2. Unlike standard feed-forward neural networks, RNN contains cycles that feed the network activation from a previous time-step to influence predictions at the current step. By using the recursive learning process, contextual information of adjacent

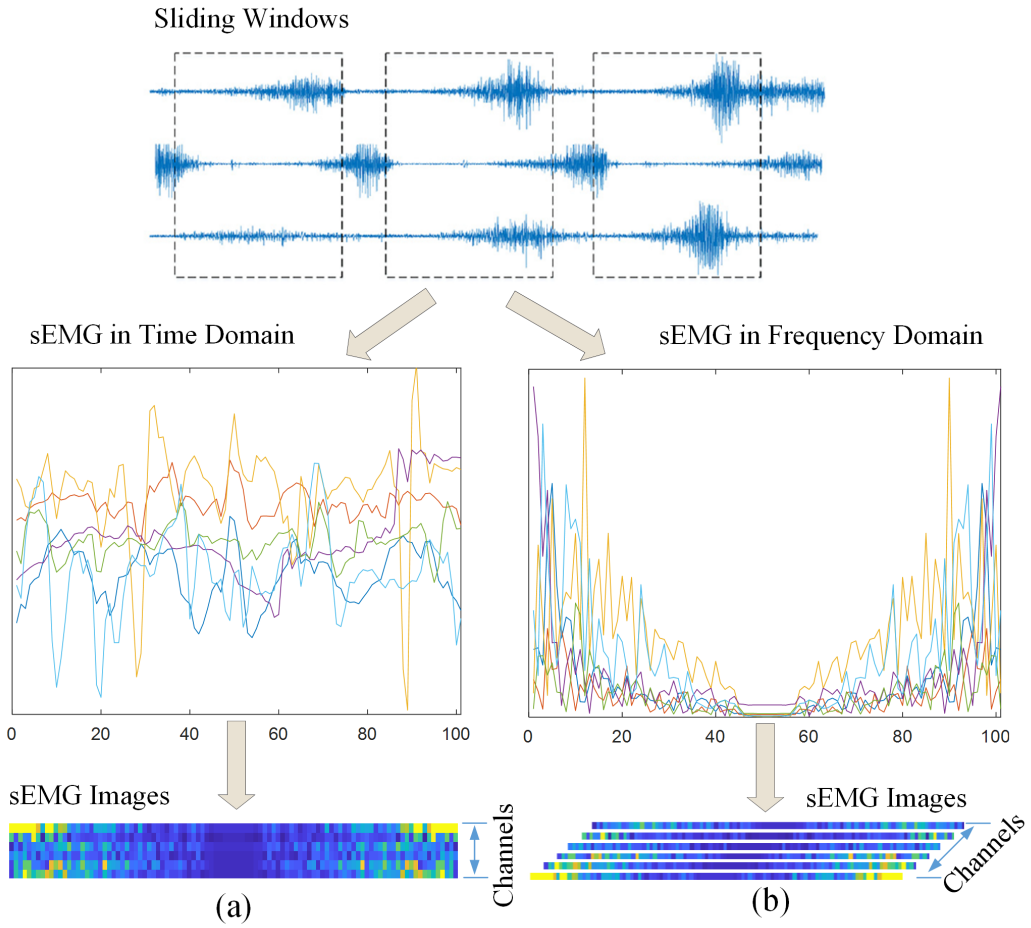


Figure 2.1: Examples of sEMG images that are constructed from sparse multi-channel signals for CNN-based motion estimation: (a) 2D time-domain image; (b) 1D spectrum image.

inputs can be better explored. As demonstrated in Figure 2.2, the unfolded RNN can be seen as very deep feed-forward networks in which all the layers share the same weights [16]. Therefore, one drawback of conventional RNN is that it is difficult to remember content very long due to gradients disappearing. To solve this question, LSTM contains the forget gate to determine the proportion of preceding information that should remain or be thrown away. This is also the way that human brains work. In fact, LSTM has become the most popular version of RNN in a variety of research fields.

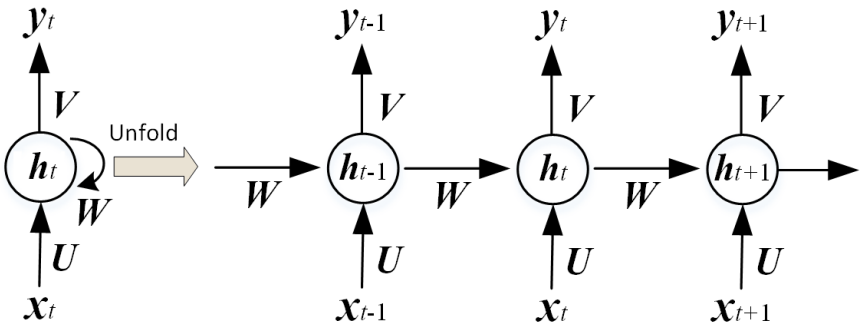


Figure 2.2: The structure of RNN and the unfolding in time of the computation. The current input $\boldsymbol{x}_t$ together with the state of the previous hidden layer $\boldsymbol{h}_{t-1}$ will be sent into the current hidden layer to compute the current output $\boldsymbol{y}_t$. In this way, $\boldsymbol{y}_t$ depends on all the previous $\boldsymbol{x}_{t'}$ $(t' \leq t)$.

Different from CNN and RNN, AE is an unsupervised technique that leverages neural networks for representation learning from unlabelled data. It typically consists of an encoder part to project sEMG features into the latent space (i.e. hidden vector), and a decoder to regenerate these features from the encoded data. By minimising the difference of original inputs and regenerated outputs, the non-linear relationships in sEMG features can be captured. A typical structure of AE can be found in Figure 2.3. The number of neurons in the hidden vector is usually less than that in the input layer, which will lead to a decline in data dimension. This process is similar to PCA, but AE can learn non-linear transformations by

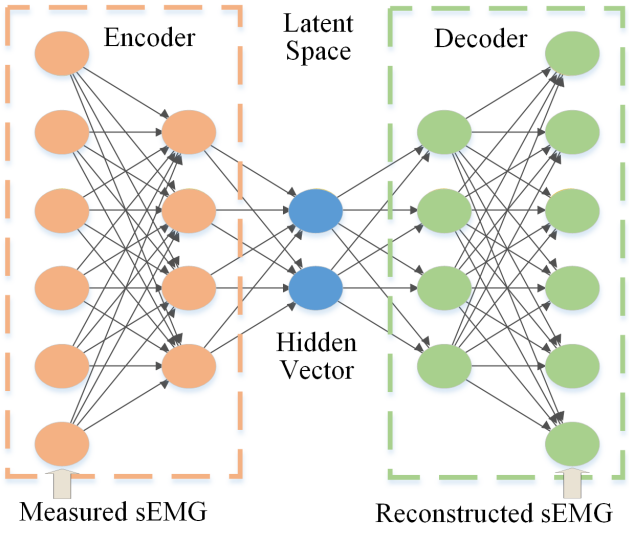


Figure 2.3: The structure of a general model of AE in myoelectric control.

using non-linear activation functions and multiple layers. After model training, the encoder is normally remained for feature extraction [49, 50], muscle synergy extraction [51, 52], and multi-DOFs online control [53]. For instance, Rehman *et al.* [49] applied stacked sparse AE (SSAE) to multi-day EMG recordings, with the output of encoder working as features for a softmax classifier. The classification results were shown to be competitive to both classical ML methods and CNN. More recently, the combination of CNN and AE, e.g. the convolutional AE, has also been proposed to enable a better feature learning process [54].

2.2.4 Discussion

The main difference between ML and DL can be summarised as how features are extracted from raw sEMG. As discussed above, a larger variety of hand-crafted features have been developed to provide temporal or/and spectral information of sEMG. Undoubtedly, the combination of hand-crafted features with different types of ML models, as well the associated channel selection and dimension reduction, can be greatly influential to ML performances but do have a high requirement on the expertise in both domain knowledge and experimental ex-

perience. By contrast, DL exploits multiple layers of neural networks to extract representative features from raw sEMG directly and automatically, until the final layer easily performs the mapping tasks. As a result of sufficient data, hardware improvements and algorithmic innovations, this procedure has rapidly evolved in the past decade [55], and a few encouraging attempts have been reported in previous literature. Nevertheless, these efforts were mainly made for gesture recognition in static scenarios, whereas investigations on decoding performances in more complicated tasks have not been conducted yet.

2.3 Post-processing to Improve Recognition Reliability

2.3.1 Problem Statement

In real-time scenarios, characteristics of sEMG signals can be significantly impacted by a variety of disturbances, including 1) electrophysiological changes (e.g. muscle fatigue) [56], 2) varying electrode-skin impedance due to perspiration or humidity [57], 3) external/measurement factors caused by electrode shift [58], and 4) user issues such as variations of contraction intensity, hand orientations and arm positions [3]. Due to the shift of sEMG patterns, performances of pre-trained ML/DL models can decay greatly, which may cause unintended movements and even risks to users. In this context, the improvement of recognition reliability is of vital importance to the user-safety in myoelectric systems. In this section, several representative disturbances that can impact the model robustness will be introduced. Then an overview will be provided with respect to post-processing techniques that are proposed to enhance the reliability of PR-based control scheme.

2.3.2 Disturbances in Practical Scenarios

Muscle Fatigue

The fatigue occurs to muscles that are being contracted with high strengths for a long time [59]. It can be quantified as a decline in the maximal force or power capacity of the muscle [60]. Due to the decrease of active motor units and the change of conduction velocity of muscle fibres [61], sEMG characteristics such as the time-domain amplitudes and some frequency-domain features can be significantly affected. For instances, Tkach *et al.* [62] compares power spectral density of EMG signals with and without fatigue in elbow flexion. As we can see, sEMG signals recorded during fatigue are observed to contain higher frequency components as compared with those recorded without fatigue. Accordingly, the estimation performances of ML/DL will be impacted, especially when some sensitive features are involved [62]. To this end, most studies on ML/DL based myoelectric control are elaborately designed to avoid the presence of fatigue, by limiting the duration of a training/testing session, controlling the contraction intensity, and providing enough breaks between trials. Although these settings can work in the laboratory environment, they become infeasible in real-time utilisation of myoelectric systems.

Electrode Shift

Electrode shift is another challenge that impacts the motion estimation in the long-term utilisation. It occurs when users re-wear the myoelectric control systems, such as the donning/doffing or re-positioning of the prosthetic socket, which results in the displacement of electrodes from the original position [63]. In fact, the existence of electrode shift is nearly inevitable since it is hard for users to attach all sensors, particularly the sparse multi-channel electrodes, to the fixed positions on target muscles. In addition, movements of limbs may also cause small shifts of electrodes [59]. Due to the significant changes of sEMG signals acquired in each channel, the feature space after displacement varies accordingly, which results in the great degradation of pre-trained models [64–66]. Besides, it

is also observed that displacement perpendicular to muscle fibres, i.e. the transversal shift such as the rotation of high-density sEMG matrix or armband-typed sensors, have worse impacts than longitudinal shifts [64, 66]. For instance, in a myoelectric system with four EMG channels and ten target gestures, a shift of 1 cm increases the classification error from roughly 5-20% (for longitudinal shift) and to 40% (for transversal shift) [64].

Arm Posture

In most studies, sEMG signals used to train and test a ML/DL model were recorded in a specific arm position. However, this setting is too ideal to match the practical use of myoelectric prosthesis, since the arm postures of users would unavoidably change even when they are performing hand movements [67]. As verified by numerous studies [68–70], sEMG recordings can differ from those that are measured from the same muscles but with different arm posture. The main reason for the variation is that, to keep the limb stabilisation or to counteract the effect of gravity, muscles will work with different effort for the same hand movement. Besides, the muscle geometry will also change accordingly, which contributes to varied myoelectric excitation [63]. For instance, Liu *et al.* [69] observed that sEMG patterns of different arm postures were shifted in the feature space, and performances of motion estimation decayed consequently. To this end, many researchers suggested to perform hand gestures in several pre-defined arm position, or even a dynamic motion of the arm in 3D space [71–73]. Nevertheless, to eliminate the negative effect of arm position change, more efforts are needed to develop suitable training strategies, robust features and adaptive classifiers.

2.3.3 Post-processing

Multi-window Smoothing

To apply ML/DL in myoelectric control, the sliding window method is normally utilised to extract sEMG signals into successive segments. Considering that adjacent windows of signals are likely to reflect the same motion, multi-window

smoothing approaches have been developed to smooth out noisy outputs. Of all related efforts, majority vote (MV) strategy is the most simple and prevalent one. It was firstly introduced by Englehart and Hudgins [74] to eliminate spurious misclassification in the unprocessed decision stream, and has been vastly applied in hand gesture recognition tasks [75–78]. In general, MV can be formulated as

$$\widehat{\boldsymbol{y}}_t \leftarrow \text{mode}\left(\widehat{\boldsymbol{y}}_{t-n}, \widehat{\boldsymbol{y}}_{t-n+1}, \cdots, \widehat{\boldsymbol{y}}_t, \cdots, \widehat{\boldsymbol{y}}_{t+m}\right). \tag{2.3}$$

where mode() returns the value that occurs most frequently in an array, and m/n determines the number of successive segments to be involved.

To summarise, MV presents a viable option to minimise misclassification by employing successive windows of signals to make final decision. In addition to conventional MV, some variations have also been investigated. For instance, Falk-Dahlin [79] developed three modified MV to work with the simultaneous control system. Zhai *et al.* [80] presented a MV-based label updating mechanism for CNN classifier. To increase the total number of votes for a given data stream, Wahid *et al.* [81] developed a multi-window majority voting (MWMV) strategy composed of windows with varying lengths. Experiment results verified that the accuracy differences between MWMV and MV were significant for all tested ML algorithms when utilising 30ms and 50ms window size. Apart from implementations in non-recurrent ML/DL models, Simao *et al.* [82] applied MV for a LSTM classifier to remove the false positive results of time steps that cover the transition period between gestures.

However, MV-based approaches operate on the decision stream directly but do not consider the actual probabilities of misclassification. To this end, a Bayesian fusion (BF) approach was presented based on the Bayesian rule to make the post-processing more accurately [78]. Specifically, BF utilises a number of posterior probabilities in a series of sliding windows to calculate the final probability of each class. The class with highest probability was then selected as the final output. This process can be formulated by Eq. (2.4). In addition, weighting factors were given to each sliding window, such that higher priorities are assigned for the current decisions and lower priorities for previous ones. Experiment results demonstrated that BF outperformed MV significantly in healthy subjects [78]

and competitive performances in amputees [83].

$$p\left(\widehat{\boldsymbol{y}}_t \mid \boldsymbol{x}_{t-n}, \cdots, \boldsymbol{x}_t\right) = \Delta \prod_{i=t-n}^{t} p\left(\widehat{\boldsymbol{y}}_i \mid \boldsymbol{x}_i\right), \tag{2.4}$$

where Δ is a constant to normalise the probability distribution.

Another well-known method is the Decision-Based Velocity Ramp (DVR) introduced by Simon *et al.* [84]. Different from MV and BF, DVR boosts the control reliability by changing the speed of outputted movements rather than the predictions themselves. In specific, DVR forces a new movement to be performed very slowly, and increases the speed as predictions of the same movement are made consecutively. To achieve this goal, counters are applied for each movement to track the speeds that should be performed. Once a prediction has been made by the classifier, the counter of the predicted movement is increased whilst other counters are decreased. Only the movement that is currently predicted can be output, and the speed V_{out} is calculated by

$$V_{\text{out}} = \frac{C_g}{RL} \cdot V_{\text{in}}, \tag{2.5}$$

where C_g is the counter associated with g^{th} movement, RL is the ramp length defined by the experimenter, and V_{in} is a desired speed that can be calculated based on muscle intensity. In Eq. (2.5) $C_g \in [0, RL]$. As reported by [84], a superiority of DVR to MV in classification accuracy was observed, and no delay was introduced to myoelectric control since every prediction was outputted. This is another advantage over MV and BF. To this end, DVR has gained popularity in myoelectric control [85–88].

Confidence Estimation

To enhance the usability of PR approaches in real-time scenarios, it is desirable to alleviate the negative influence of misclassification based on the analysis of model uncertainty. As illustrated by Figure 2.4, a practical solution in myoelectric control is to estimate the confidence of classification results, and a rejection process or smoothing operation will be further performed to cope with the uncertain decisions. In some early efforts, Fukuda *et al.* [89, 90] suggested to calculate

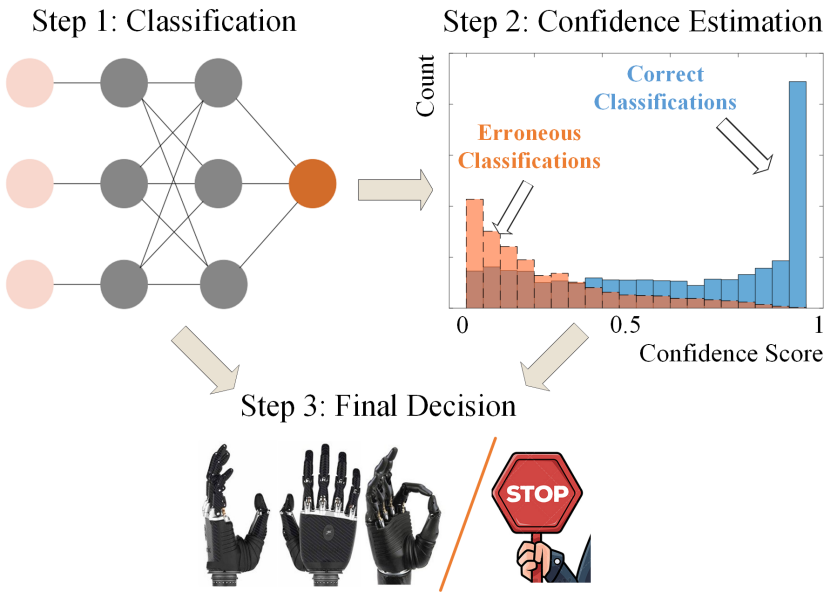


Figure 2.4: A typical workflow of confidence estimation for PR-based upper-limb motion estimation. To enhance the model reliability, confidences of classification results are estimated and then processed by a rejection or smoothing operator to obtain the final decision for myoelectric control.

the entropy of a log-linearised Gaussian mixture network to indicate the risk of incorrect discrimination. If the entropy exceeded a pre-specified threshold, meaning that the network output is ambiguous, the associated motor control should be suspended. This idea was further expanded by Sensinger *et al.* [91] to measure classification confidence, where entropy was calculated as a function of the probability that a feature set belonged in each class. Therefore, a decision obtained low entropy, i.e. high confidence, if only one class had a high probability.

Another representative study for confidence-based rejection was presented by Scheme *et al.* [92]. The authors linearised and normalised the log probability outputs of an linear discriminant analysis (LDA) classifier as the confidence metric, and estimations were regarded as no-movement when the associated confidences were below a given threshold. Differently, Amsüss *et al.* [93] applied a multilayer perceptron (MLP) to indicate the confidence of a LDA classifier and then

facilitate corrections on wrong classifications using past results. In specific, LDA outputs were relabelled as $+1$ if they were correctly classified and -1 for erroneous ones. Meanwhile, the maximum likelihood of LDA and the mean global muscle activity of the forearm worked as features for MLP-based confidence estimation. To have a deeper perspective of confidence-based rejection, Scheme *et al.* [94] examined the confidence characteristics of several conventional classifiers, and observed that low confidence were correlated with a decrease in classification accuracy. Moreover, they found that support vector machine (SVM), which allowed for more complex boundaries than other classifiers, provided a more stable rejection-to-threshold relationship during dynamic usage. Based on this finding, Robertson *et al.* [18] further investigated the range of rejection thresholds for optimal usability of SVM in real-time control.

2.3.4 Discussion

As discussed above, performances of PR-based control can be negatively affected by various real-time disturbances. In this context, even a well-trained model can result in unintended estimation, causing additional operations, cognitive burdens, and even unacceptable risks [18]. To improve the reliability of prosthetic control, a variety of post-processing methods have been proposed to reduce the potential estimation errors based on previous outputs or extra information. Herein, multi-window smoothing and confidence estimation were mainly introduced. To be specific, the former approaches attempt to reduce the spurious estimation errors by exploiting information of consecutive sliding windows, and the latter quantify model confidence such that unconfident estimations can be detected and processed. In particular, although ML/DL approaches can improve the functionality of myoelectric control, these methods suffer a lot from the lack of interpretability. In this context, confidence estimation can help to have a deeper insight of these black boxes, providing HMI the ability to not only decide what to do but also if it should be done.

2.4 Regression Scheme to Enable SPC

2.4.1 Problem Statement

In the past decades, PR approaches have been extensively explored to improve the intuitiveness and functionality of myoelectric control. Despite the enormous advances in academia and some success in commercial products [13, 95, 96], there are intrinsic drawbacks related to the unnatural way of performing tasks. Firstly, PR scheme is limited to the identification of discrete states of movements, since only one pre-defined class could be predicted at a time [97]. In order to perform coordinated tasks such as the cursor control or reaching, each individual function has to be selected sequentially. In addition, the lack of proportionality in the selected motions also limits PR approaches, since the control generally only requires on/off mode of operation (a class is either active or not) [20]. By contrast, the control of neuromuscular system towards upper-limb movements are mainly based on the simultaneous and proportional activation of multiple DOFs, which differs a lot from the sequential on/off control mode provided by the PR-based scheme [2].

Unlike classification models that match sEMG data with certain categories of gestures or movements, regression-based approaches aim to achieve a proportional and simultaneous control (SPC) by establishing a continuous mapping between sEMG and joint kinetics/kinematics [1]. In particular, when a regression model is learned to produce proportional outputs for each DOF, simultaneous control of multi-DOFs can be obtained by linear summation of single-DOF activations [98]. The concept of regression-based SPC is well illustrated in Figure 2.5. As we can see, a proportional control on each DOF can be enabled independently. Meanwhile, the coordinated task, i.e. target 3, can be achieved by co-current activations of two basic DOFs. In this way, a more fluent and natural control of the cursor movement can be facilitated.

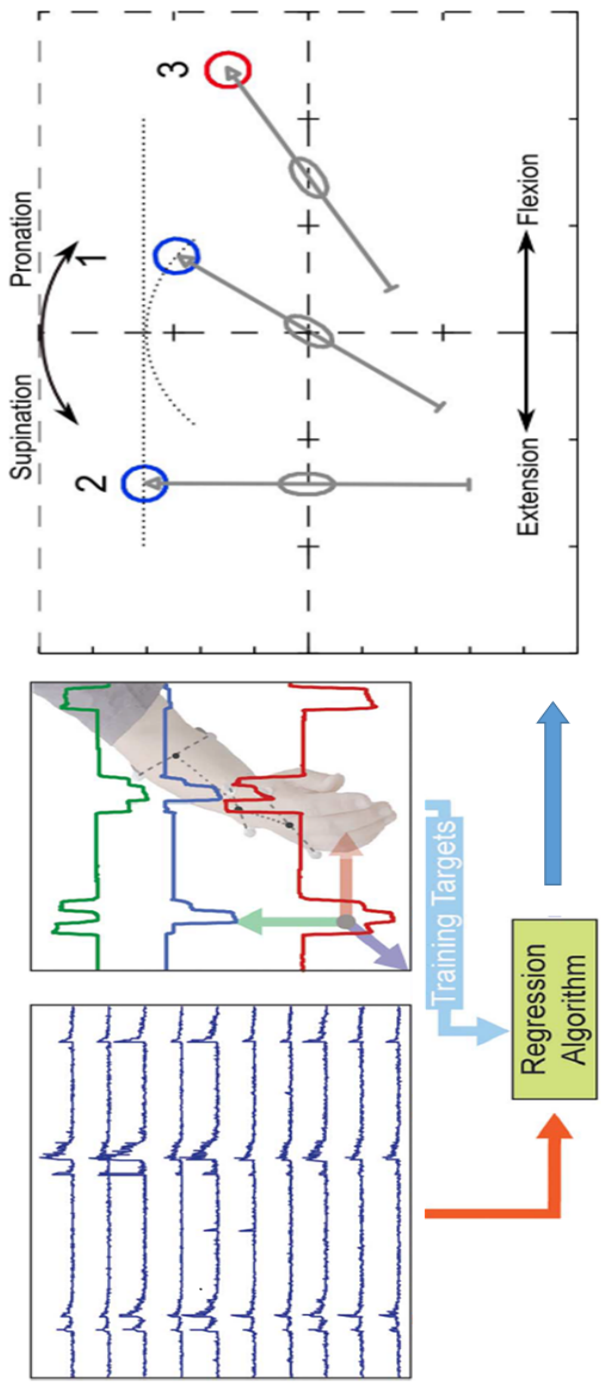


Figure 2.5: An example of regression-based approaches for the multi-DOFs SPC of the wrist. Targets 1 and 2, i.e. the blue circles, are examples of continuous movements requiring separate activation of DOF1 and DOF2, respectively. Differently, target 3 (red circle) can be reached by simultaneously activating both DOFs. This figure is adapted partially from [1] and [2].

2.4.2 Experiment Protocols

To build a regression model, a set of training data that contain both sEMG and associated movement labels are normally required. For healthy participants, the joint movements can be measured via sensors attached on the arm. For instance, a vision-based motion tracking system is often utilised to calculate wrist angles for corresponding sEMG signals [99–101]. These systems is composed of 6 to 12 infrared digital video cameras and can provide high accuracy of motion measurement in 3 dimensional space. However, the optical devices require a static workspace and heavy financial cost. Therefore, more economical and flexible tracking systems, such as IMU [102, 103] and data glove [104, 105], can be utilised alternatively. As for joint forces or torques, custom platforms integrated with load cells are usually developed for data labelling [106, 107].

In terms of experiments with amputees involved, it is not possible to measure kinematics of the missing limb. To this end, the mirror training strategy has been developed, in which the kinematics are measured from the able hand (assuming that the amputation is unilateral) and EMG is measured from the contralateral side with limb deficiency [104, 105, 108, 109]. Another approach to obtain training labels for amputees, particularly those with bilateral amputation, is to use visual cues as movement targets [20], based on which the amputees can adjust the pattern and intention of muscle contractions. For instance, in Ref [110] participants were asked to elicit the wrist movement with an intensity proportional to a progress bar on the screen. For combined motions, they then increased or decreased both DOFs simultaneously following the progress bar.

2.4.3 Estimation Models

To enhance continuous estimation of joint kinematics/kinetics, numerous efforts have been made to develop advanced mapping techniques. Based on the hypothesis of muscle synergies, Jiang *et al.* utilised the classic non-negative matrix factorisation (NMF) to extract sEMG signals into wrist kinetics [106]. Choi *et al.* then employed a non-negative muscle synergy matrix to map muscle activ-

ities in a real-time cursor control [111]. In general, conventional factorisation methods need to identify the muscle synergies and weighting functions associated with single-DOF separately. To improve experiment efficiency and model accuracy, sparseness constraints were further considered in NMF matrix, so that a quasi-unsupervised factorisation approach could be established [98].

Alternative to synergy-based schemes, a more intuitive and efficient scheme is to apply supervised ML/DL models. In previous literature, a variety of linear and nonlinear algorithms, including linear regression (LR), MLP, kernel ridge regression (KRR), support vector regression (SVR), random forest (RF), and Gaussian process regression (GPR) etc., have been extensively investigated [102, 105, 112, 113]. According to an offline test on LR, MLP and NMF, the first two ML models surpassed the semi-unsupervised NMF algorithm by a large margin [114]. This result can be explained by a consensus that, with sufficient labelled training data, supervised learning algorithms usually outperform unsupervised ones for the same ML task [115]. Besides, a study on real-time prosthetic control [116] indicated that LR based SPC could be superior to conventional control strategies in clinic tests. Apart from developments in mapping functions, considerable efforts have also been given to channel selection [35, 107], training paradigms [117], man-machine co-adaptive learning [118], and control robustness against real-time disturbances such as doing/doffing and arm position effect [119].

2.4.4 Discussion

Despite the above advancements in SPC, it is still widely accepted that only limited number of DOFs can be controlled reliably using regression techniques [120], and that the estimation is often not accurate enough to support precise movements [114]. For example, in a task to rotate the wrist to pour liquid into a cup, regression-based SPC was observed to produce inadvertent combined movements, increasing the risk of erroneous drops or squeezing the bottle [120]. To this end, DL has become a hot-spot in recent literature. As discussed in Section 2.1, the main perspective of DL is to obtain more informative sEMG features via representation learning. Nevertheless, the research on DL-based regression is still

at the very beginning, particularly compared with studies on DL in PR-based scheme. More efforts are thereby required for further improvement of regression accuracy and robustness in SPC.

2.5 Transfer Learning for Fast Model Recalibration

2.5.1 Problem Statement

Classical ML/DL assumes that training and testing data stem from the same underlying distribution [121]. However, as discussed in Section 2.2, this assumption is often violated by a variety of practical disturbances, which can be referred as domain shift in ML/DL applications. Furthermore, as a type of bio-electricity, sEMG is user-specific and time-variant, resulting in a poor generalisation of pre-trained models among subjects or in the long-term utilisation. For instance, Waris *et al.* [22] studied the distribution of sEMG patterns collected in different days, and found that the discrepancy of feature space increased over time, providing a clear indication of the high variability in sEMG signals. Besides, due to the user-specific nature in physiological, anatomical and biochemical characteristics of muscles, this variation is even more serious when DL models are implemented in the inter-subject circumstance. Although retraining a new model that is specific to the targeted scenario can help to address this issue, considerable number of data need to be recollected and relabelled, which is inconvenient or even infeasible in practical applications.

To this end, transfer learning (TL) has been explored in upper-limb motion estimation to reduce domain shift impacts. As illustrated in Figure 2.6, TL utilises the knowledge learned in the source domain to promote the learning process in a target domain where sufficient labelled data are unavailable. Mathematically, given a source domain D^S and learning task T^S, while a target domain D^T and learning task T^T, TL improves $f_{\boldsymbol{\theta}}(\bullet)$ in D^T using the knowledge in D^S,

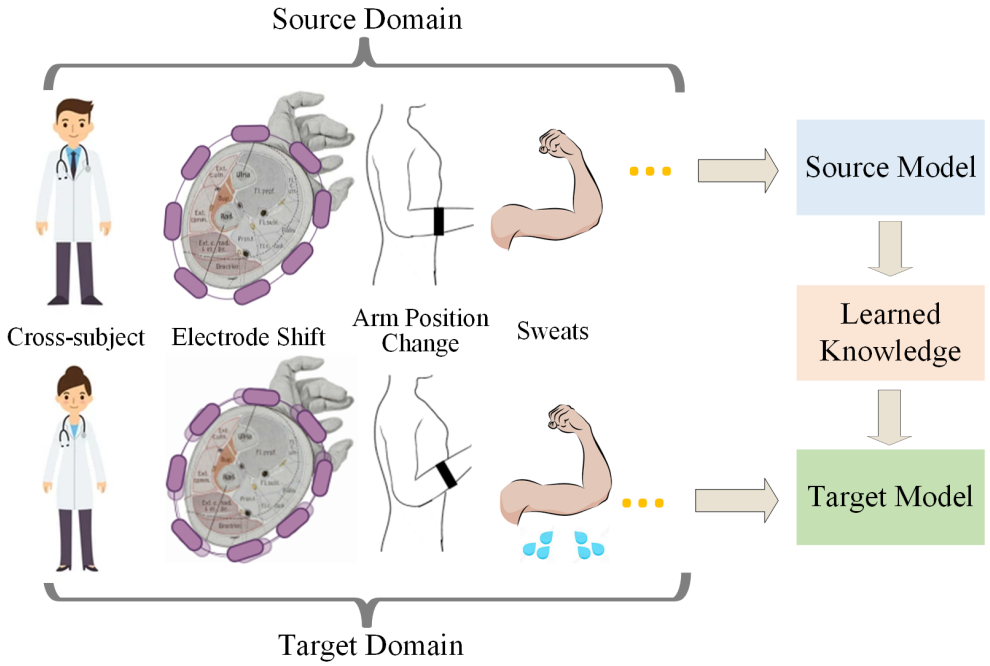


Figure 2.6: The basic process of TL in myoelectric control. The arm position change and electrode shift are adapted from [3] and [4].

where $D^S \neq D^T$ and/or $T^S \neq T^T$. As summarised by [122], when $D^S \neq D^T$ but $T^S = T^T$, the TL tasks can be narrowed as domain adaptation (DA). In this context, a label-specified but domain-invariant subspace is desired to be extracted from original feature spaces. Besides, a TL task $\left\langle D^S, T^S, D^T, T^T, f_{\boldsymbol{\theta}}(\bullet) \right\rangle$ is referred as conventional TL if $f_{\boldsymbol{\theta}}(\bullet)$ is a traditional ML model, or deep TL when $f_{\boldsymbol{\theta}}(\bullet)$ reflects a deep neural network.

2.5.2 Conventional TL

The foundation of a positive TL/DA is that D^S ($D^S \neq D^T$) can provide useful information for the estimation tasks in D^T. In another words, there are supposed to be inherent user-independent properties buried in sEMG signals. In this context, a preliminary study was firstly presented by Saponas *et al.* [24], verifying that pooling data from multiple users yielded a classification result higher

than chances for a novel user. This observation indicates the possibility to build cross-user algorithms. Orabona's *et al.* [123] then proposed an adaptation process to complete the model adaptation among different users, in which the best matched model was modified from a pool of stored datasets to fit a new subject. Chattopadhyay *et al.* [124] proposed a multi-source DA methodology based on predominantly conditional probability differences between the source and target distributions, and improved the subject independent classification accuracy by 5%. Matsubara *et al.* [125] proposed a projection approach based on bi-linear model composed of user-dependent factors and motion-dependent factors, where the latter could be further used as user-independent features for a motion classifier. More recently, Zhang *et al.* [126] introduced a dual layer transfer learning (dualTL) framework composed of two layers. The first layer leveraged the correlations of sEMG among users to label target gestures whose confidence was sufficiently high. In the second layer, other gestures of the testing user were labelled according to the consistencies of sEMG signals from himself/herself. Jiang *et al.* [21] proposed a correlation based data weighting (COR-W) method for model calibration. The domain shift level between source subjects and the target subject was firstly evaluated via correlation alignment (CORAL), then a weighted least squares was employed to develop a calibrated EMG torque model based on previous training trials.

To address model degradation in the long-term utilisation, Chen *et al.* [127] extended linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) to the self-enhancing versions using additional knowledge from the classified data in testing set. In this way, model parameters can be updated in the following days continuously. Liu *et al.* [128] proposed a DA algorithm for both LDA and a polynomial classifier, where the new model automatically reused pre-trained models as inputs. Following this study, Zhu *et al.* [129] introduced a cascaded adaptation scheme including a DA component and a self-enhancing component. In specific, the first part reused the previous model to reduce the re-training data, and the second updated model parameters according to new data samples and corresponding recognised labels. To further promote the rapid

re-calibration, Cosima *et al.* [130] investigated a TL approach based on the generalised matrix learning vector quantisation (GMLVQ) classifier, such that only a very small amount of training data was required. Benjamin *et al.* [4] presented an expectation maximisation (EM) algorithm which learned a linear transfer function between the target and source space, thereby samples in the target space could be classified correctly by the source space classifier after data mapping. By weighting the importance of training samples in the prediction of testing outputs, Vidovic *et al.* [131], Kanoga *et al.* [132] and Jung *et al.* [133] investigated the co-variate shift adaptation methods to calibrate parameters of conventional classifiers such as LDA or GPR.

Among the variety of efforts in conventional TL, canonical correlation analysis (CCA), a method with well-established theory and low computational complexity, has become a prevalent framework [134–138]. Theoretically, CCA is able to find linear transformations and extract pairs of canonical variables (CVs) to reflect the correlations between two sets [135]. In myoelectric control, CCA attempts to extract inherent user-independent properties of sEMG signals generated from the same motions under different conditions [137]. Khushaba *et al.* [134] firstly exploited CCA to promote multi-user myoelectric interfaces. In specific, a unified-style space was built by correlating feature matrices of each subject with the matrix of an expert subject. Then a classifier was trained using the CCA-based mapping. Following this idea, Cheng *et al.* [135] used CCA to extract the homogeneity of sEMG signals generated by the same gestures, such that the adverse effect of arm positions can be suppressed. Fan *et al.* [136] applied CCA to eliminate the reduction of classification accuracy caused by electrode shift. To further minimise the discrepancies in probability distribution functions between transformed features matrices of training and testing sets, Xue *et al.* [137] integrate CCA with optimal transport (OT) which can transport the source samples onto the distribution of target domain.

2.5.3 Deep TL

As aforementioned, it now becomes a trend to employ DL in myoelectric control for deriving representative features from raw sEMG. Therefore, deep TL has gained growing interests in recent years. Based on previous studies, deep TL in myoelectric control can be divided into two main categories: network-based approaches and feature-based approaches. Figure 2.7 demonstrates the typical structure of these two approaches. In the structure, a network is firstly trained in D^T with sufficient labelled data. Then, partial of this network are maintained by freezing the weights, and the non-frozen part is updated using either labelled or unlabelled target data. Differently, feature-based structure attempts to obtain domain-invariant features via domain alignment. To be specific, front-layers of the network extract features from two domains for domain loss calculation, aiming to reduce the mismatch of feature distributions in the latent space.

As a simple implementation of the network-based deep TL, fine-tuning (FT) has become the most prevalent approach in myoelectric control. A representative effort was presented by Wang *et al.* [139]. The authors utilised FT to enhance the training of RCNN module of a multi-modal DL structure, where D^S data came from a public sEMG database whilst D^T was composed of multimodal data collected from experiments. Kim *et al.* [140] proposed a subject-transfer framework by fine-tuning the supportive CNN classifiers, where the estimation model was examined to be more robust in terms of intra-user variability. Ameri *et al.* [141] employed FT to reduce the electrode shift impact on CNN performances. Experiments in both hand gesture recognition and wrist kinematics estimation verified that FT outperformed a simple aggregation of pre-shift and post-shift sets. Different from above efforts which utilised the sparse multi-channel sEMG signals, the generalisation of spatial features extracted from high-density sEMG matrix was also verified for both new subjects and new gestures through FT [142]. In addition, Demir *et al.* [143] applied the AlexNet pre-trained in computer vision tasks to fine-tune the sEMG images. Bird *et al.* [144] investigated the performances of FT for the transfer learning between sEMG and EEG signals, and observed that the knowledge could be successfully transferred between two

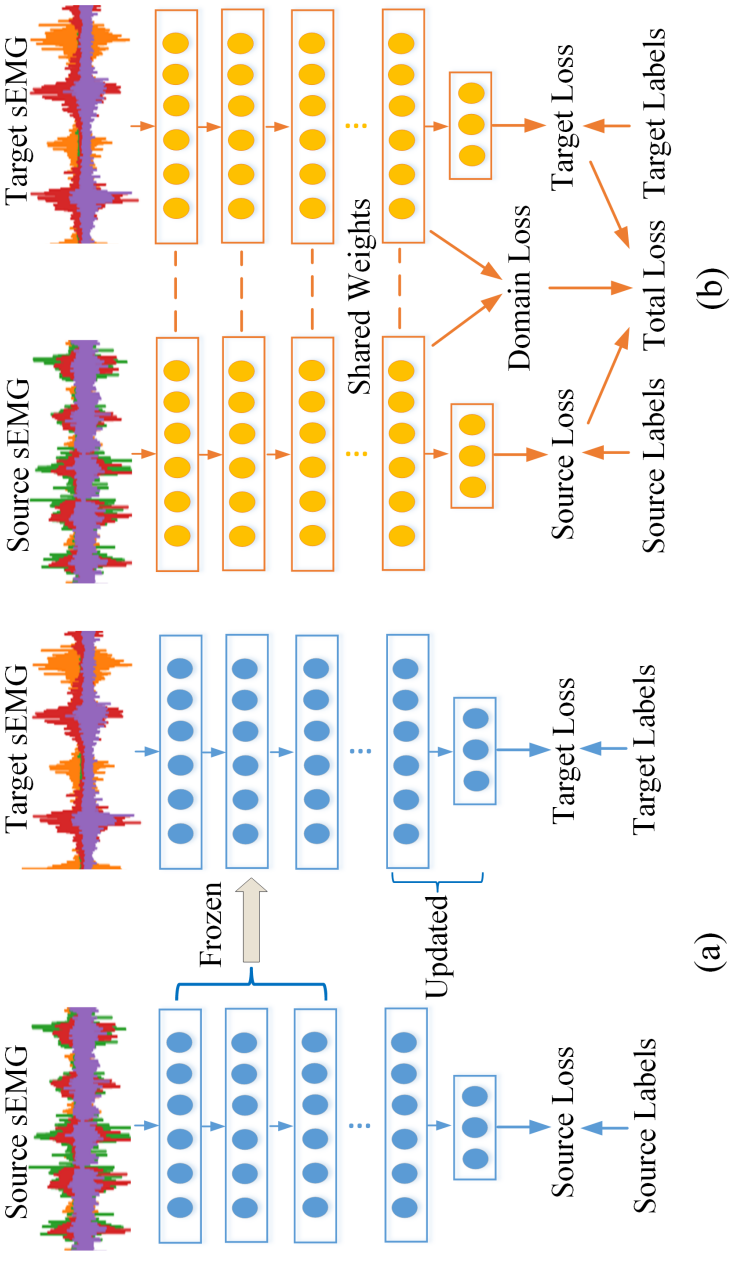


Figure 2.7: The typical structures of deep TL in upper-limb myoelectric control: (a) network-based deep TL; (b) feature-based deep TL. Note that unsupervised TL can be applied when the target labels/loss are unavailable.

modalities. To further enhance the effectiveness FT, Chen *et al.* [145] constructed the source gesture set composed of various states of elbow, wrist and finger joints. They observed that even if a new gesture was not included in the source set, a good recognition accuracy could be obtained as long the activation modes of muscles were covered.

Apart from conventional FT, several other efforts have been conducted to exploit network-based approaches. For example, Du *et al.* [41] presented a multi-stream AdaBN method to boost the inter-session performances of CNN. In the recognition phase, the adaptation process was performed by updating the statistics of batch normalisation with unlabelled calibration data. Côté-Allard *et al.* [146, 147] applied the Progressive Neural Networks (PNN) to decrease the training burden in hand gesture recognition. In this framework, the pre-training source network was frozen, and a new network with random initialisation was connected with source network using merging layers. Ketykó *et al.* [148] proposed a RNN-based two-stage framework which consists of a linear DA layer and a sequence classifier. In the adaptation stage, weights of the sequence classifier were frozen in their pre-trained state, and the DA layer weights were trained on the target data set using a supervised loss function.

For feature-based deep TL, weights of the network are updated by learning information from both source and target domain simultaneously, and DA takes place by aligning feature distributions of different domains in the latent space. Inspired by the success of Domain-Adversarial Neural Networks (DANN) [149], Côté-Allard *et al.* [37] presented the Adaptive DANN (ADANN) for cross-subject training. This objective was achieved by adding a domain classification head to a conventional CNN. During back-propagation, this operation learned to discriminate source and target domains via a gradient reversal process that forced the feature distributions over domains to be similar. Using a self-calibration strategy, the effectiveness of ADANN was then validated in the presence of confounding factors including inter-session and across-day variations [150]. In another following work, Campbell *et al.* [138] further tested ADANN in the cross-subject classification by requiring minimal training data from an end-user. Compared

with the state-of-the-art CCA methods [134, 135], ADANN was verified to be more effective for alleviating the training burden.

2.5.4 Discussion

In the past decade, both CTL and DTL have been widely investigated to address domain shift impacts in the cross-subject scenario and long-term utilisation. Based on model structures, DTL can be categorised into network-based methods and feature-based ones. Actually, as mentioned in Figure 2.7, there is another categorisation for both CTL and DTL, i.e. the supervised TL (STL) and unsupervised TL (UTL). In STL a small amount of labelled data are present in D^T, but these data are usually insufficient to train a new model from scratch. By contrast, sufficient but unlabelled D^T new data are available in UTL. According to related studies on both conventional TL [91, 137] and deep TL [141], STL usually perform significantly better than UTL. Apart from a lower accuracy, another potential drawback of UTL methods is that they usually require all exemplars in D^T to be included in the calibration process, resulting in a much larger computational load than STL. Therefore, the majority of efforts in DTL is still focusing on STL, e.g. FT [139–145] to achieve a fast and effective model recalibration.

2.6 Summary

In this chapter, a survey of previous works on ML/DL based upper-limb motion estimation is presented, with focuses on 1) differences between feature engineering and feature learning, 2) some of the main challenges that limit utilisation of ML/DL in myoelectric system, and 3) recent efforts to cope with those limitations. In Section 2.1, an overview on a variety of efforts in feature engineering is provided. In addition, feature learning achieved by CNN, RNN, and AE is also summarised. Section 2.2-2.4 review the achievements in post-processing to enhance reliability of movement/gesture recognition, regression-based joint kinematics estimation to enable SPC, and transfer learning to boost model robust-

ness and adaptation in cross-subjects or cross-days scenarios.

From this survey it can be indicated that DL, which has shown great potentials in the exploitation of sEMG patterns from raw data, can be more effective than ML in myoelectric control. However, it is also summarised that current progresses of DL are still much inadequate to overcome existed challenges in real-time applications. To be specific, experiment results presented by related literature are mainly examined under simple and static laboratory settings, whereas usability in practical scenarios has not been fully considered. In addition, most advances are designed for PR-scheme that can not well support the natural control strategy of upper-limbs. Therefore, this dissertation aims to conduct a study on DL-based motion estimation, particularly working on 1) confidence assessment to improve recognition reliability in practical scenarios, 2) regression approaches to decode joint kinematics from sEMG accurately, and 3) DTL to eliminate domain shift impacts. In the following chapters, our contributions towards above objectives will be introduced elaborately.

Chapter 3

CNN Confidence Estimation for Rejection-based Hand Gesture Classification

To improve the intuitiveness and functionality of myoelectric control, PR approaches have been extensively applied to decode user movement intentions from sEMG. However, as introduced in Section 2.3, the classification accuracy tends to degrade significantly when hand movements are performed in the daily living environment, mainly due to a variety of practical disturbances, such as muscle fatigue, electrode shift, arm position changes, etc. that result in significant variations between training and testing data. On the other hand, control reliability is critical to myoelectric systems since an unintended activation of the actuator may result in unacceptable risks to users.

Therefore, this chapter focuses on improving the reliability of DL techniques in PR-based control scheme. Specifically, a confidence estimation model is established to generate confidence scores based on posterior probabilities of CNN, and an objective function is designed to train the parameters of this model. In addition, a comprehensive metric that combines the true acceptance rate and the true rejection rate is proposed to evaluate the rejection performance of confid-

ence scores, so that the trade-off between system security and control lag could be fully considered. The effectiveness of our method is verified using data from public databases and our online platform.

3.1 Introduction

3.1.1 Motivation

Similar to many other safety-critical applications of ML/DL such as medical diagnosis and autonomous driving, myoelectric systems have high requirements on the reliability of prediction models. Currently most protocols simply assume that clean dataset are available for model training/testing. However, this assumption barely holds in practical scenarios. In fact, most classifier could only output one of the predefined gestures even though sEMG inputs have varied dramatically from training samples [18]. Moreover, even small perturbations to the input of classifiers can result in producing erroneous and over-confident predictions [151]. This issue can cause large uncertainties in classification as well as meaningless or unwanted outcomes. Therefore, the reliability of PR approaches is still significantly limited.

To this end, confidence estimation is now being investigated for quantitative evaluation of classification uncertainties. A rejection process can be cascaded with the classifier to refuse unconfident classification results, thus improving the reliability of myoelectric control systems by reducing erroneous movements. Numerous confidence estimation methods have been proposed in the past decade. For instance, Scheme *et al.* [92] calculated the maximum posterior probability of LDA as the confidence metric. Estimated hand motions would be reverted to no movement when the associated confidence was below a given rejection threshold. Sebastian *et al.* [93] combined the maximum posterior probability of LDA and RMS of sEMG as confidence features, based on which a cascaded MLP was trained to detect potentially erroneous decisions of LDA. Scheme *et al.* [94] examined the confidence characteristics of several conventional classifiers, whilst Robertson *et*

al. [18] verified the optimal rejection threshold for myoelectric control driven by SVM. However, these studies exploited mainly the confidence characteristics of traditional machine learning ML methods, where both the classification accuracy and rejection performances depend heavily on the design and selection of hand-crafted features; therefore, it is more desirable to develop learning algorithms that can extract representative features from raw data [110].

Currently, DL techniques, particularly CNN, are becoming popular in hand gesture recognition due to their strong capability of deriving data-dependent features automatically from sEMG [152], and better performance of CNN over traditional ML methods has been reported in previous studies [38, 43, 44, 80, 153, 154]. In this context, many researchers have started to link the class probability distribution to the confidence of CNN classification accuracy. For example, Ranjan *et al.* [155] predicted the task labels through the HyperFace network and recognised regions as faces when the maximum probability was above a certain threshold. Zhang *et al.* [156] utilised the probability distribution of CNN as its confidence feature and designed a decision fusion rule for remotely sensed image classification. Wang *et al.* [157] proposed the "I Don't Know" (IDK) prediction cascades framework leveraging the entropy of CNN likelihoods. Wan *et al.* [158] designed a Confidence Network (ConfNet) on the basis of probability distribution to generate confidence features and evaluate classification correctness. However, all these studies were conducted for computer vision tasks. To our best knowledge, the CNN confidence estimation and rejection analysis have yet to be investigated in myoelectric control.

3.1.2 Contribution

In this chapter, a novel approach is proposed to estimate the probability of correctness for each output of the classifier. The main contributions of the chapter include: 1) a confidence estimation model established to generate confidence scores (ConfScore) based on posterior probabilities of CNN, and an objective function designed to train parameters on this model; and 2) a comprehensive metric which combines the true acceptance rate and the true rejection rate proposed to evaluate

rejection performances so that the trade-off between system security and control lag could be fully considered. Effectiveness of ConfScore was verified using data from public databases and our online platform. Experimental results illustrate that ConfScore can better reflect the correctness of CNN classifications than traditional confidence features, i.e., maximum posterior probability and entropy of the probability vector. Moreover, the rejection performances are observed to be less sensitive to variations of rejection thresholds.

3.1.3 Chapter Organisation

The reminder of the chapter is organised as follows. Section 3.2 firstly introduces the framework of the confidence-based rejection for hand gesture recognition. It then presents the proposed CNN classifier, the confidence estimation model, the rejection rule, and the comprehensive evaluation metric. Section 3.3 introduces the setups of public databases and the online experiment. Section 3.4 demonstrates the experiment results. Section 3.5 presents the analysis of ConfScore in both confidence estimation and rejection evaluation. Conclusions are drawn in Section 3.6.

3.2 Methodology

3.2.1 CNN-based Confidence Estimation and Rejection

As illustrated in Figure 3.1, when a new classification is made by CNN, the posterior probability vector $\boldsymbol{p} = [p_1, p_2, \cdots p_m]$ is produced in the softmax layer, where m represents the number of hand gestures to be classified. Utilising this probability vector, the confidence estimation model can generate confidence scores to indicate the probability of correctness for each classification. Based on confidence scores, a threshold-based rejection process can be implemented to decide either to accept the estimated class or to revert the unconfident prediction to a no motion state. This rejection rule can be regarded as a flexible binary classi-

fier cascaded to CNN [157] and has been widely adopted in myoelectric control systems [18, 92, 94] and computer vision tasks [155, 157–159].

3.2.2 CNN Classifier

Since CNN is a neural network originally designed for processing data in the form of multiple arrays such as images, a matrix $\boldsymbol{X}_t$ is firstly constructed from sEMG signals. As aforementioned in Section 2.2.3, a sliding window method is utilised to obtain $\boldsymbol{X}_t$ from a segment of multi-channel signals. Herein, $\boldsymbol{X}_t$ is designed as a $1 \times \mathrm{L} \times \mathrm{C}$ matrix, where L denotes the window length and C represents the number of sensor channels. This 1D multi-channel format [160] is utilised since spatial correlations of all sEMG channels can be efficiently exploited via convolution operations. The sEMG matrix $\boldsymbol{X}_t$ is finally obtained by applying fast Fourier transform (FFT) to signals of each channel, as the spectrum of sEMG is observed to be less noisy and thereby more distinguishable than sEMG data in the temporal domain.

Since $\boldsymbol{X}_t$ is normally a small-scale input for CNN, architectures based on LeNet-5 [39] are still dominant in myoelectric systems [40–42]. In this study, a single stream CNN is adopted for the trade-off between classification accuracy and computational efficiency. As illustrated in Table 3.1, the presented CNN consists of two convolutional layers, one fully connected layer and a softmax layer. After each convolutional layer, a batch normalisation layer (for model robustness by reducing covariate shifts in intermediate representations after convolutional operations [161]), a ReLU layer (for non-linearisation), a max-pooling layer (for subsampling) and a dropout layer (for regularisation) are attached subsequently. Actually, our CNN classifier can be regarded as a simplification of the network proposed in [42]. It can be empirically observed that our simplified network can also work efficiently on these public datasets but with less training time.

As mentioned above, CNN can produce a posterior probability vector $\boldsymbol{p} = [p_1, p_2, \cdots, p_m]$ for each classification. Herein G is denoted as the hand gesture, and p_g ($g = 1, ..., m$ and $\sum_1^m p_g = 1$) corresponds to the probability of the g^{th} gesture $P(G_g|\boldsymbol{X}_t)$. The gesture-owning maximum probability is taken as the

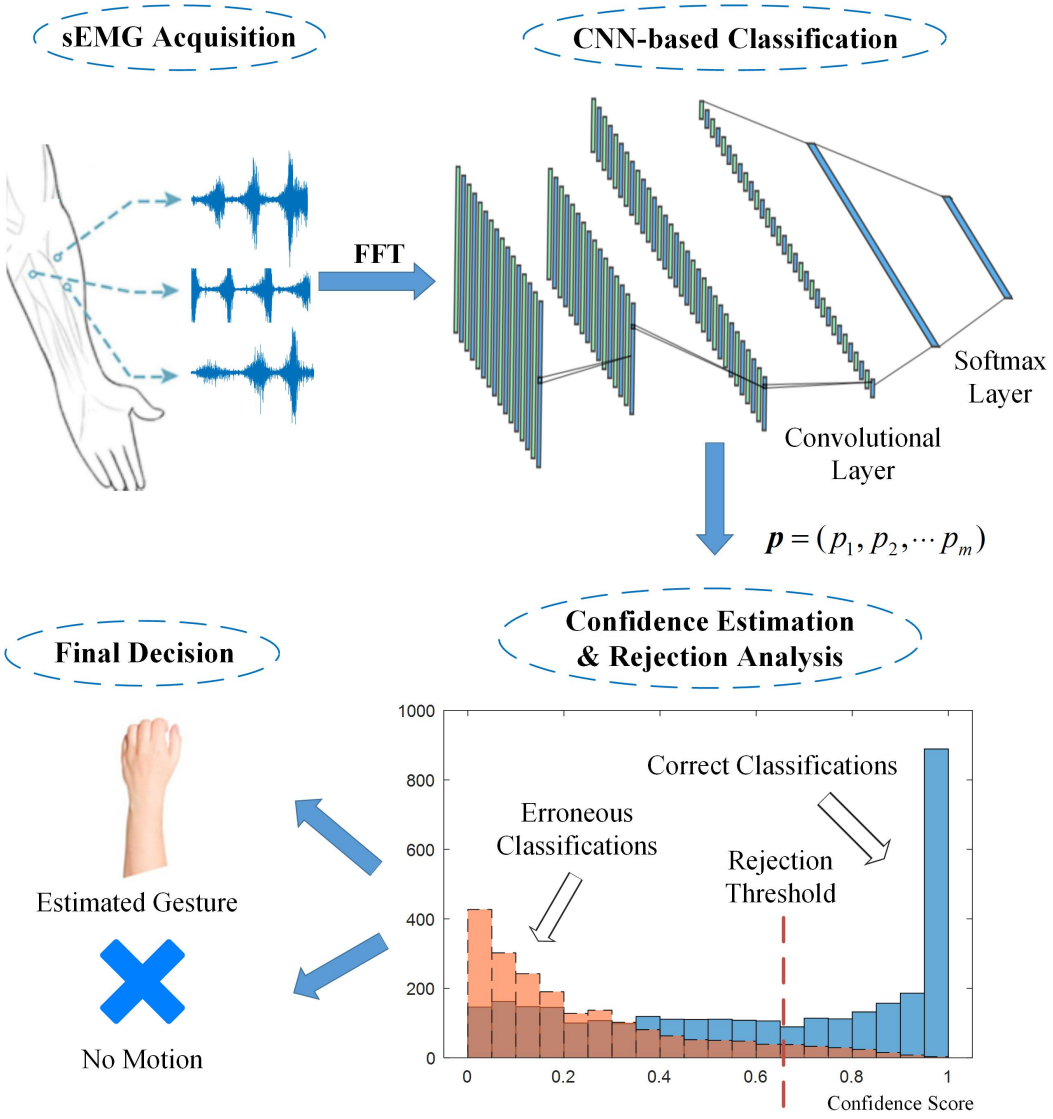


Figure 3.1: The framework for rejection-based hand gesture classification using CNN confidence.

Table 3.1: Layers configuration of implemented CNN.

Input: 1 × L × C sEMG matrix after FFT
1D Convolution, 32 kernels in size of 3
Batch Normalisation
ReLU
MaxPooling
Dropout
1D Convolution, 64 kernels in size of 3
Batch Normalisation
ReLU
MaxPooling, pool size of 3
Dropout
Fully Connected Layer
Softmax Layer

final prediction:

$$\hat{G} = \arg\max_{G_g} \left\{ P\left(G_g | \boldsymbol{X}_t\right) \right\}_{g=1,\ldots,m}. \tag{3.1}$$

Ideally $\boldsymbol{p}$ is expected to be a one-hot vector for a correct prediction, whilst a uniform distribution is reported when CNN becomes quite uncertain [158]. Thus $\boldsymbol{p}$ can be utilised to exploit confidence features for CNN.

3.2.3 Confidence Estimation

To indicate how confident the CNN classifier is about its prediction, a confidence estimation model is proposed by applying a zero-order smooth-step function to the weighted posterior probability distribution of CNN. The mathematical expression of this confidence model is

$$\text{ConfScore}\ (\boldsymbol{p}^*, \boldsymbol{\beta}) = \begin{cases} 0 & \boldsymbol{p}^*\boldsymbol{\beta}^T \leq \gamma_1 \\ \frac{\boldsymbol{p}^*\boldsymbol{\beta}^T - \gamma_1}{\gamma_2 - \gamma_1} & \gamma_1 < \boldsymbol{p}^*\boldsymbol{\beta}^T < \gamma_2 \\ 1 & \boldsymbol{p}^*\boldsymbol{\beta}^T \geq \gamma_2 \end{cases}, \tag{3.2}$$

where $\boldsymbol{p}^* = [p_1^*, p_2^*, \cdots p_m^*]$ is obtained by sorting the posterior probability vector $\boldsymbol{p}$ in a descending order. The element p_1^* is the largest posterior probability in $\boldsymbol{p}$ and p_m^* is the smallest. $\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots \beta_m]$ is a coefficient vector, γ_1 and γ_2 are user-defined hyper-parameters to decide left and right edges. Similar to Confnet proposed in [158], $\text{ConfScore}\,(\boldsymbol{p}^*, \boldsymbol{\beta})$ can be regarded as a feed-forward neural network cascaded with the softmax layer of CNN. Due to characteristics of the smooth-step function, outputs of $\text{ConfScore}\,(\boldsymbol{p}^*, \boldsymbol{\beta})$ are mapped between [0, 1]. In the following part, ConfScore is used to denote estimations of $\text{ConfScore}\,(\boldsymbol{p}^*, \boldsymbol{\beta})$.

In this study, $\boldsymbol{\beta}$ is designed as a learnable parameter that can be tuned in a supervised manner. To obtain training data for $\text{ConfScore}\,(\boldsymbol{p}^*, \boldsymbol{\beta})$, a group of classification results of CNN is firstly prepared as follows

$$T_r = \left\{ \left(\boldsymbol{p}_1^*, \hat{G}_1, \tilde{G}_1 \right), \cdots, \left(\boldsymbol{p}_N^*, \hat{G}_N, \tilde{G}_N \right) \right\}, \tag{3.3}$$

where $\hat{G}_j$ is the estimated class obtained by Eq. 3.1, $\tilde{G}_j$ $(j = 1, ..., N)$ denotes the ground truth gesture for the j^{th} sample.

Then T_r is relabelled using l_j by

$$l_j = \begin{cases} 1 & \hat{G}_j = \tilde{G}_j \\ -1 & \hat{G}_j \neq \tilde{G}_j \end{cases}, \tag{3.4}$$

where a relabelled dataset $T = \{(\boldsymbol{p}_1^*, l_1), \cdots, (\boldsymbol{p}_N^*, l_N)\}$ can be obtained to provide ground-truth labels for the supervised learning of $\text{ConfScore}\,(\boldsymbol{p}^*, \boldsymbol{\beta})$. Specifically, the ground truth of $\text{ConfScore}\,(\boldsymbol{p}^*, \boldsymbol{\beta})$ is +1 for correct CNN outcomes or -1 for erroneous ones. To tune parameter $\boldsymbol{\beta}$ effectively, a metric should be determined to evaluate the performance of $\text{ConfScore}\,(\boldsymbol{p}^*, \boldsymbol{\beta})$ in T. Considering that higher scores should be correlated with more accurate predictions, Wan *et al.* [158] defined the mean effective confidence (MEC) as follows:

$$\text{MEC} = \frac{1}{N} \sum_{j \in T} \text{ConfScore} \left(p_j^d, \boldsymbol{\beta} \right) * l_j, \tag{3.5}$$

where MEC $\in [-1, 1]$. From Eq. 3.4 and Eq. 3.5 we can see that a larger MEC can be obtained when correct classifications have the higher ConfScore results whilst erroneous decisions are with the lower scores. In other words, when a better confidence estimation is conducted for T, the separation between correct and erroneous CNN predictions is expected to be more distinguishable [94].

However, as shown in Eq. 3.5, MEC averages the confidence features of all samples in T. Thus MEC is sensitive to an unbalanced T which is composed of either too many correct or erroneous classifications. To solve this problem, Balanced MEC (BMEC) is further defined as

$$\begin{aligned} \text{BMEC} = & \frac{1}{N_1} \sum_{i \in T_C} \text{ConfScore} \left(\boldsymbol{p}_i^*, \boldsymbol{\beta} \right) * l_i \\ & + \frac{1}{N_2} \sum_{j \in T_E} \text{ConfScore} \left(\boldsymbol{p}_j^*, \boldsymbol{\beta} \right) * l_j, \end{aligned} \tag{3.6}$$

where BMEC $\in [-1, 1]$, T_C is only composed of correct classifications ($l = 1$) whilst T_E only consists of erroneous classifications ($l = -1$). The number of samples in T_C and T_E is defined as N_1 and N_2, respectively. Compared with MEC, BMEC is more robust to the imbalance of T_C and T_E.

Based on Eq. 3.6, the optimisation of $\boldsymbol{\beta}$ can be defined as

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \text{BMEC}. \tag{3.7}$$

Since Eq. 3.7 works as the objective function and is non-differentiable, heuristic methods can be adopted to find obtain the local optimal $\hat{\boldsymbol{\beta}}$. Herein, the genetic algorithm (GA) is applied in which solutions can evolve efficiently over generations. GA is one of the most widely applied evolution algorithms in the optimisation of intricate problems in different fields. Compared with many other heuristic algorithms, it is believed to own better global searching capability [162]. The whole process can be summarised in Algorithm 1.

Algorithm 1: The Proposed ConfScore $(\boldsymbol{p}^*, \boldsymbol{\beta})$.

Input: A group of CNN outcomes T_r, hyper-parameters γ_1 and γ_2.

Output: Optimal weights $\hat{\boldsymbol{\beta}}$

1: Construct the relabelled dataset T from T_r.

2: Initialise $\boldsymbol{\beta}$.

3: Use GA to search $\hat{\boldsymbol{\beta}}$:

4: Based on Eq. 3.2, calculate ConfScore $(\boldsymbol{p}^*, \boldsymbol{\beta})$ for each sample in T.

5: Based on Eq. 3.6, calculate the BMEC value using outputs of the obtained ConfScore $(\boldsymbol{p}^*, \boldsymbol{\beta})$ together with corresponding labels l in T.

6: Update $\boldsymbol{\beta}$ using GA operators, where BMEC works as the objective function.

7: Until GA is converged.

8: Return $\hat{\boldsymbol{\beta}}$ to obtain the final confidence model.

3.2.4 Rejection Rule

In the rejection process, CNN classifications whose ConfScore is smaller than a user-defined threshold $\alpha \in (0, 1)$ should be rejected to no motion states. Thus, the rejection function can be formulated as

$$\mathrm{R}\left(\boldsymbol{p}^*, \alpha\right) = \begin{cases} \text{accept} & \text{ConfScore}\left(\boldsymbol{p}^*, \hat{\boldsymbol{\beta}}\right) \geq \alpha \\ \text{reject} & \text{otherwise} \end{cases}. \tag{3.8}$$

As illustrated in Figure 3.1, once $\hat{\boldsymbol{\beta}}$ is calculated and α is determined, these parameters can be applied in the rejection framework to decrease erroneous movements and thereby enhance model reliability in myoelectric control systems.

3.2.5 Rejection Analysis

Since $\mathrm{R}\left(\boldsymbol{p}^*, \alpha\right)$ works as a binary classifier to further identify CNN predictions, the rejection results can be divided into true acceptance (TA) cases, false ac-

Table 3.2: Descriptions of TA/FA/FR/TR cases in rejection process.

		CNN Classifications	
		Correct	Erroneous
$\mathrm{R}(\boldsymbol{p}^*, \alpha)$	Accept	TA: accepted CNN estimations that are actually correct	FA: accepted CNN estimations that are actually erroneous
	Reject	FR: rejected CNN estimations that are actually correct	TR: rejected CNN estimations that are actually erroneous

ceptance (FA) cases, false rejection (FR) cases and true rejection (TR) cases. Descriptions of TA/FA/FR/TR are listed in Table 3.2. Based on these cases, the true acceptance rate (TAR) can be calculated to denote the proportion of correct CNN estimations to be accepted by $\mathrm{R}(\boldsymbol{p}^*, \alpha)$, whereas the true rejection rate (TRR) is the rejection ratio of erroneous classifications:

$$\begin{aligned} \mathrm{TAR} &= \frac{\sum \mathrm{TA}}{\sum \mathrm{TA} + \sum \mathrm{FR}} \\ \mathrm{TRR} &= \frac{\sum \mathrm{TR}}{\sum \mathrm{TR} + \sum \mathrm{FA}}. \end{aligned} \tag{3.9}$$

Herein, TRR represents the rejection efficiency whilst TAR corresponds to the cost. According to previous work [94], a trade-off between TRR and TAR is essential to the evaluation of α because a small α can result in the acceptance of too many erroneous classifications, whilst a large one may reject too many correct decisions. In this study, a novel evaluation metric Fit is proposed to consider both TRR and TAR:

$$\mathrm{Fit} = \mathrm{TAR} + \mathrm{TRR} - 1. \tag{3.10}$$

From Table 3.2 and Eq. 3.9 we can see that TRR and TAR correspond to specificity and sensitivity in binary classification, respectively. Thus Fit in Eq. 3.10 is equivalent to the Youden's J statistic, a commonly used measure of overall differentiation effectiveness in disease diagnoses [163, 164].

The error rate (Err) is related to user security in myoelectric control systems, hence it is also of great concern in rejection. For a CNN, Err can be simply equal

to $1 - \text{Acc}$, where Acc denotes the classification accuracy [43]:

$$\text{Acc} = \frac{\text{number of correct classifications}}{\text{number of test samples}} \times 100\%. \tag{3.11}$$

When $\text{R}(\boldsymbol{p}^*, \alpha)$ is applied, part of the erroneous classifications will be rejected as no motion states, thus only those that are wrongly accepted by $\text{R}(\boldsymbol{p}^*, \alpha)$ will be counted as errors. Referring to Table 3.2, Err is revised as

$$\text{Err} = \begin{cases} 1 - \text{Acc} & \text{CNN} \\ \frac{\sum \text{FA}}{\sum \text{TA} + \sum \text{FR} + \sum \text{FA} + \sum \text{TR}} & \text{CNN+Reject} \end{cases}. \tag{3.12}$$

3.2.6 Baseline Methods

To evaluate the effectiveness of the ConfScore model, two popular confidence features, i.e., the maximum posterior probability (MaxProb) and the entropy of the probability distribution [157, 158], are utilised as the baseline. In accordance with [158], MaxProb is normalised from $\left[\frac{1}{g}, 1\right]$ to $[0,\ 1]$ for a fair comparison, where g is the number of classes. Considering that entropy is in general negatively related to classification accuracy, the inverse entropy (IEntropy) is defined as

$$\text{IEntropy} = \frac{\log_2 m + \sum_{g=1}^{m} p_g \log_2 p_g}{\log_2 m}. \tag{3.13}$$

3.3 Experiment Setup

3.3.1 Public Datasets

To evaluate the confidence estimation and rejection performance, six datasets of the NinaPro database (denoted as DB1-DB6, respectively) were utilised. DB1 was recorded using 10 Otto Bock 13-E200 electrodes. DB2, DB3 and DB6 were recorded using a Delsys Trigno wireless system. DB4 was recorded with a Cometa Wave Wireless sEMG system using Dormo SX-30 ECG electrodes. DB5 utilised the two Thalmic Myo armbands which is a low-cost device. More details can

be found in Table 3.3 and [165, 166]. The sampling rates are 100Hz for DB1, 200Hz for DB5 and 2000Hz for the other databases. These datasets have been widely applied in pilot studies for sEMG-based hand gesture classification [38, 42–44]. In terms of experiment protocols, DB1, DB2, DB4 and DB5 include more than 50 different hand or wrist movements of intact subjects. For example, 49 movements relevant to the activities of daily living are present in the DB2, including 8 isometric and isotonic hand configurations, 9 basic wrist movements, 23 grasping and functional movements and 9 force patterns. Different from these datasets, DB3 is composed of data collected from upper limb amputees, whilst the data of DB6 were recorded from 10 intact subjects repeating 7 grasps twice a day for 5 consecutive days [166].

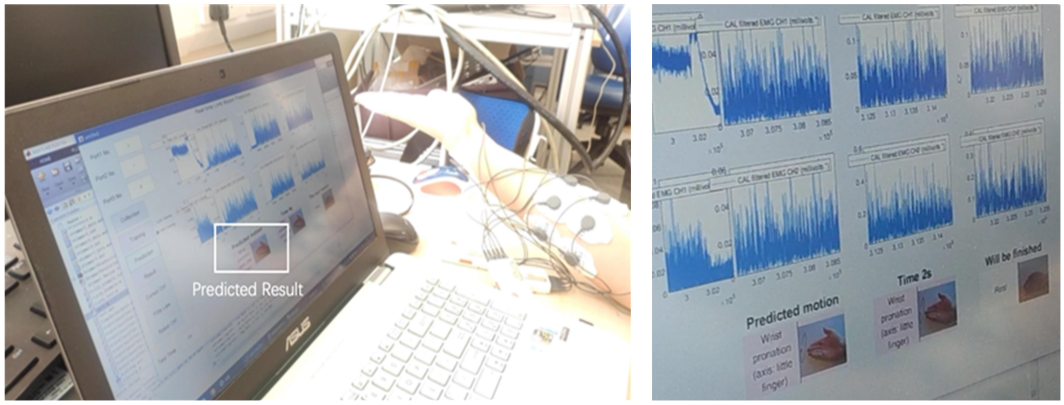
3.3.2 Online Verification

To validate the effectiveness of ConfScore in real-time applications, a customised online platform was developed based on Shimmer wearable sensors and the Shimmer MATLAB Instrument Driver [167]. The platform was composed of several main modules: sEMG collection and streaming, data processing and plotting, CNN training, ConfScore $(\boldsymbol{p}^*, \boldsymbol{\beta})$ tuning, online classification and rejection analysis. The experiment involved six basic wrist/hand gestures: wrist flexion, wrist extension, supination, pronation, palm open, and palm close. Figure 3.2 illustrates the prediction interface, where the above six diagrams plot the filtered real-time sEMG signals, and the bottom three images display the predicted gesture, the prompted gesture and the following gesture, respectively.

Approved by the MaPS and Engineering joint Faculty Research Ethics Committee of University of Leeds, UK (reference MEEC 18-006), four healthy subjects (three males and one female, aged 20-55) took part in the experiment. In the training phase, participants were asked to perform predefined gestures following instructions given by the system. The ground-truth labels were created by requiring the user to hold each gesture for five seconds. Twelve bipolar electrodes were placed on the proximal portion of the left forearm to collect sEMG signals in six channels. The sampling rate of sEMG was set as 1024Hz.

Table 3.3: Specifications of the public databases used in this paper.

Database	No. of gestures	Major gestures	Upper limb amputation	No. of channels	Devices
DB1	52	finger/wrist/grasping movements	healthy	10	10 Otto Bock 13-E200, 100Hz
DB2	49	finger/wrist/grasping movements force patterns	healthy	12	Delsys Trigno wireless system, 2000Hz
DB3	49	finger/wrist/grasping movements force patterns	amputated	12	Delsys Trigno wireless system, 2000Hz
DB4	52	finger/wrist/grasping movements	healthy	12	Cometa Wave Wireless sEMG system, 2000Hz
DB5	52	finger/wrist/grasping movements	healthy	16	Thalmic Myo armbands, 200Hz
DB6	7	grasping movements	healthy	16	Delsys Trigno wireless system, 2000Hz



(a) (b)

Figure 3.2: Online testing using the customised platform (a) and screen shot of the predicted result (b).

3.3.3 Data Pre-processing

DB1 provides a bandpass-filtered and Root-Mean-Square (RMS) rectified version of sEMG. DB4 was processed by a 10Hz high-pass filter and a 1000Hz low-pass filter. A Hampel filter was adopted to clean 50Hz power-line interference from sEMG collected by the Delsys and Cometa sensors, i.e., DB2, DB3, DB4, and DB6. For DB5, Thalmic Myo incorporated a notch filter at 50Hz. Based on filtered sEMG, a min–max normalisation was implemented for each subject individually [168]. This normalisation method was adopted since it can keep the original distribution of sEMG. To construct sEMG matrices for CNN, the window length was set as 300ms with a 50ms step for DB1 and DB5. By contrast, 150/50ms was set empirically for other databases. Herein the window length of DB1 and DB5 is comparatively larger because the sampling rates of these two databases are quite low (100Hz and 200Hz, respectively), thus the matrices constructed from shorter time windows could not support CNN. In the online platform, a 3^{rd} order Butterworth band pass filter (20-450Hz) and a 50Hz notch filter were applied for noise reduction. To construct the sEMG matrix for CNN, the window length and segmentation step were set as 150ms and 25ms, respectively.

3.3.4 Data Split

Following previous work [38, 42, 43, 169], approximately two-thirds of the gesture trials in each subject of DB1-DB5 were utilised to train CNN and tune ConfScore subsequently. The remaining trials of the participant worked as the testing set to analyse confidence/rejection performances. Specifically, repetitions 2, 5 and 7 were combined as the testing set in DB1; in DB2-DB5, repetitions 2 and 5 were used for testing. Since DB6 consists of data from multiple days, CNN was trained using data on the first day (Day 1), tuned ConfScore on Day 2, and tested the performances on Days 3-5.

3.3.5 Training of CNN

Hyper-parameters of CNN were first identified according to [42, 43] and then tuned empirically. Specifically, the network was trained in a 128-sized mini batch using stochastic gradient descent with momentum (SDGM). The momentum rate of SDGM was set as 0.9. The dynamic learning rate was initialised to be 0.0005. The dropout rate was 0.1 after every 10 epochs. The $L2$ regularisation rate was set to be 0.01, and 30 epochs were adopted for data training in DB1-DB5 and our online platform. The training data were shuffled in every epoch. To reduce over-fitting, the dropout rate was set as 0.2 for DB1-DB5. Fewer epochs (20) and a higher dropout rate (0.5) were used for DB6 due to the domain variation among training and testing sessions in different days.

3.3.6 Training of ConfScore

In ConfScore $(\boldsymbol{p}^*, \boldsymbol{\beta})$, the left and right edges of activation function can be defined flexibly by adjusting γ_1 and γ_2. Empirically, it is observed that an explicit tuning of γ_1 and γ_2 for each dataset can further optimise confidence estimation. For the sake of simplicity and generalisation, in this study γ_1/γ_2 was fixed as $\{\gamma_1 = 0, \gamma_2 = 1\}$ for all trials in six databases and the online platform. In addition, the GA was utilised to search $\hat{\boldsymbol{\beta}}$ by exploiting reproduction, crossover,

and mutation operators. In our implementation, an elitist strategy was further incorporated in this algorithm to enhance convergence.

3.3.7 Statistical Analysis

In this study, statistical analysis was performed using the Statistics and Machine Learning Toolbox in MATLAB R2012a. In particular, the assumption of data normality was first checked via the Shapiro–Wilk test (the level was set to be 0.05) [170]. If the assumption was satisfied, the one-way analysis of variance (ANOVA) test was applied to verify the differences in methods of confidence estimation and rejection process; otherwise, its non-parametric equivalent, i.e., Kruskal–Wallis (KW) test, was performed alternatively.

3.4 Results

3.4.1 Distribution of Confidence Features

As suggested in [158], distributions of confidence features of a classifier are expected to be indicative of classification accuracy, i.e., correct classifications are with high scores (close to one) whereas wrong predictions result in lower scores (close to zero). Figure 3.3 visualises the distributions of correct and erroneous classifications following three different confidence features. As we can see, correct classifications are overwhelmingly gathering in the range of [0.95, 1] when ConfScore is utilised. In opposite, erroneous classifications are gathering mainly in bins of smaller ConfScore. Differently, distributions between correct and erroneous classifications tend to be less distinguishable when MaxProb or IEntropy is utilised. Therefore, it can be inferred that ConfScore can be more relevant with CNN confidence in terms of the classification accuracy. In the following sections how the distribution differences further influence BMEC and Fit of three confidence features will be further explored.

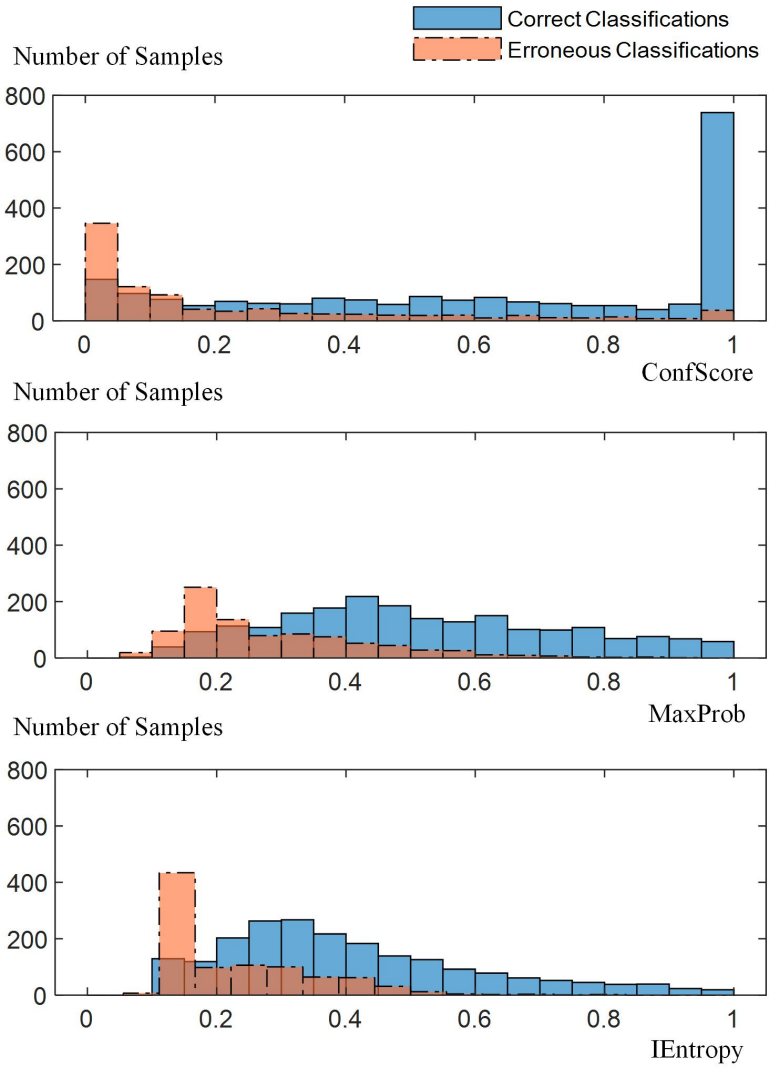


Figure 3.3: Distributions of correct and erroneous classifications in testing sets of Subject 2-DB1. The width of a bin is 0.05 and the amplitude of each bin denotes the number of samples (CNN outputs) whose confidence features are located in the corresponding range.

Table 3.4: BMEC of ConfScore, MaxProb and IEntropy for all subjects in six databases. SD denotes the standard deviation.

Database	Confidence Feature	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Mean	SD
DB1	ConfScore	**0.39**	**0.41**	**0.36**	**0.34**	**0.29**	**0.34**	**0.3**	**0.38**	**0.28**	**0.35**	**0.34**	0.04
	MaxProb	0.31	0.34	0.3	0.3	0.23	0.27	0.26	0.3	0.22	0.29	0.28	0.04
	IEntropy	0.27	0.3	0.25	0.25	0.2	0.23	0.22	0.27	0.19	0.25	0.24	0.03
DB2	ConfScore	**0.47**	**0.43**	**0.31**	**0.29**	**0.29**	**0.33**	**0.22**	**0.5**	**0.38**	**0.37**	**0.36**	0.09
	MaxProb	0.34	0.3	0.21	0.22	0.21	0.24	0.17	0.36	0.25	0.23	0.25	0.06
	IEntropy	0.27	0.24	0.16	0.18	0.15	0.18	0.13	0.3	0.2	0.18	0.2	0.05
DB3	ConfScore	**0.22**	**0.31**	**0.69**	**0.14**	**0.08**	**0.12**	0.03	**0.31**	**0.33**	**0.09**	**0.23**	0.19
	MaxProb	0.16	0.22	0.56	0.12	0.06	0.09	**0.04**	0.24	0.23	0.07	0.18	0.15
	IEntropy	0.13	0.15	0.55	0.1	0.05	0.06	0.01	0.19	0.17	0.06	0.15	0.15
DB4	ConfScore	**0.57**	**0.42**	**0.22**	**0.66**	**0.63**	**0.42**	**0.69**	**0.65**	**0.48**	**0.47**	**0.52**	0.15
	MaxProb	0.43	0.3	0.18	0.46	0.46	0.32	0.5	0.48	0.37	0.36	0.39	0.10
	IEntropy	0.4	0.23	0.13	0.42	0.42	0.26	0.47	0.45	0.31	0.32	0.34	0.11
DB5	ConfScore	**0.62**	**0.6**	**0.64**	**0.64**	**0.62**	**0.58**	**0.61**	**0.57**	**0.64**	**0.62**	**0.61**	0.04
	MaxProb	0.52	0.47	0.5	0.53	0.47	0.46	0.48	0.42	0.45	0.56	0.49	0.04
	IEntropy	0.47	0.4	0.48	0.48	0.45	0.43	0.45	0.37	0.41	0.51	0.44	0.04
DB6 (Day3)	ConfScore	**0.28**	**0.03**	**0.22**	**-0.02**	**0.18**	**0.17**	**0.62**	**0.1**	**0.11**	**0.39**	**0.21**	0.19
	MaxProb	0.14	0.01	0.1	-0.05	0.06	0.16	0.41	0.07	0.1	0.22	0.12	0.13
	IEntropy	0.11	0.03	0.07	-0.07	0.05	0.16	0.41	0.09	0.07	0.19	0.11	0.13
DB6 (Day4)	ConfScore	**0.39**	**0.3**	**0.01**	**0.04**	**0.13**	**0.22**	**0.34**	**0.27**	**0.26**	**0.11**	**0.21**	0.13
	MaxProb	0.22	0.21	-0.01	0.02	0.06	0.14	0.19	0.19	0.17	0.06	0.13	0.08
	IEntropy	0.21	0.19	-0.01	0	0.06	0.11	0.18	0.15	0.17	0.07	0.11	0.08
DB6 (Day5)	ConfScore	**0.32**	**0.24**	**0.18**	**0.23**	**0.4**	**0.33**	**0.22**	**0.19**	**0.18**	**0.38**	**0.27**	0.08
	MaxProb	0.17	0.15	0.16	0.12	0.3	0.28	0.13	0.19	0.12	0.2	0.18	0.06
	IEntropy	0.12	0.13	0.17	0.08	0.26	0.3	0.14	0.15	0.12	0.17	0.16	0.07

3.4.2 BMEC of Confidence Features

Table 3.4 lists BMEC of ConfScore, MaxProb and IEntropy in six public databases. In summary, BMEC of ConfScore is larger than MaxProb and IEntropy for most subjects in each database. From Table 3.4 it is also observed that the average of BMEC of all three confidence features in DB1, DB2, DB4, and DB5 are comparatively larger than the average of all three confidence features in DB3, DB6 because confidence distributions of correct and erroneous classifications are less distinguishable in DB3 and DB6. The deterioration occurs since the experimental protocols in DB3 and DB6 are more challenging. In terms of DB3, it is hard for trans-radial amputees to produce reliable ground truth because of the inability to operate any sensor on the missing limbs [169]. In DB6, both CNN and ConfScore are trained and tested in different days, where the electrode shift can have severe impacts on model accuracy.

3.4.3 Analysis of Rejection Process

In this section, the effect of $\mathrm{R}(\boldsymbol{p}^*, \alpha)$ on the basis of ConfScore is further analysed. Referring to Eq. 3.8 and Table 3.2, given a specific α, Err, TAR, TRR, and Fit can be calculated accordingly. Figure 3.4 shows variations of these four metrics following a changing α in the testing set of Subject 1-DB4. The variation step for α is set to be 0.05. Figure 3.4 shows that Err is decreasing monotonically along with α. When $\alpha = 0$, there is no rejection cascaded with CNN, thus $\mathrm{Err} = 1 - \mathrm{Acc}$. By contrast, Err becomes zero when $\alpha = 1$, since all CNN decisions are rejected in this case.

Since TAR decreases monotonically and TRR increases inversely, a concave downward curve of Fit is thus obtained. More specifically, Fit increases continuously when α is comparatively smaller, and there comes a turning point (the aubergine circles in Figure 3.4) when α becomes larger. Hence, focusing on Err or TRR alone can result in an decreased TAR, verifying the necessity of Fit for the trade-off between TRR and TAR in rejection analysis. When the rejection is conducted around the turning point shown in Figure 3.4, a smaller Err (1.2% for

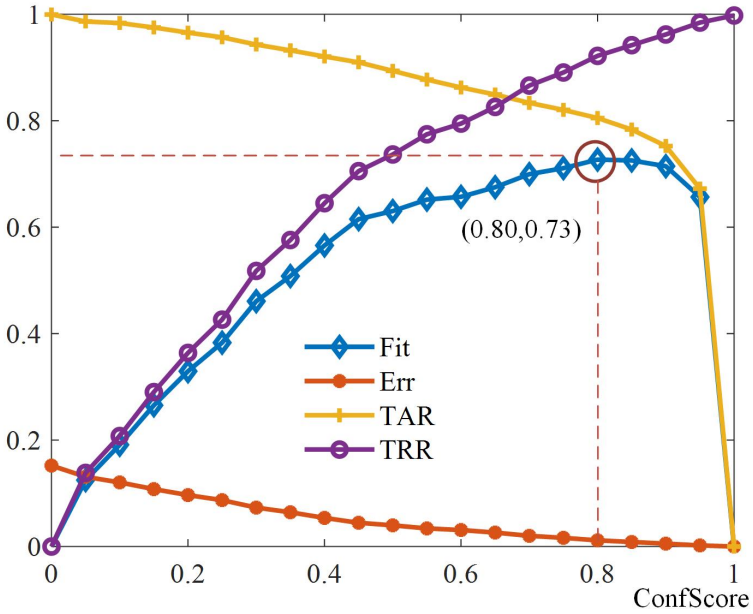


Figure 3.4: Err, TAR, TRR, and Fit for various rejection thresholds in testing sets of Subject 1-DB4.

Subject 1-DB4) with an acceptable TAR (80.5%) can be obtained.

To further illustrate the effectiveness of Fit in rejection, the optimal α of training sets is applied to the corresponding testing sets in each subject. Figure 3.5 compares Err with and without rejection for testing sets in all databases. Moreover, Figure 3.6 shows statistical results of TAR and TRR in Fit of all databases. In this study, ANOVA/KW is conducted for statistical analysis. The feature factor has two levels (Rejection/NoRejection for Figure 3.5, TAR/TRR for Figure 3.6), and subjects of each database work as the random variable. From Figure 3.5 we can see that in each database Err is reduced significantly by rejecting low confidence classifications. Moreover, since Fit attempts to compromise TAR and TRR, these two metrics can be close to each other in most databases. Taking DB5 as example, the mean value of TAR is 0.83 whilst TRR reaches 0.9 on average, indicating that the majority of erroneous classifications can be rejected whilst causing small loss on correct classifications, indicating a promising trade-off between system security and rejection cost.

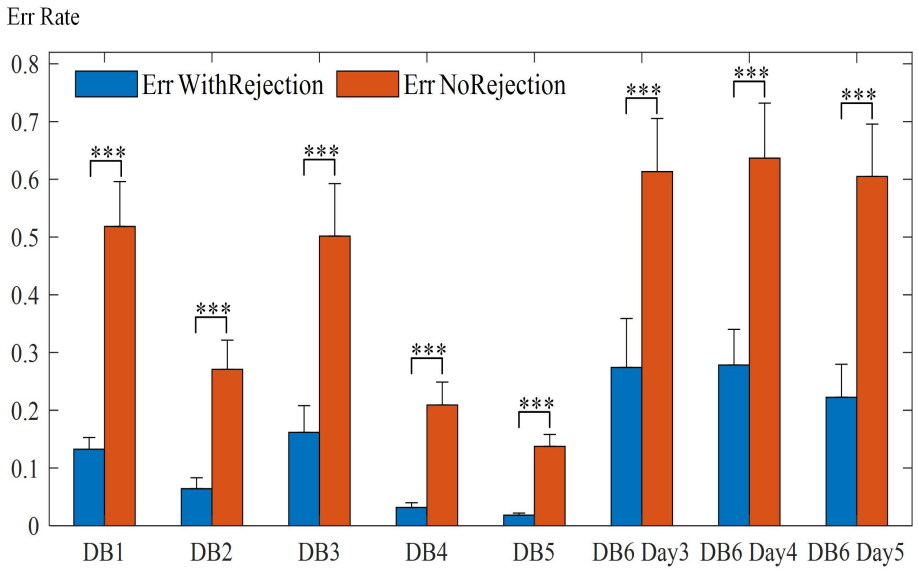


Figure 3.5: Statistical analysis of Err for testing sets with and without rejection in all databases (***p-value < 0.001, **p-value < 0.01, *p-value < 0.05).

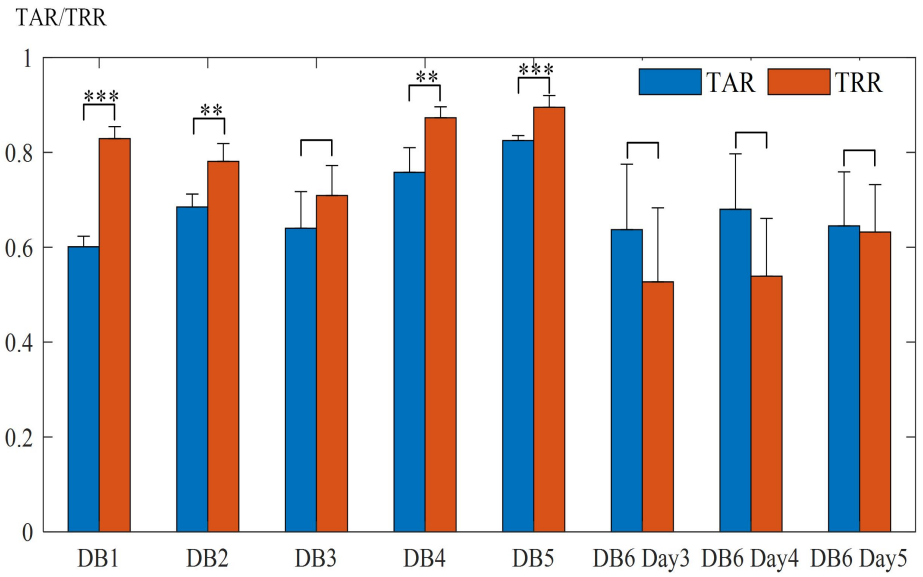


Figure 3.6: Statistical analysis of TAR and TRR for testing sets in all databases (***p-value < 0.001, **p-value < 0.01, *p-value < 0.05).

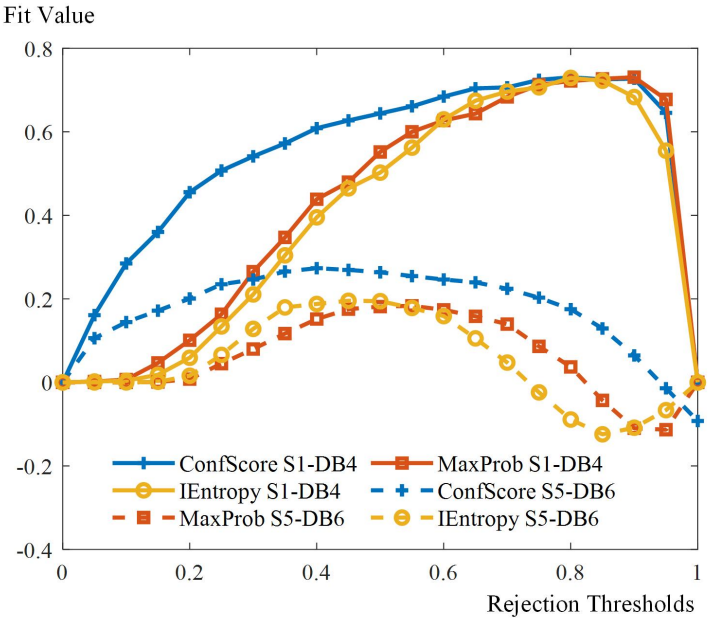


Figure 3.7: Fit values for various rejection thresholds when three confidence features are used in testing sets of Subject1-DB4 and Subject 5-DB6.

3.4.4 Comparison of Confidence Features in Rejection

Figure 3.7 shows Fit values for various rejection thresholds in testing sets of Subject1-DB4 and Subject 5-DB6. Recall that Fit can be smaller than zero, in which case $\mathrm{R}(\boldsymbol{p}^*, \alpha)$ either accepts too many erroneous CNN decisions or rejects too many correct ones. From this figure, several interesting results can be observed. Firstly, for each confidence feature the Fit of α in Subject1-DB4 are in general much higher than in Subject 5-DB6. This observation will be further discussed in Section 3.5.2. Secondly, although the maximal Fit of three confidence features can be close in some cases (such as Subject1-DB4), the Fit curves of ConfScore are always much wider and more flattened than the curves of two other features. This characteristic means that when adopting ConfScore as a CNN confidence feature, Fit is less sensitive to variations of α in the rejection process, which contributes to a wider range for threshold selection. In the following part, this characteristic is denoted as the rejection robustness for Fit.

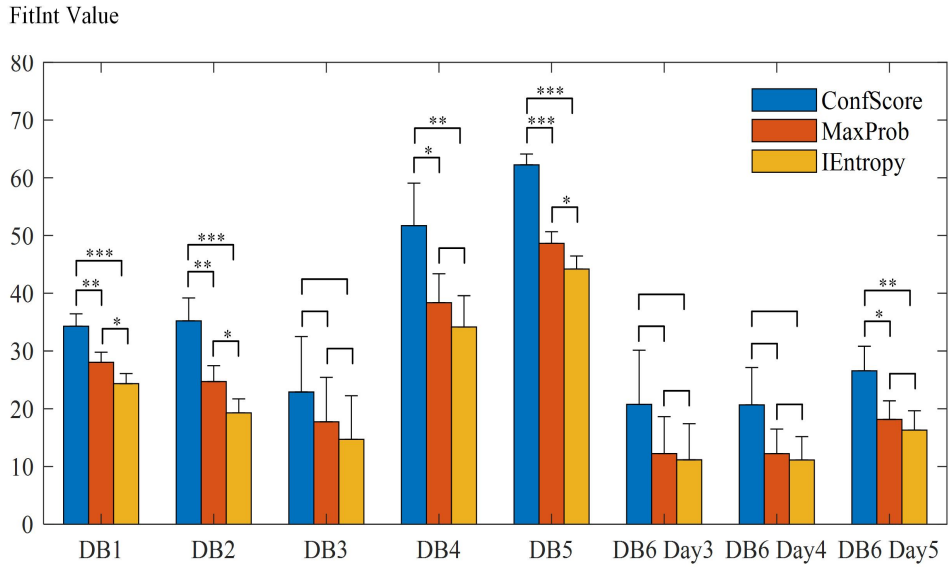


Figure 3.8: Statistical analysis of FitInt values of rejection in all databases when three confidence features are adopted (***p-value < 0.001, **p-value < 0.01, *p-value < 0.05).

To further quantify the rejection robustness, the integral of Fit curves (denoted as FitInt) is calculated using the trapezoidal method. Figure 3.8 illustrates the mean and standard deviations of FitInt for all subjects in each database. As we can see, FitInt of ConfScore is much larger than FitInt of MaxProb and IEntropy in most databases. Thus, the rejection performance can be less sensitive to threshold variations when ConfScore is used as the confidence feature. In addition, it can also be observed that FitInt values of three confidence features in DB3 and DB6 are with smaller means but larger standard deviations. The degradation is consistent with BMEC performance shown in Table 3.4.

3.4.5 Online Results

Table 3.5 and Table 3.6 list the BMEC and FitInt of three confidence features in online testing. These tables show that ConfScore outperforms MaxProb and IEntropy significantly in both confidence distribution and rejection robustness (as for

Table 3.5: BMEC of ConfScore, MaxProb, and IEntropy in online testing.

Confidence Feature	S1	S2	S3	S4	Mean	SD
ConfScore	0.45	0.53	0.38	0.28	0.41	0.11
MaxProb	0.22	0.26	0.12	0.13	0.18	0.06
IEntropy	0.18	0.20	0.05	0.07	0.13	0.08

Table 3.6: FitInt of ConfScore, MaxProb, and IEntropy in online testing.

Confidence Feature	S1	S2	S3	S4	Mean	SD
ConfScore	40.35	49.76	39.92	29.66	39.92	8.21
MaxProb	18.95	23.93	13.82	13.65	17.59	4.89
IEntropy	14.81	18.52	5.37	6.62	11.33	6.36

BMEC, the p-value of ConfScore versus MaxProb/IEntropy is 0.011/0.005; as for FitInt, the p-value is 0.003/0.002). These outcomes are consistent with results of offline analysis (Table 3.4 and Figure 3.8). Moreover, since the confidence-based rejection can be regarded as a post-processing method, performance of our approach is compared with a widely applied smoothing strategy, i.e. Majority Vote (MV) [171]. Table 3.7 lists the Err of CNN, CNN+MV, and CNN+Rejection for four subjects in online testing. Specifically, the ConfScore works as the confidence feature in rejection, whilst MV makes the final decision using the current classification result along with six previous results. As shown in Table 3.7, although MV can further reduce Err in classification, its performance is evidently worse than the performance of the confidence-based rejection because the final result of MV is still an active motion rather than a no-motion state. Compared with MV, the rejection method is more useful when the control security is crucial.

3.5 Discussion

In this study, preliminary research was conducted to estimate model confidence for CNN-based hand gesture recognition. Specifically, a novel confidence estima-

Table 3.7: Err of CNN, CNN+MV, and CNN+Rejection in online testing. The ConfScore works as the confidence feature in rejection.

Method	S1	S2	S3	S4	Mean	SD
CNN	0.26	0.28	0.32	0.37	0.31	0.05
CNN+MV	0.24	0.25	0.31	0.34	0.29	0.05
CNN+Rejection	0.07	0.07	0.10	0.15	0.10	0.04

tion model ConfScore $(\boldsymbol{p}^*, \boldsymbol{\beta})$ is proposed to generate confidence scores based on posterior probabilities of CNN. Following a threshold-based rejection rule, unconfident classifications can be identified and rejected. In addition, although the main target of rejection is to refuse erroneous classifications and improve system security, focusing on this target alone may result in a serious control lag by setting overlarge rejection thresholds arbitrarily. To address this issue, a comprehensive evaluation metric named Fit is proposed to combine TAR and TRR, so that both system security and control lag can be fully considered in threshold selection or rejection analysis.

3.5.1 Confidence Estimation Model

In the design of ConfScore $(\boldsymbol{p}^*, \boldsymbol{\beta})$, the zero-order smooth-step function is utilised to map posterior likelihood of CNN to associated confidence features. A main advantage is that left and right edges of the activation function can be defined flexibly by adjusting γ_1 and γ_2. Table 3.8 lists the average BMEC of subjects in six databases when four different pairs of γ_1 and γ_2 are utilised. As we can see, an explicit tuning of γ_1 and γ_2 can further optimise confidence distributions in each database. Another novel design is the BMEC-based objective function to train ConfScore $(\boldsymbol{p}^*, \boldsymbol{\beta})$ in a supervised way. As discussed in [158], the confidence feature is expected to be closer to one when CNN is certain about the decision (i.e., the classification is prone to be correct) and to be near zero when CNNs are making uncertain decisions. As summarised in Table 3.4, for most cases of six datasets, BMEC values of ConfScore are comparatively larger than the values of

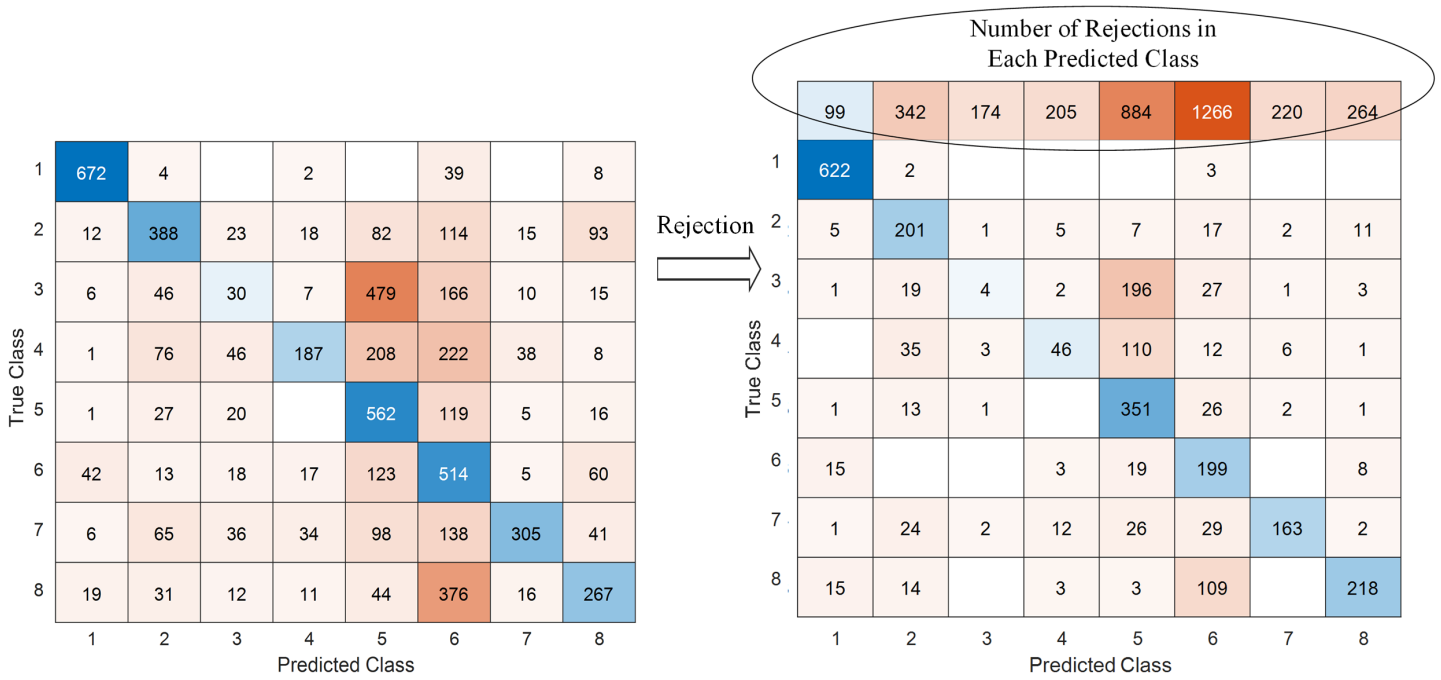


Figure 3.9: Confusion matrix without/with rejection based on ConfScore. The presented cased is Subject 1 of DB6. The optimal rejection threshold is pre-determined based on the training data. The error rate of conventional classification is high since training and testing datasets are from different days of a subject. The Fit of the rejection performance is 0.36.

Table 3.8: Average BMEC of ConfScore in each database using different pairs of γ_1 and γ_2.

Parameter Pairs	DB1	DB2	DB3	DB4	DB5	DB6
γ_1=0, γ_2=0.2	0.45	0.45	0.32	0.58	0.72	0.22
γ_1=0, γ_2=0.4	0.44	0.44	0.33	0.58	0.71	0.23
γ_1=0, γ_2=0.6	0.41	0.41	0.28	0.56	0.69	0.26
γ_1=0, γ_2=0.8	0.34	0.35	0.23	0.53	0.67	0.24

MaxProb and IEntropy. Therefore, it can be inferred that the proposed ConfScore can better reflect correctness of CNN classifications.

3.5.2 Design of Fit and FitInt

As illustrated in Figure 3.1, given the ConfScore of classification results, a threshold-based rejection can be utilised to improve the control reliability. To illustrate this process, Figure 3.9 compares the confusion matrix without/with rejection. As we can see, in the conventional classification without rejection, a high error rate is obtained in the case presented. Differently, by identifying classifications whose ConfScore values are smaller than a predetermined threshold, a large number of erroneous classifications can be identified and rejected. This is how ConfScore can help to improve the robustness of gesture classification. However, from Figure 3.9 it can also be observed that some of the correct classifications (those in the diagonal of the confusion matrix without rejection) are rejected mistakenly. These mistakes should be regarded as the cost of the rejection process. Therefore, as illustrated in Eq. 3.9 and Table 3.2, Fit is proposed to conduct a quantitative evaluation of rejection performance. From Figure 3.4 we can see that Fit compromises TAR and TRR to achieve a good balance between control continuity and system security.

In this study the rejection robustness of different confidence features (see Figure 3.8) is also compared. Our concerns are: 1) as shown in Figure 3.7, the selection of the rejection threshold can affect the Fit value; 2) the threshold

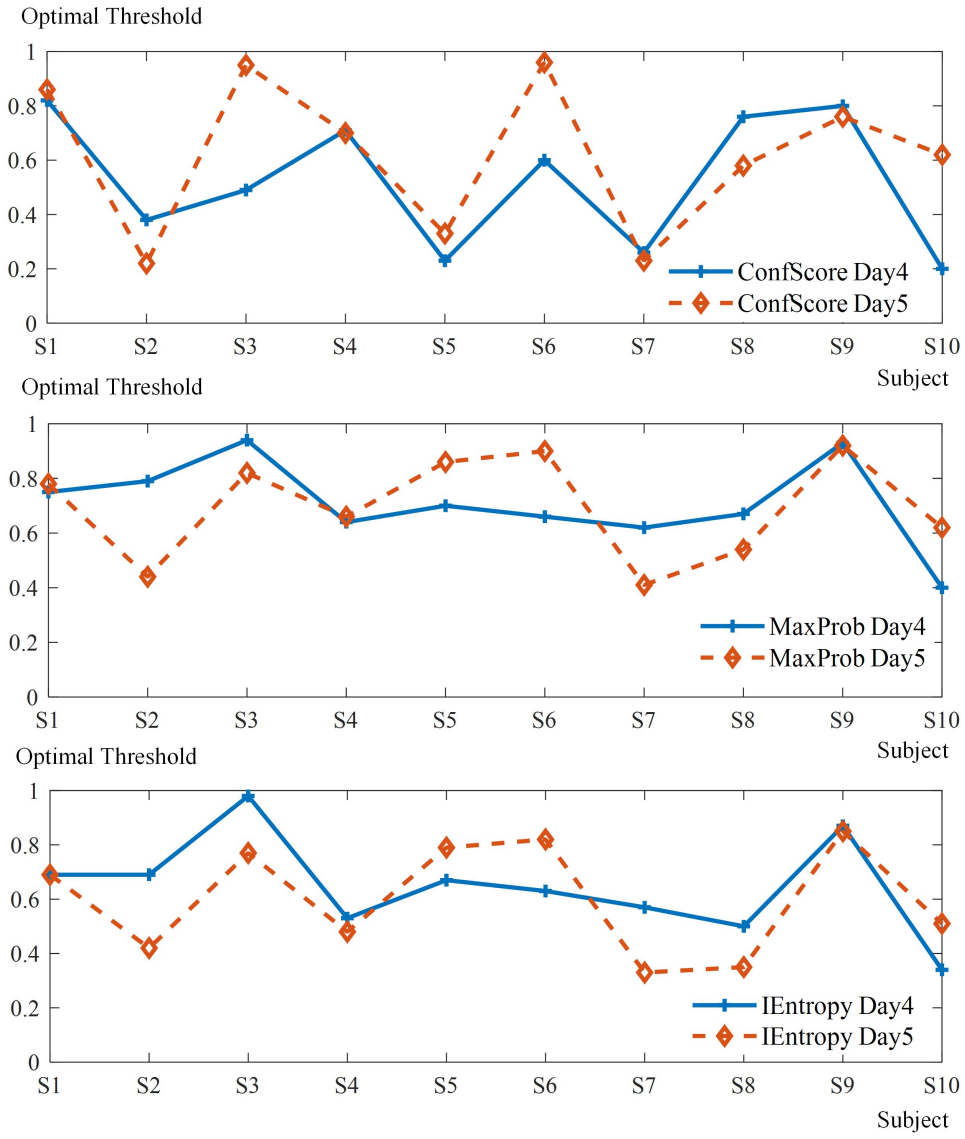


Figure 3.10: Optimal thresholds for Fit curves of three confidence features in Day4 and Day5 of DB6.

determination is usually made empirically based on previous datasets, and this strategy can be affected by the inconsistency of Fit curves between datasets. Figure 3.10 lists the optimal thresholds for three confidence features using data of DB6 (Day4) and DB6 (Day5). Apparently, for most subjects the optimal thresholds in Day4 and Day5 are very different; and the thresholds determined based on previous datasets may result in a poor Fit for the target dataset. Based on these observations, it can be inferred that the rejection robustness, i.e. FitInt, can help to indicate how the Fit value is robust to variations of threshold. As shown in Figure 3.8, FitInt of ConfScore is much larger than the FitInt of MaxProb and IEntropy in most databases. Similarly, previous research [18, 94] also suggested that a desirable confidence characteristic should leave a wider range for threshold adjustment.

An interesting observation in Figure 3.7 is the difference of Fit curves among datasets. Specifically, for each confidence feature, Fit curve of Subject1-DB4 is always higher than the Fit curve of Subject 5-DB6, which indicates a better rejection performance of the former participant. According to Table 3.4, BMEC values of three confidence features in Subject 5-DB6 are smaller than the BMEC values of three confidence features in Subject1-DB4. A main reason is that the CNN classifier is trained and tested using data of different days in DB6, thus the confidence features could become less qualified due to the degradation of CNN accuracy.

3.6 Conclusions

In this chapter, a preliminary attempt is introduced to estimate CNN confidence for rejection-based hand gesture classification in myoelectric control systems. By analysing posterior likelihood of softmax layer, the proposed confidence model can provide ConfScore that is highly related to correctness of CNN predictions. The superiority of ConfScore to two commonly utilised confidence features is fully verified via analysis of BMEC and rejection robustness using data from public databases and our online experiments. With help of confidence-based rejection,

the error rate of CNN can be reduced significantly with small loss of correct classifications, thereby enhancing the model reliability in sEMG-based gesture recognition.

Chapter 4

CNN-LSTM Hybrid Framework for Joint Kinematics Estimation

Although PR scheme has been working as the primary method for ML/DL based myoelectric control, it is inherently different from the natural way of upper-limb movements. To be specific, this strategy cannot simultaneously identify multiple movements when the user voluntarily performs a combined motion [172], and the on/off mode of operation further limits its capacity to support the proportional control [20]. As discussed in Section 2.4, regression techniques have shown considerable potentials to overcome these limitations. However, related efforts on regression-based scheme were mostly conducted using ML approaches, whereas DL approaches have been rarely investigated.

In this chapter, a CNN-LSTM hybrid model is proposed to improve regression performances in joint kinematics estimation. Specifically, CNN is utilised to extract deep features from sEMG spectrum, then these features are processed via LSTM-based sequence regression to estimate wrist kinematics. In this way, the temporal-spatial information in sEMG can be better exploited. Six healthy participants are recruited for the participatory collection and motion analysis under various experimental setups. Estimation results in both intra-session and inter-session evaluations illustrate that CNN-LSTM significantly outperforms CNN,

LSTM and several representative machine learning approaches, particularly when complex wrist movements are activated.

4.1 Introduction

4.1.1 Motivation

In the past decade, regression approaches have gained more and more attentions in myoelectric control. Unlike PR-based methods which discriminate hand gestures in a discrete and sequential manner [173], regression focuses on continuous joint kinematics estimation [174] and thus can enable SPC in multiple DOFs. To further improve regression accuracy and robustness, two DL techniques are now becoming the research interests: 1) CNN for feature extraction from sEMG; 2) RNN for recursive learning over time-series data. Due to the stochastic nature of sEMG and serious cross-talk among muscles in the upper limb, useful information can be easily buried in conventional hand-crafted features. To this end, CNN have been applied to extract features from raw sEMG automatically. For instance, Ameri *et al.* investigated a CNN-based regression technique which outperformed a traditional SVR-based scheme in an online Fittsâ€™ law test [110]. Yang *et al.* presented several data-augmentation approaches for CNN in decoding 3-DOF wrist movements [175], and verified that the proposed CNN structure outperformed SVR significantly when confounding factors were involved [152]. Although CNN is good at extracting spatial correlations of multi-channel sEMG signals, it inherently ignores the temporal information during continuous muscle contractions.

To address the above issue, many researchers begin to implement RNN, especially LSTM, for sEMG-based hand pose estimation. For example, Quivira *et al.* applied LSTM to build an accurate regression model for predicting hand joint kinematics from sEMG features [45]. Teban *et al.* claimed that LSTM performed better than a non-recurrent ANN in replicating a non-linear mechanism of a real human hand [46]. He *et al.* combined LSTM with ANN to exploit both the

dynamic and static information of sEMG [47]. Ali *et al.* validated that a bidirectional LSTM with attention mechanism could outperform other tested RNN variations [48]. Despite that LSTM shows great effectiveness in capturing temporal dependencies based on learning contextual information from past inputs [176], all those pilot studies have only applied conventional hand-crafted features rather than deep spatial features in their regression process. To this end, The combination of CNN and RNN/LSTM is now becoming a trend in sEMG-based motion estimation. In particular, Xia *et al.* [177] proposed a recurrent convolutional neural networks (RCNN) architecture to integrate CNN and LSTM layers for the tracking of arm movements. Huang *et al.* [178] applied a similar architecture in hand gesture classification and verified that the proposed model outperformed ML classifiers in three different exercises of Ninapro Database 2.

4.1.2 Contribution

Inspired by advantages and limitations of CNN and LSTM, in this chapter a CNN-LSTM hybrid model is proposed to combine deep feature extraction and sequence regression efficiently, so that the temporal-spatial correlations of sEMG can be fully exploited. With deep features extracted from CNN and then processed by LSTM, wrist kinematics in single/multiple DOFs can be reconstructed accurately. Compared with conventional CNN and LSTM, CNN-LSTM is more robust to localised distortions along time. In this study, six healthy participants take part in experiments to perform a series of wrist movements. Experimental results indicate that CNN-LSTM outperforms CNN, LSTM, and several representative ML approaches in both intra-session and inter-session scenarios, especially when complex wrist movements are activated in multi-DOFs. Contributions of this chapter can be summarised in three aspects: 1) it firstly investigates the effective combination of CNN and LSTM in wrist kinematics estimation using sEMG signals; 2) a separate training strategy is utilised to improve the computational efficiency and model feasibility; 3) visual explorations of two types of features indicate that distributions of CNN features can be better correlated with wrist motions than many hand-crafted features.

4.1.3 Chapter Organisation

The remainder of this chapter is structured as follows. Section 4.2 describes the proposed hybrid model, where the implementation of deep feature extraction and sequence regression are separately elaborated. Section 4.3 introduces experimental setups and Section 4.4 presents estimation results in both intra-session and inter-session evaluations. In Section 4.5 a discussion is presented to analyse model merits and limitations. The conclusion is then drawn in Section 4.6.

4.2 CNN-LSTM Hybrid Model

As illustrated in Figure 4.1, our CNN-LSTM model consists of two steps: the first step is to implement CNN for feature extraction and the second step is to construct LSTM for sequence regression. In the first step CNN is utilised to extract deep feature vector $\boldsymbol{f}_t$ from the sEMG matrix $\mathbf{X}_t$ which is constructed on the t^{th} segment of multi-channel sEMG signals. In the second step, successive deep feature vectors are rearranged into a series of feature sequences, such as $[\boldsymbol{f}_1, \boldsymbol{f}_2 \cdots \boldsymbol{f}_k]$, $\left[\boldsymbol{f}_2, \boldsymbol{f}_3 \cdots \boldsymbol{f}_{k+1}\right]$, etc. The parameter k is the number of feature vectors in a feature sequence, which denotes the time-steps in recurrent regression. A LSTM is built to convert $[\boldsymbol{f}_1, \boldsymbol{f}_2 \cdots \boldsymbol{f}_k]$ into wrist angles $[\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, \cdots, \hat{\boldsymbol{y}}_k]$. In this study, the last output y_k is adopted as the final observation of this sequence. In the following part the implementation of CNN and LSTM, together with the training process of each model, will be elaborated.

4.2.1 CNN-based Deep Feature Extraction

The basis architecture of CNN is in general consistent with that introduced in Chapter 3. To further exploit the capacity of feature learning in regression tasks, a deeper model is usually adopted [141]. As illustrated in Figure 4.2, the presented CNN consists of 4 convolutional blocks (Conv Block) and 2 fully connected blocks (FC Block). Each Conv Block has a convolutional layer, a batch normalisation layer, a leaky ReLU layer, a max-pooling layer, and a dropout layer.

The convolution layer uses a kernel size of 3, a boundary padding of 1, and the stride of 1. There are 16 kernels in the 1st and 2nd Conv Block whilst 32 in the 3rd and 4th block. The batch normalisation layer is attached to mitigate alternation made by convolutional layers [179]. As suggested in our previous work [5], the leaky ReLU layer is used in case of the dying ReLU problem [180]. The max-pooling layer (a pool size of 3 and a stride of 1) is added for sub-sampling while a dropout layer is attached for regularisation. In each FC Block, the batch normalisation layer, leaky ReLU layer and dropout layer are added subsequently to the fully connected layer. There are 100 hidden units in the 1st FC Block and 20 in the 2nd. Outputs of the 2nd FC Block will be utilised as the deep feature $\boldsymbol{f}$ for LSTM-based sequence regression. Thus the dimension of CNN features is 20.

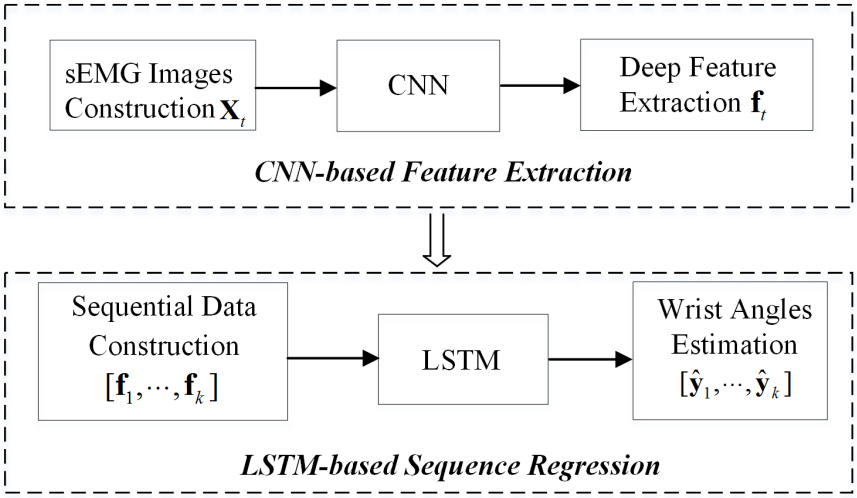


Figure 4.1: Block diagram of CNN-LSTM hybrid model.

4.2.2 LSTM-based Sequence Regression

Topology of LSTM

LSTM is a network designed to encode contextual information of a temporal sequence with feedback loops. It contains cycles that feed the network activation from a previous time-step to influence predictions at the current time-step [181].

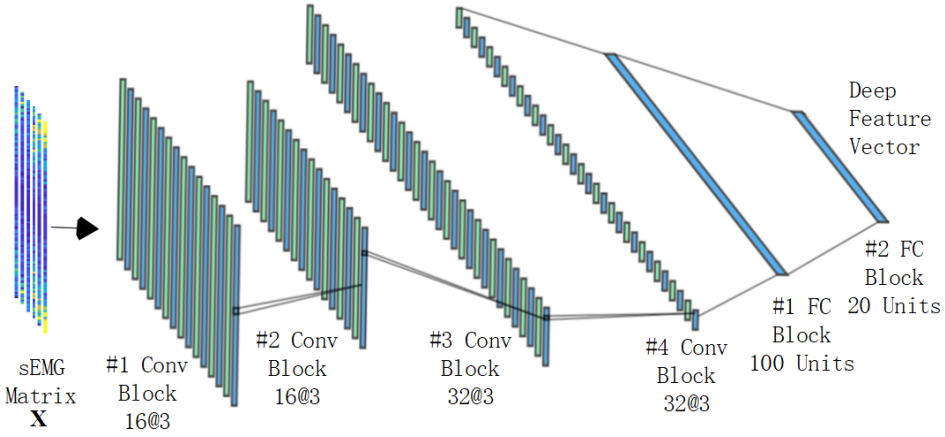


Figure 4.2: The single stream CNN architecture for deep feature extraction.

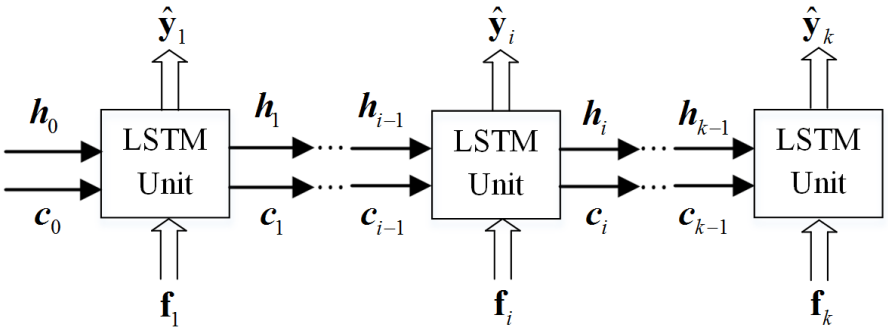


Figure 4.3: The unfolded chain structure of LSTM in time sequence with deep CNN features.

The unfolded chain structure of LSTM in an input sequence $[\boldsymbol{f}_1, \boldsymbol{f}_2 \cdots \boldsymbol{f}_k]$ is illustrated in Figure 4.3 [182], where $\boldsymbol{h}_i$ $(i = 1, 2 \cdots k)$ is the hidden state at time-step i and $\boldsymbol{c}_i$ is the cell state. In recurrent regression, LSTM unit uses previous state $(\boldsymbol{h}_{i-1}, \boldsymbol{c}_{i-1})$ and current feature $\boldsymbol{f}_i$ to update current state $(\boldsymbol{h}_i, \boldsymbol{c}_i)$ and compute wrist angle y_i. In this way the historical information can be passed recursively in the whole loop of LSTM.

Update of LSTM Units

Basic elements of LSTM include an input gate to control activation for the memory cell, a forget gate to drop useless information of the past cell status, and an output gate to control the output activation for the ultimate state. The update of LSTM units at time-step i can be described [183]

$$
\begin{aligned}
\boldsymbol{i}_i &= \delta\left(\boldsymbol{W}_i\left[\boldsymbol{h}_{i-1,}\boldsymbol{f}_i\right]+\boldsymbol{b}_i\right), \\
\boldsymbol{m}_i &= \delta\left(\boldsymbol{W}_m\left[\boldsymbol{h}_{i-1,}\boldsymbol{f}_i\right]+\boldsymbol{b}_m\right), \\
\boldsymbol{o}_i &= \delta\left(\boldsymbol{W}_o\left[\boldsymbol{h}_{i-1,}\boldsymbol{f}_i\right]+\boldsymbol{b}_o\right), \\
\boldsymbol{c}_i &= \boldsymbol{i}_i \odot \tanh\left(\boldsymbol{W}_c\left[\boldsymbol{h}_{i-1,}\boldsymbol{f}_i\right]+\boldsymbol{b}_c\right)+\boldsymbol{m}_i \odot \boldsymbol{c}_{i-1}, \\
\boldsymbol{h}_i &= \boldsymbol{o}_i \odot \tanh\left(\boldsymbol{c}_i\right), \\
\hat{\boldsymbol{y}}_i &= \boldsymbol{W}_y \boldsymbol{h}_i+\boldsymbol{b}_y,
\end{aligned}
\tag{4.1}
$$

where $\boldsymbol{i}_i$ is the input gate, $\boldsymbol{m}_i$ is the forget gate, $\boldsymbol{o}_i$ is the output gate, δ is the logistic sigmoid function, $\boldsymbol{W}$ is the weight matrix in each gate and layer, $\boldsymbol{b}$ is the corresponding bias vector and $\odot$ is the scalar product. The initial state $(\boldsymbol{h}_0, \boldsymbol{c}_0)$ will be settled after model training for subsequent predictions.

4.2.3 Training of CNN-LSTM

Following reference [184], the idea of separate training is adopted herein. Specifically, the tuning of CNN and LSTM is conducted in two subsequent steps. Firstly, a regression layer is attached to the presented CNN architecture to complete a supervised learning. In this step, the model inputs are sEMG matrices and observations are wrist angles. Secondly, deep feature vectors are extracted from a fully connected layer of the well-trained CNN, based on which feature sequences are constructed to train LSTM for sequence regression. Different from structures such as Long-term Recurrent Convolutional Networks (LRCNs) which trains CNN and LSTM jointly [185], our model can be more efficient in model training since the input in each time-step of LSTM is a constant vector rather than convolution operations. Besides, the sequence regression part can be easily

optimised or replaced without re-training the entire model [184].

Training Setting of CNN

Hyper-parameters of presented CNN are mainly identified referring to pilot studies in PR schemes [42] and then determined via empirical manual tuning. As a general setting in this study, the network is trained in a 128-sized mini-batch as employed in [110] for 50 epochs by stochastic gradient descent with momentum (SDGM). The dynamic learning rate of CNN is 0.0001 in initialisation and drops 90% after every 10 epochs. The slope scale is set as 0.1 in all leaky ReLU layers. The dropout rate in each dropout layer is 30%. Other training strategies follow default settings in MATLAB 2018b.

Training Setting of LSTM

In our study the time duration of a regression sequence is set to be 1 second. This achieves a trade-off between the information quantity of temporal dependencies and computational loads in practical implementation. LSTM is trained in a 64 sized mini-batch for 100 epochs via adaptive moment estimation (ADAM). The dynamic learning rate is initialised to be 0.001 and drops 90% after every 10 epochs. Since LSTM is prone to over-fitting more easily than conventional recurrent neural networks, herein only one LSTM layer with 50 hidden units is adopted. A dropout layer with 30% dropout rate is added for regularisation.

4.3 Materials and Experimental Methods

4.3.1 Experiment Setup

Approved by the Mathematics, Physical Science and Engineering joint Faculty Research Ethics Committee of University of Leeds, UK (reference MEEC 18-006), six healthy subjects (five males and one female, aged 24-30) took part in the experiment. The written informed consent was obtained from each subject

before data collection. Following Figure 4.4(a), 12 bipolar electrodes were placed on the proximal portion of the forearm to collect sEMG signals in 6 channels. Reference electrodes were placed near the wrist. The inter-electrode distance in the proximal-distal direction was around 20 mm for reducing the cross-talk effect.

As shown in Figure 4.4(b), in experiments participants were asked to perform four pre-defined wrist movement protocols. They were allowed to quit the experiments in case of any discomfort. The tested hand should be kept in a relaxing state to avoid muscle fatigue, with upper limb supported vertically on the desk and palm facing inside. All motions started from this rest position. Each protocol consisted of 3 sub-trials/sessions, and each session was composed of continuous wrist movements lasting around 3 minutes. A detailed description is reported in Table 4.1.

From Table 4.1 we can see that in P1-P3 only one DOF of the wrist motions was activated to complete single-DOF tasks. On the contrary, P4 aimed at multi-DOF tasks and all 3 DOFs were involved simultaneously. Obviously, P4 is naturally more complex and challenging compared with P1-P3 [186], but it bears closer similarity with real-life movements [103] and can speed up the training process. The frequency of sinusoidal contractions was around 0.1Hz, meaning that a cycle of wrist rotation (such as rest-flexion-rest-extension-rest in P1) was about 10 seconds.

In this study an attitude heading reference system (AHRS), composed of a tri-axial accelerometer, gyroscope and magnetometer, was utilised to obtain hand orientation [187]. Wrist angles, which worked as the ground-truth in supervised learning, were calculated based on Euler angles from AHRS. Referring to Figure 4.4 (b), both sEMG signals and wrist movements were recorded simultaneously with Shimmer wearable sensors [167] attached on the back of the testing hand. Sampling rates for accelerometer, gyroscope, magnetometer, and sEMG were set as 100Hz, 100Hz and 75Hz, and 1024Hz respectively. The online data streaming was implemented in a home-made software based on Shimmer MATLAB Instrument Driver [167].

(a) (b)

Figure 4.4: Experiment setup [5]. (a) Electrodes placement. (b) Data acquisition.

4.3.2 Data Pre-processing

Raw sEMG signals obtained from the front-end acquisition equipment are generally noisy because of electromagnetic radiation, motion artefact, and instability of signals, etc. [188]. In particular, noise produced by motion artefact is in the range of 0-20Hz. Thus signals below 20Hz are unstable and cannot provide a reliable contribution to sEMG [189]. A simple method is to filter it out with a high-pass filter. In addition, since energy of sEMG is mainly blow 450Hz, a low-pass filter is often used to remove unwanted noise with high frequencies [28]. As suggested by these studies, a 3^{rd} order Butterworth band-pass filter (20-450Hz) was applied to reduce noise of raw sEMG. A Min-Max scaling was applied to normalise sEMG in each channel [190]. As for data segmentation, the analysis window was set to be 100ms with increment of 50 ms. Thus the size of sEMG matrix $(1 \times L \times C)$ was $1 \times 101 \times 6$ in our experiments. Since the time duration of a feature sequence was set to be 1 second empirically, there were 18 time-steps in $[\boldsymbol{f}_1, \boldsymbol{f}_2 \cdots \boldsymbol{f}_k]$, i.e. $k = 18$.

4.3.3 Model Evaluation

The analysis of sEMG-based wrist kinematics estimation was composed of intra-session and inter-session evaluations. To implement intra-session evaluations, the

Table 4.1: List of Performed Contractions.

Protocol	Description	Active DOF
P1	Sinusoidal contractions	Flexion-extension (F-E)
P2	Sinusoidal contractions	Pronation-supination (P-S)
P3	Sinusoidal contractions	Radial-ulnar deviation (R-U)
P4	Co-contractions of the wrist	F-E+P-S+R-U

data in one session/trial of each protocol was split into four folds, where the first three were used for model training and the last for testing. To avoid data leakage, the splitting should be conducted before data pre-processing. In inter-session evaluations one whole session was used for model training and another session in the same protocol was used for testing. This method could better validate the model robustness against time-dependent changes of sEMG signals. The number of training samples extracted by sliding windows is around 1500-2000 for intra-session evaluations and 2000-3000 for inter-session evaluations.

In this study the coefficient of determination (R^2) [191] was used as the metric to quantify the regression performances. The mathematical expression of R^2 is

$$R^2 = 1 - \frac{\sum_d \mathrm{Var}\left(\boldsymbol{y}^d - \hat{\boldsymbol{y}}^d\right)}{\sum_d \mathrm{Var}\left(\boldsymbol{y}^d\right)}, \tag{4.2}$$

where $\boldsymbol{y}^d$ are measured wrist angles in d^{th} DOF and $\hat{\boldsymbol{y}}^d$ are model estimations. According to Eq. 4.2, the numerator of R^2 is the mean squared error (RMSE) which is normalised by the variance of correct labels in the denominator [102]. Compared with RMSE, R^2 is more robust to the numerical range of labels. R^2 at perfect estimation is equal to one, whilst a negative value means that estimation errors are larger than the variance of target values.

4.4 Experimental Results

4.4.1 Visual Exploration of sEMG Features

Visual exploration allows intuitive analysis of the distributions or potential correlations between certain variables. In this section, t-Distributed Stochastic Neighbour Embedding (t-SNE) is utilised to project extracted CNN features (in testing sets) into two principal dimensions for visualisation [192]. For comparison, a widely applied temporal-spatial feature set [26, 99, 193] consisting of MAV, RMS, VAR, and 4^{th} AR are calculated. Details of these hand-crafted features can be found in Chapter 2.2.2. Scatter plots of projected sEMG features in P1 (F-E), P4 (F-E) and P4 (P-S) of intra-session evaluations are shown in Figure 4.5, where the two axes represent two principal features, respectively. The angles of scatters (features) are reflected in parula colormap, with the pure yellow representing the positive maximal values in one DOF and pure purple for the negative maximum.

From Figure 4.5 we can see that in each dataset the clustering of scatters projected from CNN features is significantly better than that of hand-crafted features. In the left part of each sub-figure, scatters with similar colour are gathering whilst those with different colours are highly distinguishable. On the contrary, scatters in the right one are overlapped heavily, even among the yellow ones and the blue ones. Compared with P1 (F-E), the clustering of scatters becomes worse for hand-crafted features in P4 (F-E). This deterioration becomes more evident in P4 (P-S), where distributions among scatters from CNN features become also ambiguous. A possible reason for the deterioration is that the cross-talk of sEMG can be quite serious in multi-DOFs tasks due to our forearm anatomy [26]. Since muscle fibres of extensors and flexors are much thicker and also located in a more superficial layer of the forearm, information of other DOFs are easier to be buried in compounded sEMG.

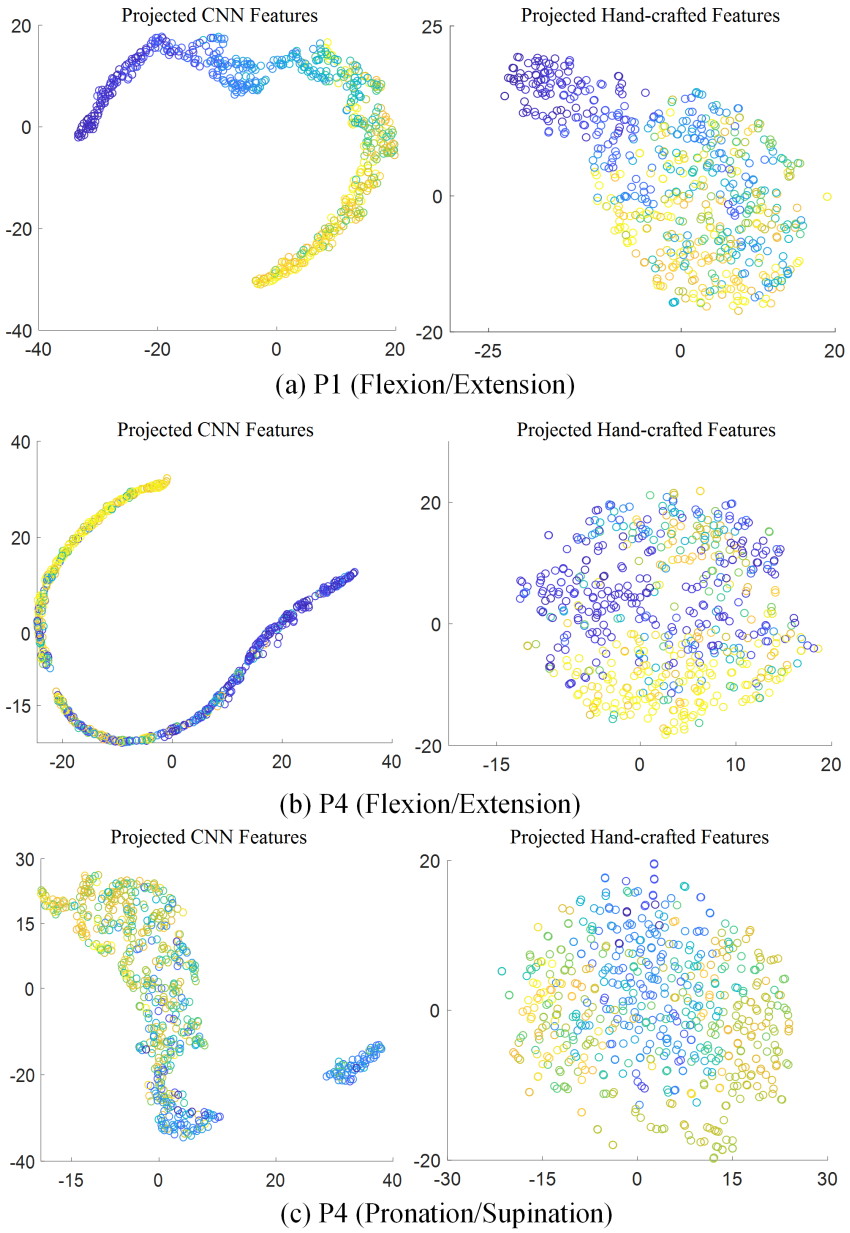


Figure 4.5: Distribution of CNN features and hand-crafted features in testing sets of Subject 5 after dimension reduction. Scatters in (a)-(c) correspond to features from P1(F-E), P4(F-E), and P4(P-S), respectively.

4.4.2 Intra-session Estimations in Single-DOF Tasks

Figure 4.6 shows wrist angles captured by AHRS system in P1-P3 of Subject 5 together with estimations of CNN and CNN-LSTM. As illustrated in the figure, trajectories reconstructed by CNN-LSTM (in red) are smoother and much closer to the ground-truth (in blue) than CNN trajectories (in yellow) in all tasks. This is because the history information of successive deep feature vectors in a sequence is further exploited by CNN-LSTM, which improves estimation accuracy significantly. Another interesting result is that the estimated trajectories of both CNN and CNN-LSTM in P1 are better than their corresponding results in P2 and P3. As shown in visual exploration, feature scatters in the F-E DOF are much more distinguishable than those in the other two DOFs.

In this section two representative ML models, i.e., SVR and RF, are implemented to compare with DL techniques. SVR can project sEMG features into a higher dimensional space via kernel functions, whilst RF is currently the most popular ensemble learning technique. The outperformance of SVR and RF over other shallow models such as LR and ANN have been verified in pilot studies [102, 113]. To evaluate the effectiveness of CNN features for sequence learning, the conventional LSTM with hand-crafted features is also applied for comparison. Same to visual exploration, MAV, RMS, VAR, and 4^{th} AR are working as hand-crafted features for ML models. To be consistent with CNN-LSTM, the dimension of hand-crafted features was reduced to 20 using PCA. Following previous studies [102], a radial basis function (RBF) is adopted for SVR. Besides, the hyper-parameters of SVR and RF were optimised via 5-fold inner cross-validation.

Table 4.2 summarises intra-session performances of SVR, RF, CNN, LSTM, and CNN-LSTM in P1-P3 of Subject 1-6. As we can see, the presented hybrid model outperforms other models in all trials of all protocols. The outperformance can be more evident in some datasets, such as P2 and P3 in nearly all participants. In addition, by exploiting the correlations among adjacent sEMG samples, performances of conventional LSTM are also better than SVR/RF/CNN in most cases. In protocol P1, the average R^2 values of LSTM and CNN-LSTM are closer, whereas in other protocols LSTM is evidently inferior to CNN-LSTM.

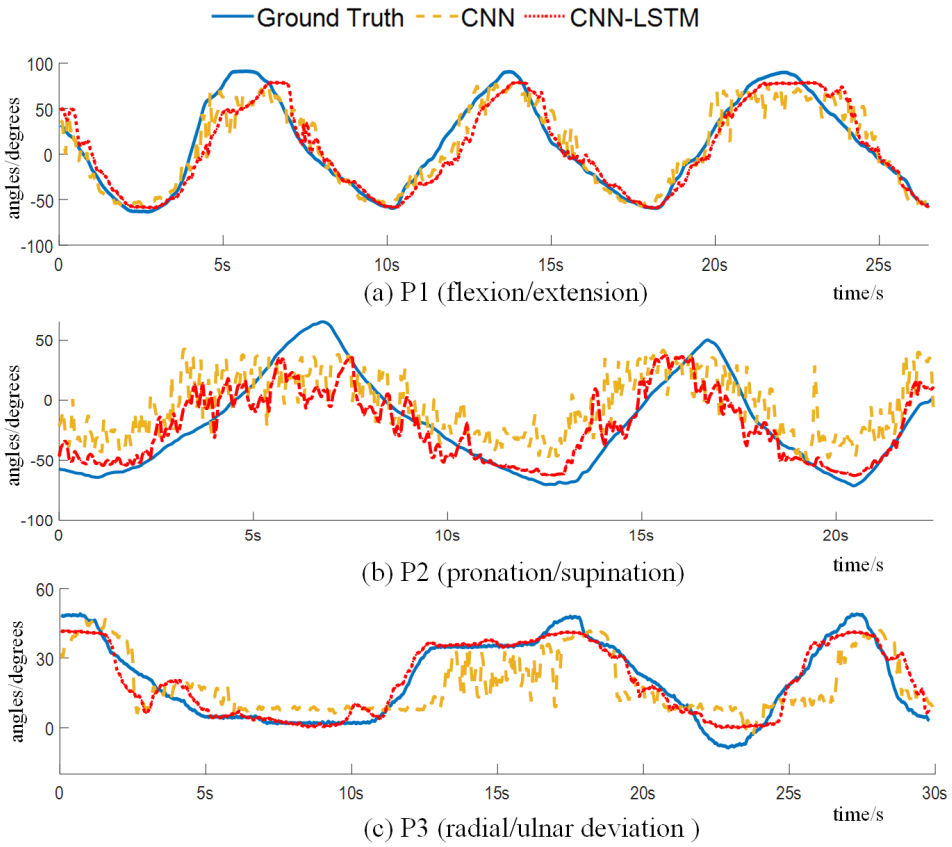


Figure 4.6: Wrist motions and intra-session estimations of CNN and CNN-LSTM for P1-P3 in Subject 5.

A possible reason is the deterioration of hand-crafted features can be more serious then CNN features in these two DOFs (P-S and R-U). A more detailed verification can be found in the visual exploration (Section 4.4.1). Besides, the conventional single-stream CNN is in general comparable to SVR and RF in sEMG-based wrist kinematics estimation. This result is similar to pilot studies in pilot studies [42].

4.4.3 Intra-session Estimations in Multi-DOF Tasks

Different from single-DOF tasks (P1-P3), the multi-DOF task (P4) requires co-activation of 3 DOFs. Figure 4.7 demonstrates the intra-session estimations of CNN and CNN-LSTM in P4 of Subject 5. In accordance with single-DOF tasks,

Table 4.2: R^2 of SVR, RF, CNN, LSTM, and the proposed hybrid model in Single-DOF Tasks (P1-P3) of Intra-session Evaluations.

Subjects	Protocols	SVR	RF	CNN	LSTM	Proposed
1	P1(F-E)	0.56	0.71	0.56	0.84	***0.92***
	P2(P-S)	0.26	0.28	0.32	0.53	***0.65***
	P3(R-U)	0.56	0.59	0.66	0.67	***0.87***
2	P1(F-E)	0.60	0.69	0.66	0.81	***0.85***
	P2(P-S)	0.37	0.48	0.45	0.53	***0.56***
	P3(R-U)	0.22	0.25	0.31	0.46	***0.64***
3	P1(F-E)	0.35	0.38	0.42	0.7	***0.80***
	P2(P-S)	0.46	0.63	0.58	0.75	***0.83***
	P3(R-U)	0.18	0.19	0.22	0.52	***0.56***
4	P1(F-E)	0.35	0.40	0.41	0.67	***0.75***
	P2(P-S)	0.17	0.16	0.21	0.34	***0.46***
	P3(R-U)	0.40	0.48	0.43	0.66	***0.88***
5	P1(F-E)	0.84	0.86	0.84	0.82	***0.91***
	P2(P-S)	0.51	0.52	0.62	0.65	***0.71***
	P3(R-U)	0.59	0.71	0.67	0.80	***0.90***
6	P1(F-E)	0.71	0.76	0.74	0.81	***0.91***
	P2(P-S)	0.21	0.30	0.36	0.42	***0.64***
	P3(R-U)	0.36	0.32	0.40	0.51	***0.69***
Average	P1(F-E)	0.57	0.63	0.61	0.78	***0.86***
	P2(P-S)	0.33	0.40	0.42	0.54	***0.64***
	P3(R-U)	0.39	0.42	0.45	0.60	***0.76***

the reconstructed trajectories of CNN-LSTM are much closer to the ground-truth in all DOFs. As for R^2 values, CNN-LSTM reaches higher scores than other four models, indicating an evident improvement in model accuracy. R^2 values of each DOF in six subjects are listed in Table 4.3. Same to P1-P3, performances of CNN, RF, and SVR are in general close to each other, whilst LSTM outperforms

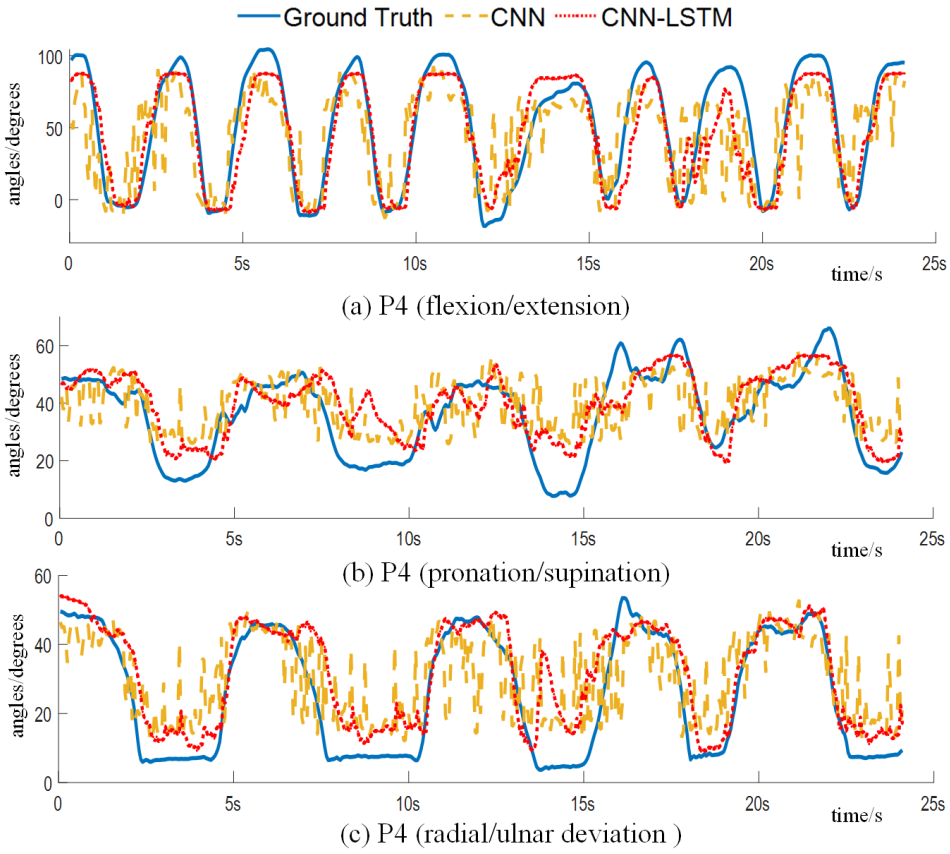


Figure 4.7: Wrist motions and intra-session estimations of CNN and CNN-LSTM in P4 of Subject 5.

these non-recurrent regression techniques in most cases. Consistent with results in visual exploration, deterioration in estimation accuracy can be found in each DOF of P4 compared with those in P1-P3, indicating that the features of samples become harder to recognise in the multi-DOF tasks.

4.4.4 Inter-session Estimations in Single/Multiple DOFs Tasks

Figure 4.8 illustrates the inter-session performance of CNN and CNN-LSTM in P1-P3 of Subject 5. Performances of both CNN and CNN-LSTM become a little

Table 4.3: R^2 of SVR, RF, CNN, LSTM, and the proposed hybrid model in Multi-DOF Tasks (P4) of Intra-session Evaluations.

Subjects	Protocols	SVR	RF	CNN	LSTM	Proposed
1	F-E	0.44	0.52	0.55	0.78	***0.87***
	P-S	0.30	0.31	0.30	0.51	***0.58***
	R-U	0.40	0.38	0.39	0.61	***0.69***
2	F-E	0.63	0.63	0.62	0.73	***0.82***
	P-S	0.19	0.27	0.28	0.39	***0.47***
	R-U	0.37	0.36	0.40	0.48	***0.61***
3	F-E	0.35	0.44	0.46	0.67	***0.70***
	P-S	0.37	0.39	0.50	0.49	***0.70***
	R-U	0.15	0.30	0.27	0.36	***0.42***
4	F-E	0.51	0.43	0.44	0.62	***0.67***
	P-S	0.21	0.23	0.25	0.40	***0.40***
	R-U	0.53	0.52	0.55	0.65	***0.73***
5	F-E	0.49	0.54	0.50	0.81	***0.86***
	P-S	0.31	0.37	0.40	0.59	***0.65***
	R-U	0.44	0.49	0.54	0.70	***0.83***
6	F-E	0.63	0.69	0.73	0.78	***0.89***
	P-S	0.25	0.29	0.34	0.43	***0.53***
	R-U	0.66	0.65	0.55	0.68	***0.74***
Average	F-E	0.51	0.54	0.55	0.73	***0.80***
	P-S	0.27	0.31	0.35	0.47	***0.56***
	R-U	0.43	0.45	0.45	0.58	***0.67***

bit worse compared to intra-session evaluations in Figure 4.6 due to domain shifts among different sessions, but the curves reconstructed by CNN-LSTM still manage to match the ground-truth. Figure 4.9 illustrates the comparison among all regression techniques following P1-P4 of Subject 5, in which the outperformance of CNN-LSTM are still evident. As for wrist motions in flexion and extension,

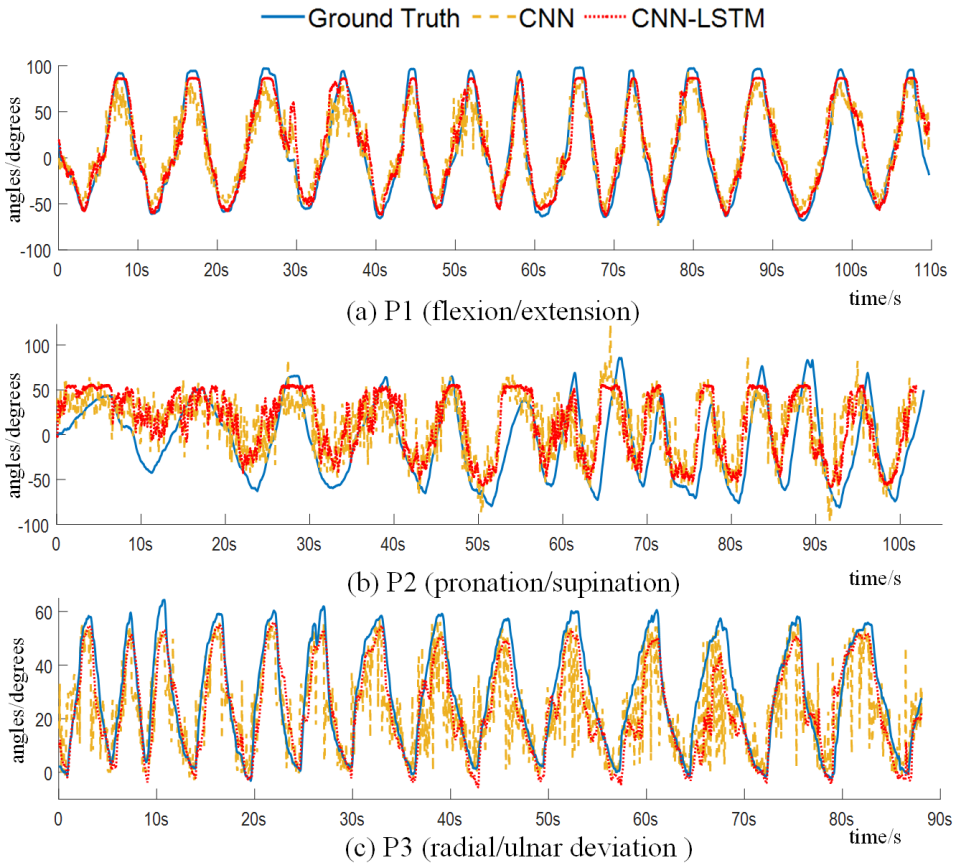


Figure 4.8: Inter-session estimations of CNN and CNN-LSTM following P1-P3 of Subject 5.

R^2 values of CNN-LSTM can be as high as 0.93 and 0.74 in new testing sessions of P1 and P4 (F-E), respectively, indicating a reliable proportional myoelectric control in this DOF. Besides, promising accuracy can be achieved by ML models in P1 (SVR and RF reach 0.73 and 0.79, respectively). As is discussed in visual exploration, the higher accuracy in F-E are mainly caused by the upper limb anatomy, which on the other hand leads to non-negligible cross-talks for sEMG of other two DOFs.

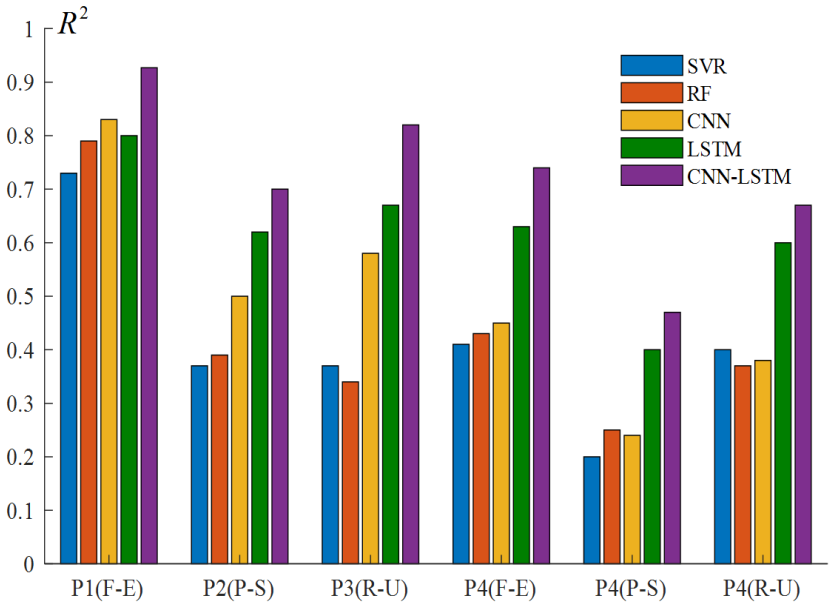


Figure 4.9: Inter-session evaluations of SVR, RF, CNN, LSTM, and the proposed CNN-LSTM in P1-P4 of Subject 5.

4.4.5 Comparison of Time-steps in CNN-LSTM

The time-step k in feature sequence $[\boldsymbol{f}_1, \boldsymbol{f}_2 \cdots \boldsymbol{f}_k]$ determines the number of sEMG samples to be included in sequence regression. A larger k denotes a longer term of time dependencies which may contribute to a higher accuracy but also results in a heavier computational load. In this subsection four different time-steps are evaluated for CNN-LSTM, i.e. 8, 18, 58, 98 for the value of k which correspond to 0.5s, 1s, 3s and 5s in time duration, respectively. Estimation results in inter-session evaluations are illustrated in Figure 4.10. In general, the R^2 of CNN-LSTM improves gradually along with the increase of time-steps, indicating that the exploitation of long-term time dependencies contributes to a higher estimation accuracy in most scenarios. Empirically, a sequence in 1s duration can reach a compromise in model effectiveness and efficiency. Besides, a too large sequence is inapplicable for real-time myoelectric control since the intention prediction is expected to be implemented without evident time delays.

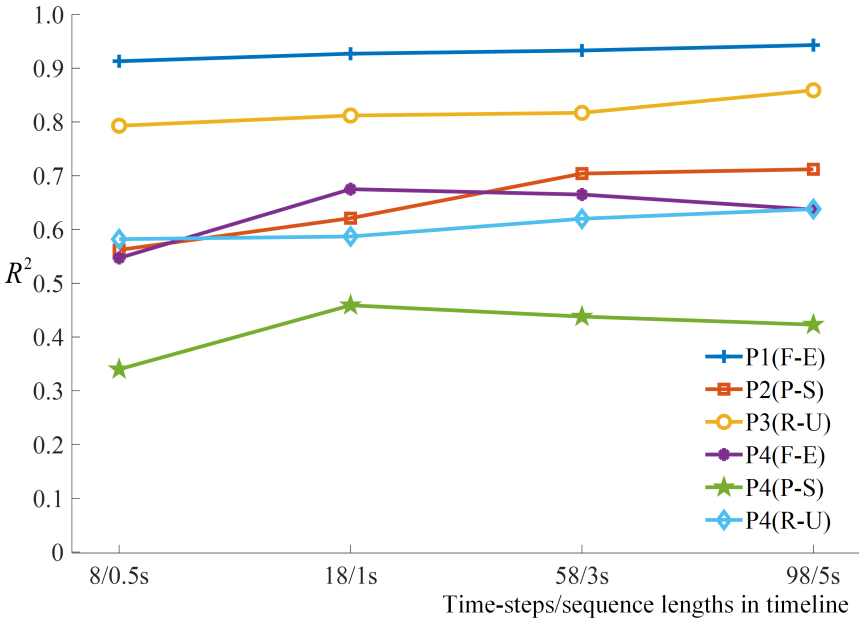


Figure 4.10: Comparison between time-steps/sequence lengths of CNN-LSTM in inter-session evaluations of Subject 5.

4.4.6 Comparison of sEMG Matrices

Besides the architecture and hyper-parameters, sEMG input matrices also have a non-negligible impact on CNN-based feature extraction and can then influence the estimation accuracy of CNN-LSTM. In Section 4.2, the spectrum-based sEMG matrices is obtained by applying FFT on each sliding window. A more intuitive method is to construct matrices in the time domain directly. The comparison of CNN and CNN-LSTM with temporal and spectral sEMG matrices in intra-session evaluations can be found in Figure 4.11. For simplicity, CNN/CNN-LSTM with temporal or spectral inputs are shorted as CNNt, CNNs, CNN-LSTMt and CNN-LSTMs, respectively. It can be observed that CNNs outperforms CNNt in all protocols, which contributes to the outperformance of CNN-LSTMs over CNN-LSTMt accordingly. This superiority becomes more significant in multi-DOF tasks. A possible reason is that the sEMG collected by sparse electrodes can be regarded as the superimposition of signals from multiple muscles. Dur-

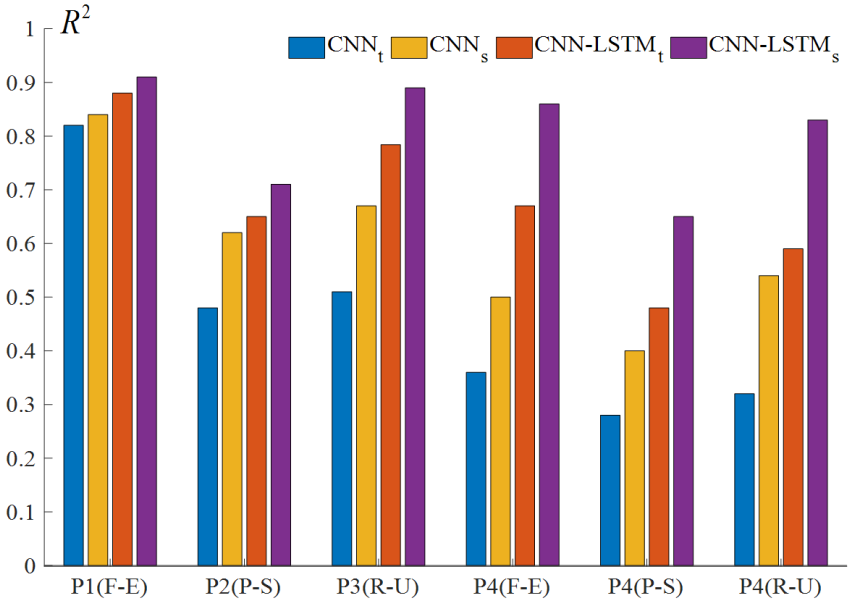


Figure 4.11: Intra-session evaluations of CNN and CNN-LSTM with two types of sEMG matrices. CNN/CNN-LSTM with temporal or spectral inputs are shorted as CNNt, CNNs, CNN-LSTMt, and CNN-LSTMs, respectively.

ing voluntary contractions, the firing rates of motoneuron in these muscles are different [194], thus the spectrum information can be more representative and distinguishable.

4.4.7 Comparison of Deep Feature Dimensions

As shown in Figure 4.2, the sEMG matrix is imported into a pre-trained CNN and the vector extracted from the 2nd FC Block works as the CNN feature for sequence learning in each time-step. Since there are 20 hidden units in this layer, the dimension of deep features is 20. Compared with CNN architectures in many previous studies [41, 42, 152], the dimension of our last FC Block is smaller since it is empirically found that a too large dimension might not be able to benefit the performances of CNN and CNN-LSTM significantly. Table 4.4 illustrates the R^2 values of CNN and CNN-LSTM when using different number of hidden units

Table 4.4: R^2 of CNN and CNN-LSTM when using different number of hidden units in the 2nd FC layer of CNN.

Protocols	Models	Number of hidden units in the 2nd FC layer					
		2	5	10	20	50	100
P1 (F-E)	CNN	0.40	0.66	0.78	0.82	0.83	0.81
	CNN-LSTM	0.79	0.85	0.88	0.93	0.91	0.91
P2 (P-S)	CNN	0.33	0.41	0.45	0.48	0.49	0.51
	CNN-LSTM	0.60	0.65	0.68	0.69	0.70	0.69
P3 (R-U)	CNN	0.42	0.48	0.51	0.57	0.60	0.61
	CNN-LSTM	0.77	0.80	0.80	0.82	0.83	0.84
P4 (F-E)	CNN	0.32	0.38	0.40	0.44	0.43	0.44
	CNN-LSTM	0.70	0.73	0.74	0.76	0.75	0.77
P4 (P-S)	CNN	0.15	0.22	0.24	0.25	0.23	0.26
	CNN-LSTM	0.40	0.42	0.46	0.45	0.44	0.48
P4 (R-U)	CNN	0.25	0.31	0.34	0.37	0.37	0.39
	CNN-LSTM	0.63	0.65	0.67	0.68	0.67	0.68

in the 2nd FC layer, which indicates that in our experiments 20 dimensions can be a good choice for both two models.

4.5 Discussion

To further improve regression accuracy and robustness, both CNN and LSTM are now becoming prevalent in sEMG-based motion estimation. In this paper, a hybrid model is presented to combine these two techniques, i.e. CNN for automatic feature extraction and LSTM for sequence regression, such that the temporal-spatial correlations in sEMG signals can be extracted more efficiently. Conventional ML techniques rely deeply on manual feature extraction and selection. This process requires good domain knowledge or experience, and useful information may be easily buried in hand-crafted features. On the contrary, CNN extracts

features from raw sEMG directly and automatically by learning the signal characteristics via layer-by-layer processing. The convolution operations also enables CNN to extract spatial correlations of sEMG signals from multi-channels. As mentioned in many previous works, CNN features can be useful to represent patterns of muscle activation, and the automatic feature extraction can help to reduce the information loss. Visual explorations of two types of features (details can be found in Chapter 4.4.1) indicate that distributions of CNN features can be better correlated with wrist motions than many hand-crafted features. This should be the reason why CNN-LSTM can outperform conventional LSTM which only uses hand-crafted features.

Secondly, SVR, RF, and CNN are all non-recurrent models, which inherently ignore the temporal dependencies of successive sEMG samples. In fact, during continuous muscle contractions there are supposed to be strong temporal-dependencies in sEMG signals. Thus it is reasonable to consider sEMG as time-series data in regression tasks. In this study the feature sequences $[\boldsymbol{f}_1, \boldsymbol{f}_2 \cdots \boldsymbol{f}_k]$ are reconstructed for LSTM to further exploit the history information of successive deep feature vectors. From previous literature [46] it can inferred that the recurrent networks such as LSTM have shown superiority to many non-recurrent models. Our experiment results also demonstrate the outperformance of recurrent architectures. Therefore, by efficiently extracting the temporal-spatial correlations in sEMG signals, CNN-LSTM further improves regression accuracy in both single and multiple DOF tasks.

A main limitation of our current method is the model generalisation in multi-days and multi-subjects. Due to the non-stationary characteristics of sEMG signals, it is reported that classification/regression performances could decrease substantially over time [22]. Besides, sEMG signals have a user-dependent nature, causing recordings to differ even when signals are measured from the same location with the same motion. Therefore, a pre-trained model may not be able to perform accurately in a new subject. These issues can be summarised as the domain shift problems in machine learning applications, since data-driven methods rely on the assumption that training and testing data should stem from same

underlying distributions. To this end, domain/rule adaptation approaches can be further investigated to improve the generalisation of CNN-LSTM. Besides, more volunteers are going to be recruited for a better verification of our method.

4.6 Conclusions

To enhance the performances of DL techniques in wrist kinematics estimation, a hybrid framework was presented to combine CNN-based feature extraction and LSTM-based sequence regression, which could extract temporal-spatial correlations in sEMG efficiently. Through visual exploration, it is verified that deep features extracted by CNN were more representative than traditional hand-crafted features. By exploiting contextual information in deep features, the proposed CNN-LSTM outperformed conventional CNN, LSTM as well as representative ML approaches in both intra-session and inter-session evaluations.

Chapter 5

LSTM-based Kalman Filter For Sequence Regression

Although CNN-LSTM hybrid framework has shown considerable advances in joint kinematics estimation by simultaneously exploiting the temporal and spatial information of sEMG, its performances still suffer a lot from the noisy measurements that are common in the detection of sEMG signals. The main reason is that DL, as a type of data-driven technique, only intends to learn the relationship between sEMG and target kinematics. Therefore, the prior knowledge of the system, such as the characteristics of wrist rotations in regression task, is normally ignored. By contrast, Kalman filter is a model-based method that recursively provides current estimate using the previous estimate and the most recent observation. It applies Kalman gain to combine the internal transition model and the observation model effectively. Nevertheless, it is hard to pre-determine parameters of transition/observation model in Kalman filters.

In this chapter, LSTM-KF is proposed to further enhance the sequence regression on high-level features extracted by CNN. Specifically, LSTM-KF adopts the computational graph of Kalman filter but estimates parameters of the transition/observation model and the Kalman gain from data using LSTM modules. With this process, advantages of Kalman filter and LSTM can be exploited jointly.

For the sake of demonstration, the two-step hybrid structure, i.e. CNN+LSTM-KF, is named as deep Kalman filter network (DKFN). Experimental results demonstrate that the proposed DKFN can further outperform CNN-LSTM in both wrist and fingers kinematics estimation.

5.1 Introduction

5.1.1 Motivation

Conventional DL techniques, including CNN and RNN, can neither include prior knowledge of the system nor capture the uncertainty information. Therefore, their performances are easy to suffer from noisy measurements which are quite common in neural activities [195]. Apart from data-driven techniques, Kalman filter has also been employed to infer limb movements from neural recordings [196, 197]. To estimate the states of a system throughout time, Kalman filter incorporates the information both from the internal process and actual measurement. Therefore, this technique is especially useful when the states are not directly observable or the measurement is very noisy [195]. In particular, a Kalman gain is used to determine the weights of a internal transition model and an observation model. If the internal model is more accurate, then a larger weight will be given for the computation of final estimation, and vice versa for the measurement model. In this way, it can provide a better estimation based on noisy data.

However, Kalman filter could not handle the non-linear relationship between the neural activity and limb movements. Besides, it is hard to pre-determine the value of parameters in transition and measurement models. To overcome such limitations, many attempts have been made to directly learn models from training data with LSTM. For example, Coskun *et al.* firstly proposed the LSTM-KF framework which integrated three LSTM modules into the Kalman filter to learn the transition, observation and noise models in pose estimation tasks [198]. Zhao *et al.* proposed a Learning Kalman Network (LKN) to filter the car trajectory given a sequence of measurements. In this network, both the observation model

and transition model were learned by feed-forward neural networks, whilst the Kalman Gain iteration was enhanced by a LSTM module [199]. More recently, Ju *et al.* designed a Interaction-aware Kalman Neural Networks (IaKNN), in which they incorporated two LSTM modules for learning the time-varying process and measurement noises for the update step of the Kalman filter [200]. Nonetheless, all these studies were conducted in computer-vision tasks, such as human pose reconstruction or trajectory prediction of self-driving cars. To our best knowledge, there has yet any research on sEMG-based motion estimation.

5.1.2 Contribution

In this chapter, a novel variation of CNN-LSTM framework, i.e. DKFN, is presented to estimate hand kinematics more accurately. DKFN firstly utilises CNN to extract high-level features from raw sEMG signals, and then employs LSTM-KF to conduct sequence regression of CNN features. The main contribution lies in the design of LSTM-KF which integrates LSTM into Kalman filter for recursive learning, such that the advantages of Kalman filter and LSTM can be exploited jointly. In particular, LSTM-KF adopts the computational graph of Kalman filter but estimates parameters of the transition/observation model and the Kalman gain from data using LSTM modules. To validate the effectiveness of DKFN, eight healthy subjects were recruited to perform continuous wrist movements. Moreover, a public benchmark is further utilised to verify the generalisation of DKFN in mapping sEMG to finger kinematics. Experimental results demonstrate that DKFN can outperform CNN and CNN-LSTM in the sequence regression for both wrist and finger kinematics estimation.

5.1.3 Chapter Organisation

The remainder of this chapter is structured as follows. Section 5.2 describes the pipeline of DKFN. The application of Kalman filter in myoelectric control is firstly introduced, and then the design of LSTM-KF is described elaborately. Section 5.3 introduces experimental setups of wrist/finger kinematics estimation,

and section 5.4 presents experimental results in these two tasks. The chapter is concluded in section 5.5.

5.2 Methodology

5.2.1 Overview

As shown in Figure 5.1, DKFN also takes the two-step structure utilised by CNN-LSTM, i.e. a CNN for deep feature extraction and a LSTM-KF for sequence regression. Herein, the proposed LSTM-KF follows the computational graph of Kalman filter and estimates parameters of the transition/observation model from data using LSTM modules. Thus LSTM-KF works recursively over time and produces a sequence of outcomes, i.e. $[\boldsymbol{s}_1, \cdots, \boldsymbol{s}_k]$ for CNN features $[\boldsymbol{f}_1, \cdots, \boldsymbol{f}_k]$ accordingly, where k denotes the sequence length. The final state $\boldsymbol{s}_k$ is taken as the regression output of this sample. In order to estimate kinematics in every time step, the increment between training sequences is set to be one feature vector. The design of CNN-based extraction is consistent with that in Chapter 4.2.1. In the following part, the design of LSTM-KF will be elaborated.

5.2.2 Design of LSTM-KF

For the sake of demonstration, the typical workflow of Kalman filter to decode upper limb kinematics from neural activities [195, 196] is firstly introduced. Specifically, the system dynamics is formulated as a transition model and an observation model:

$$\boldsymbol{s}_i = \boldsymbol{A}\boldsymbol{s}_{i-1} + \boldsymbol{w}, \tag{5.1}$$

$$\boldsymbol{f}_i = \boldsymbol{H}\boldsymbol{s}_i + \boldsymbol{v}, \tag{5.2}$$

where $\boldsymbol{f}_i$ denotes the extracted feature of neural signals in the i^{th} ($i \in [1, k]$) time step and $\boldsymbol{s}_i$ is the kinematic state of the hand. $\boldsymbol{A}$ is the state transition matrix, $\boldsymbol{H}$ is the measurement transformation matrix, $\boldsymbol{w}$ and $\boldsymbol{v}$ are the state

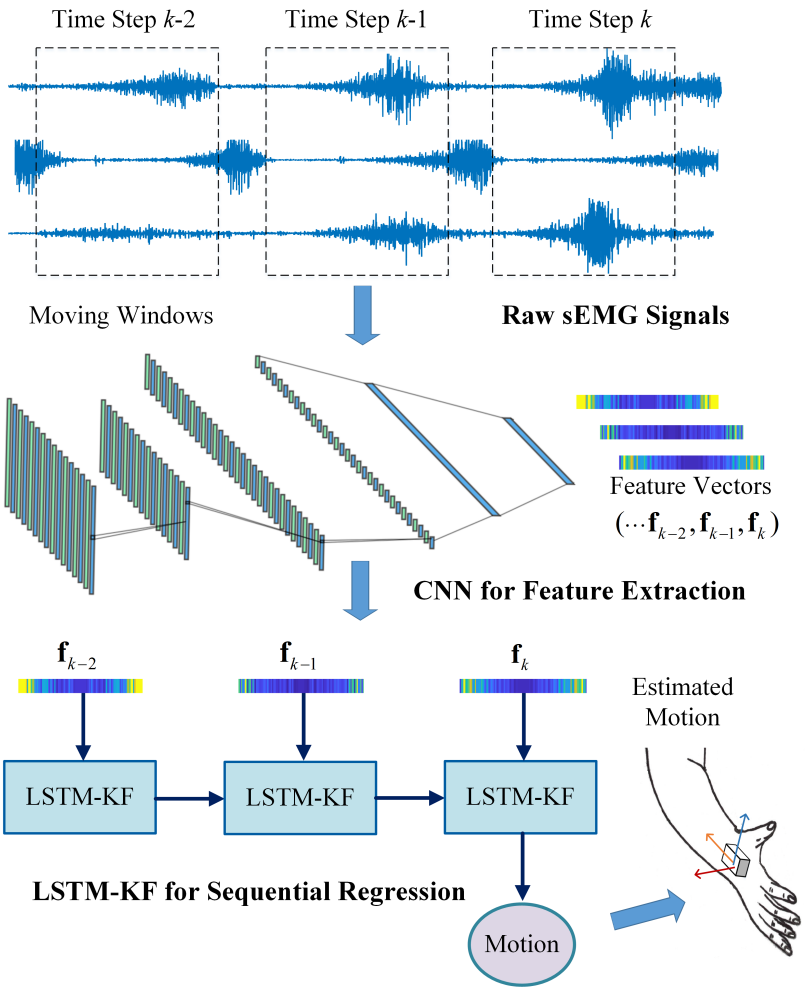


Figure 5.1: Pipeline of DKFN in hand motion estimation using sEMG. It is noted that DKFN is composed two parts, i.e. a CNN for deep feature extraction and a LSTM-KF for sequence regression.

and measurement uncertainties drawn from Gaussian distributions $N(\mathbf{0}, \boldsymbol{Q})$, and $N(\mathbf{0}, \boldsymbol{R})$.

With these parameters determined, Kalman filter utilises an iterative feedback loop with a prediction step and an update step to fuse two models. In the prediction step, the mean and covariance of the current state are estimated by

$$\hat{\boldsymbol{s}}_i = \boldsymbol{A}\boldsymbol{s}_{i-1}, \tag{5.3}$$

$$\hat{\boldsymbol{P}}_i = \boldsymbol{A}\boldsymbol{P}_{i-1}\boldsymbol{A} + \boldsymbol{Q}, \tag{5.4}$$

where $\hat{\boldsymbol{s}}_i$ is the prior state estimate for the i^{th} time step and $\hat{\boldsymbol{P}}_i$ denotes the prior state covariance matrix.

In the update step, the Kalman gain $\hat{\boldsymbol{K}}_i$ is calculated by

$$\hat{\boldsymbol{K}}_i = \hat{\boldsymbol{P}}_i\boldsymbol{H}^T\left(\boldsymbol{H}\hat{\boldsymbol{P}}_i\boldsymbol{H} + \boldsymbol{R}\right)^{-1}. \tag{5.5}$$

Based on $\hat{\boldsymbol{K}}_i$ and the observed measurement $\boldsymbol{f}_i$, the mean and covariance of the posterior state estimate, i.e., $\boldsymbol{s}_i$ and $\boldsymbol{P}_i$, can be updated as

$$\boldsymbol{s}_i = \hat{\boldsymbol{s}}_i + \boldsymbol{K}_i\left(\boldsymbol{f}_i - \boldsymbol{H}\hat{\boldsymbol{s}}_i\right), \tag{5.6}$$

$$\boldsymbol{P}_i = \left(\boldsymbol{I} - \hat{\boldsymbol{K}}_i\boldsymbol{H}\right)\hat{\boldsymbol{P}}_i, \tag{5.7}$$

where $\boldsymbol{I}$ is an identity matrix.

To apply Kalman filter, parameters $\boldsymbol{A}$, $\boldsymbol{H}$, $\boldsymbol{Q}$, and $\boldsymbol{R}$ are usually hand-designed and tuned under simplistic assumptions, which might be arbitrary and crude for many practical tasks [198]. Besides, the transition model and the observation model of Kalman filter tend to capture linear relationships between sEMG signals and hand kinematics, whereas a high non-linearity in data has been verified in numerous literature [45, 48, 152].

Different from Kalman filter, LSTM also exploits the recursive structure but captures the non-linear dynamics between the input and output with its weight matrices learned from data. The implementation of LSTM in sEMG-based kinematics estimation can refer to Eq. 4.1. However, LSTM mainly focuses on the relationship between hand kinematics and sEMG features, but can neither include

prior knowledge of the system nor capture the uncertainty information. In this study, the proposed LSTM-KF intends to solve these limitations by combining merits of Kalman filter and LSTM. Specifically, LSTM-KF follows the computational graph of Kalman filter described in Eq. 5.3-5.7, but $\boldsymbol{A}$, $\boldsymbol{H}$, $\boldsymbol{Q}$, and $\boldsymbol{R}$ are designed to be learnable on the basis of LSTM modules [198].

In the transition model, $\boldsymbol{A}_i$ and $\boldsymbol{Q}_i$ are produced by LSTM modules using previous state estimate $\boldsymbol{s}_{i-1}$:

$$\boldsymbol{A}_i, \boldsymbol{h}_i^A = LSTM_A\left(\boldsymbol{s}_{i-1}, \boldsymbol{h}_{i-1}^A\right), \tag{5.8}$$

$$\boldsymbol{Q}_i, \boldsymbol{h}_i^Q = LSTM_Q\left(\boldsymbol{s}_{i-1}, \boldsymbol{h}_{i-1}^Q\right), \tag{5.9}$$

where $LSTM_A$ and $LSTM_Q$ are LSTM modules for the approximation of $\boldsymbol{A}_i$ and $\boldsymbol{Q}_i$, whilst $\boldsymbol{h}_{i-1}^A$ and $\boldsymbol{h}_{i-1}^Q$ denote the hidden states of $LSTM_A$ and $LSTM_Q$, respectively.

In the observation model, CNN features $\boldsymbol{f}_i$ can be used as external measurement to produce $\boldsymbol{H}_i$ and $\boldsymbol{R}_i$:

$$\boldsymbol{H}_i, \boldsymbol{h}_i^H = LSTM_H\left(\boldsymbol{f}_i, \boldsymbol{h}_{i-1}^H\right), \tag{5.10}$$

$$\boldsymbol{R}_i, \boldsymbol{h}_i^R = LSTM_R\left(\boldsymbol{f}_i, \boldsymbol{h}_{i-1}^R\right). \tag{5.11}$$

The obtained $\boldsymbol{A}_i$, $\boldsymbol{R}_i$, $\boldsymbol{Q}_i$, and $\boldsymbol{R}_i$ are then fed into the prediction step and then the update step of Kalman filter to compute $\boldsymbol{s}_{i-1}$ and $\boldsymbol{P}_{i-1}$. The feed-forward architecture of LSTM-KF in one loop is depicted in Figure 5.2. As we can see, in LSTM-KF, both the transition/observation matrices and the corresponding covariance matrices can change through time dynamically. Moreover, as suggested by [199], $\hat{\boldsymbol{K}}_i$ in Eq. 5.5 is defined as the intermediate gain, and an extra LSTM module is exploited to learn the final gain $\boldsymbol{K}_i$ based on $\hat{\boldsymbol{K}}_i$:

$$\boldsymbol{K}_i, \boldsymbol{h}_i^K = LSTM_K\left(\hat{\boldsymbol{K}}_i, \boldsymbol{h}_{i-1}^K\right). \tag{5.12}$$

5.2.3 Model training

Following the training approaches in [198], CNN and LSTM-KF are also trained separately in this study. To train CNN, a regression layer is attached to the

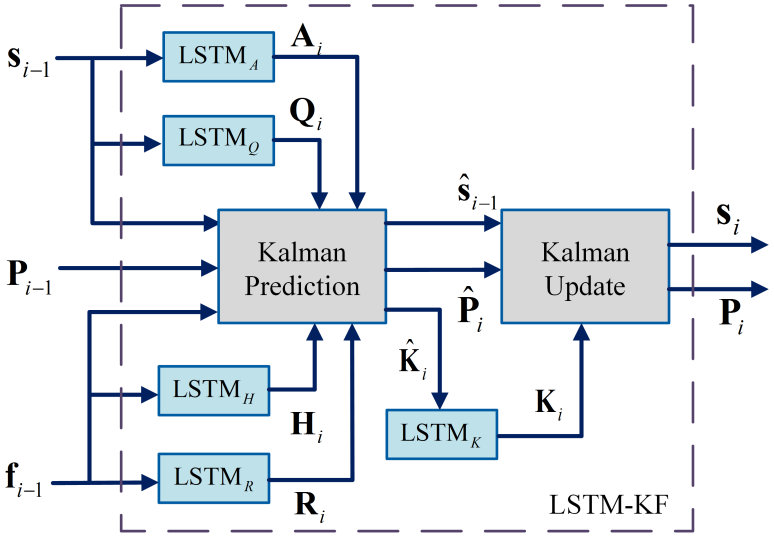


Figure 5.2: The feed-forward architecture of LSTM-KF in one loop of the recursive learning.

presented CNN architecture to complete a supervised learning process, in which inputs are sEMG matrices and labels are measured kinematics. After this step, feature vectors can be extracted from the second FC layer of CNN to construct training sequences LSTM-KF. From Eq. 5.3-5.12 we can see that feed-forward computation of LSTM-KF is conducted with iterative loops, and that all learnable weights come from LSTM modules. The standard Euclidean $L2$ loss is employed in the loss function of LSTM-KF, with a regularisation term to mitigate against over-fitting:

$$\mathcal{L}(\boldsymbol{W}) = \sum_{j=1}^{N}\sum_{i=1}^{k}\left\|\boldsymbol{s}_{ij} - \boldsymbol{y}_{ij}\right\|_2 + \lambda\|\boldsymbol{W}\|_2, \tag{5.13}$$

where $\boldsymbol{W}$ denotes the weights of LSTM modules, N is the number of training sequences in a training-batch, $\boldsymbol{s}_{ij}$ is the state of the i^{th} time step in the j^{th} sequence sample, $\boldsymbol{y}_{ij}$ is the measured kinematics, and λ is the scaling factor for regularisation.

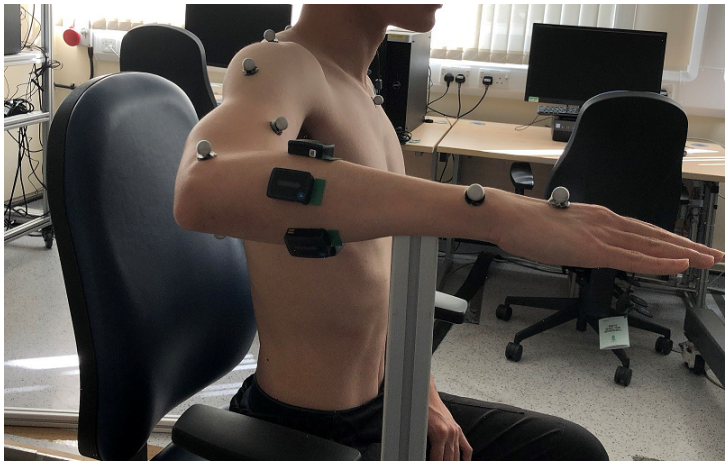


Figure 5.3: The placement of electrodes and markers in wrist kinematics estimation. The current gesture was regarded as the neutral position in the continuous wrist movement. More details can be found in [6].

5.3 Experimental methods

This study mainly focused on the joint kinematics estimation of wrist and fingers. Two data sources were employed to validate performances of LSTM-KF. The first data source was obtained from our experiments, in which wrist kinematics (flexion/extension) of eight healthy subjects were captured. The second source is a public benchmark which involves multi-DOF finger motions (index flexion, middle flexion, ring flexion) from ten able-bodied subjects and two amputees.

5.3.1 Wrist Kinematics Experiment

In this chapter, a wrist kinematics dataset constructed by our research group was utilised for model evaluation [6]. Approved by the MaPS and Engineering joint Faculty Research Ethics Committee of University of Leeds, UK (MEEC 18-002), six males and two females (aged 25 to 31) participated in this experiment. A written informed consent was obtained from each subject. As shown in Figure 5.3, subjects are asked to seat on the armchair, with torso fully straight and forearm relaxed. The current position of hand was set as the neutral position. During

data collection, participants were performed wrist flexion/extension following a continuous cycle trial: the wrist was rotated from neutral position to the flexion direction, it was then moved back to the extension direction and finally returned to neutral position. Five repetitive trials were conducted by each participate and each trial lasted for about 20s. A three minutes break was given between each trial to prevent muscle fatigue.

In our experiment, the Delsys TrignoTM system was used to record sEMG signals. Avanti electrodes were placed over five wrist muscles (Flexor Carpi Radialis, lexor Carpi Ulnaris, Extensor Carpi Radialis Longus, Extensor Carpi Radialis Brevis, and Extensor Carpi Ulnaris) over right forearm. The sampling rate of was set as 2000Hz. To capture wrist movements through the motion capture system (Vicon Motion Systems Ltd. UK), 16 reflective markers were placed on the right upper limb. As illustrated in Figure 5.3, markers were allocated over the spinous process of the 7th and 10th thoracic vertabra, right scapula, xiphoid, acromio-clavicular joint, clavicle, lateral/medial humerus medial epicondyle, right radial/ulnar styloid, middle forearm and the right third metacarpus. The sampling rate of Vicon Motion System was 250Hz and the synchronisation of the kinematic data and sEMG were conducted using a trigger module. The Vicon upper limb model were applied to calculate wrist joint angles as the measured angles or ground truth.

5.3.2 Finger Kinematics Experiment

In order to verify the generalisation of LSTM-KF, the finger kinematics estimation was conducted using data from the 8^{th} Ninapro database [104]. Ten able-bodied subjects (nine males, one female) and two male, right-hand transradial amputees were recruited. The Delsys TrignoTM system was utilised to collect sEMG data from the right forearm of participants at a rate of 1111Hz. The finger kinematic data were recorded with a dataglove (Cyberglove 2, 18-DOF model) worn on the left hand. During data acquisition, participants were asked to perform bilateral mirrored movements, in which nine hand motions were activated. Each motion execution lasted around 7s with a 3s interval. Three trials were recorded for each

participant. The first two trails comprised ten repetitions of each hand movement and the third trial comprised two repetitions. A ten minutes break was given to participants after each trial.

5.3.3 Data Pre-processing

In both two experiments, sEMG signals were processed using a 3rd order Butterworth high pass filter (20Hz) to remove movement artefacts [189] and a low pass filter (450Hz) to remove unusable high frequency noise [28]. A Min-Max scaling was applied to normalise sEMG in each channel [190]. These procedures are consistent with Chapter 4.3.2. To complete feature extraction, the size of sliding windows was set to be 100ms length with 20ms increment (80% overlap). Thus the size of sEMG matrix ($1 \times L \times C$) was $1 \times 199 \times 5$. Since subjects were asked to rotate the wrist in a comparatively low speed, the label of a sample was obtained by computing the mean value of angles (captured by Vicon) within the processing window. To implement sequence regression, the time duration of a training sequence was set as 300ms, resulting in 10 time steps in $[\boldsymbol{f}_1, \cdots, \boldsymbol{f}_k]$, i.e. $k = 10$. As for the labels in finger kinematics estimation, glove measurements were also averaged within a processing window and then converted into finger kinematics via a linear mapping [104]. The sliding windows were 200ms in length with 50ms increment, and the time duration of a training sequence was set as 500ms.

5.3.4 Hyper-parameter Settings

The training of CNN and LSTM modules is implemented using Tensorflow backend. Specifically, CNN was trained in a 32 sized mini-batch for 100 epochs via adaptive moment estimation. The dynamic learning rate was 0.0001 with a decay rate of 0.001 for each iteration. The slope scale of leaky ReLU layers was set as 0.1. The max-pooling layer used a pool size of 3, whilst the dropout rate is set to be 30%. As for LSTM modules, weights were initialised using Xavier initialisation and then trained in a 32 sized mini-batch for 100 epochs via ADAM. The initial

learning rate was set as 0.001 and retained 0.99 in each epoch. The regularisation factor λ was 0.01. In general, the number of hidden layers in LSTM will affect the regression performance. In our experiment, 20 hidden layers were empirically observed to be a good trade-off for two datasets since more layers would result in a much heavier computational load with limited improvement in model accuracy.

5.4 Results and Discussion

In this study, performances of DKFN were evaluated using inter-session results. In the wrist kinematics experiment, the model was trained in the first four trials and tested in the last trial. To validate the finger kinematics estimation, first two sessions were used for model training the last session for testing. Same to Chapter 4, R^2 also served as the performance metric to indicate how the estimated curve is related to measured joint motion. Since DKFN can be regarded as a variation of the work in last chapter, CNN and CNN-LSTM were selected as baseline methods for model comparison.

5.4.1 Wrist Kinematics Estimation

Figure 5.4 illustrates the estimation performances of CNN, CNN-LSTM, and DKFN in Subject 1-4. As we can see, wrist motions can be effectively reconstructed from sEMG signals using three methods. In each subject, the trajectory reconstructed by DKFN (red dashed line) is closer to the ground truth (blue solid line) than other two trajectories. Another interesting observation in Figure 5.4 is that trajectories of both CNN-LSTM and DKFN are comparatively smoother than the trajectory of CNN. A main reason is that the sequence regression conducted by LSTM and LSTM-KF can reduce some fluctuations by exploiting temporal dependencies among adjacent sEMG features. Different from conventional smoothing techniques such as the moving average, LSTM and LSTM-KF do not cause lags. In terms of the mappings of sEMG with wrist kinematics, LSTM and LSTM-KF are capable of capturing some of those buried in CNN features.

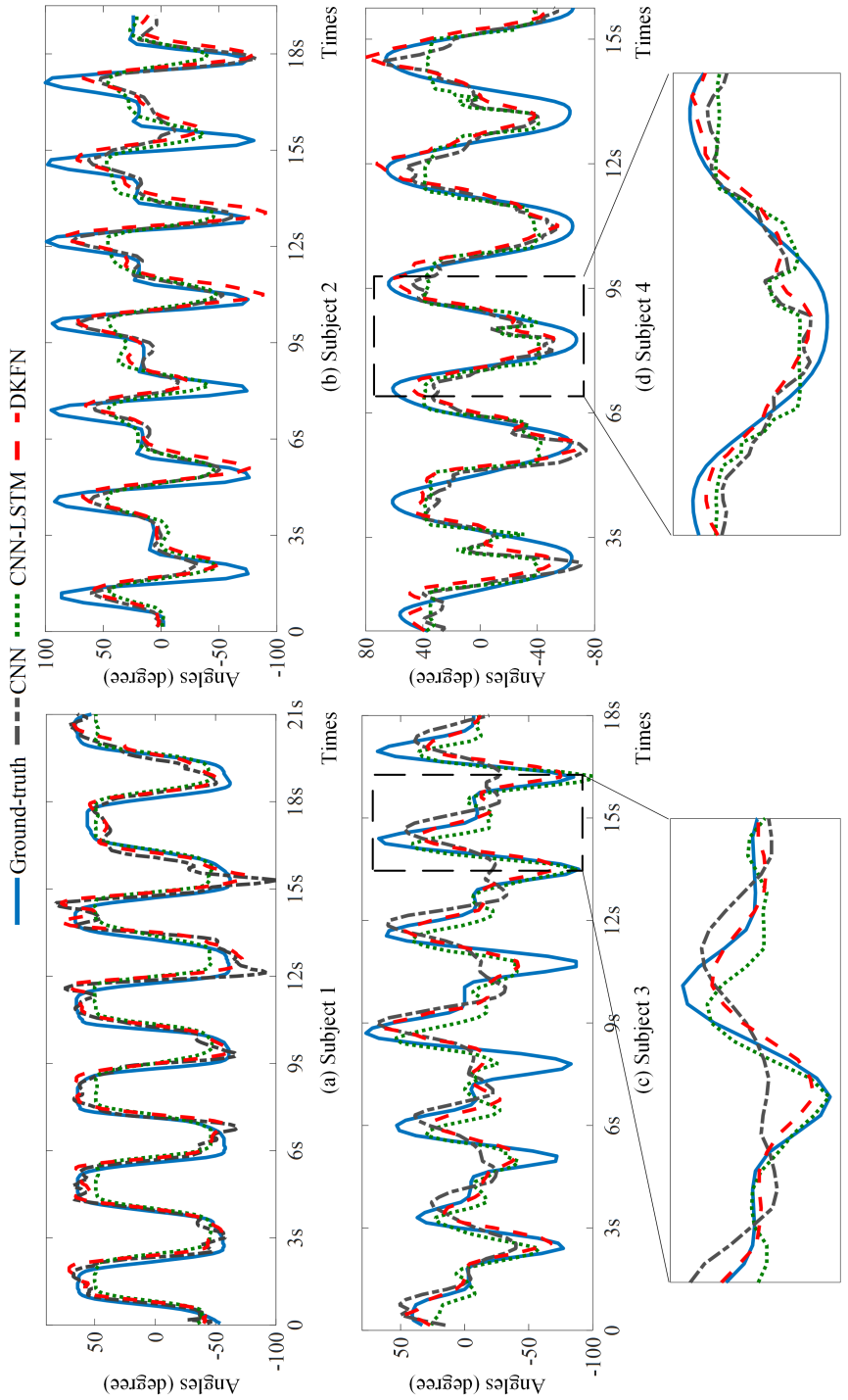


Figure 5.4: Performances of CNN, CNN-LSTM, and DKFN in wrist kinematics estimation.

Table 5.1: R^2 of CNN, CNN-LSTM, and DKFN in wrist kinematics estimation.

Participant	CNN	CNN-LSTM	DKFN
Subject1	0.87	0.89	***0.92***
Subject2	0.66	0.77	***0.83***
Subject3	0.47	0.57	***0.69***
Subject4	0.61	0.74	***0.86***
Subject5	0.84	0.87	***0.90***
Subject6	0.71	***0.95***	0.93
Subject7	0.82	0.86	***0.90***
Subject8	0.75	0.89	***0.94***
Average	0.72	0.82	***0.87***

As shown in Figure 5.4 (c), CNN-LSTM and DKFN managed to track target motions performed by Subject 3 between 13s-16s, whereas CNN only captured half of that trajectory.

To compare estimation performances quantitatively, Table 5.1 summarises the R^2 and RMSE of three methods in all subjects. The average performances (mean±standard deviation) of CNN, CNN-LSTM, and DKFN are 0.72±0.13, 0.82±0.12, 0.87±0.08 for R^2. As we can see, DKFN achieves much larger R^2 than other two methods. The outperformance of DKFN can be more evident in Subject 3-4. A special case in our experiment is Subject 6, where CNN-LSTM outperforms DKFN slightly. In fact R^2 values of 0.95 (CNN-LSTM) and 0.93 (DKFN) are actually quite close, and the predicted curves can both catch the ground-truth with very limited errors. It can be inferred that in this subject the extracted CNN features can be very well matched with the sequence regression conducted by vanilla LSTM or LSTM-KF.

5.4.2 Finger Kinematics Estimation

Different from the experiment settings of wrist kinematics estimation, finger motions were activated in multi-DOFs simultaneously. In this study, all three models

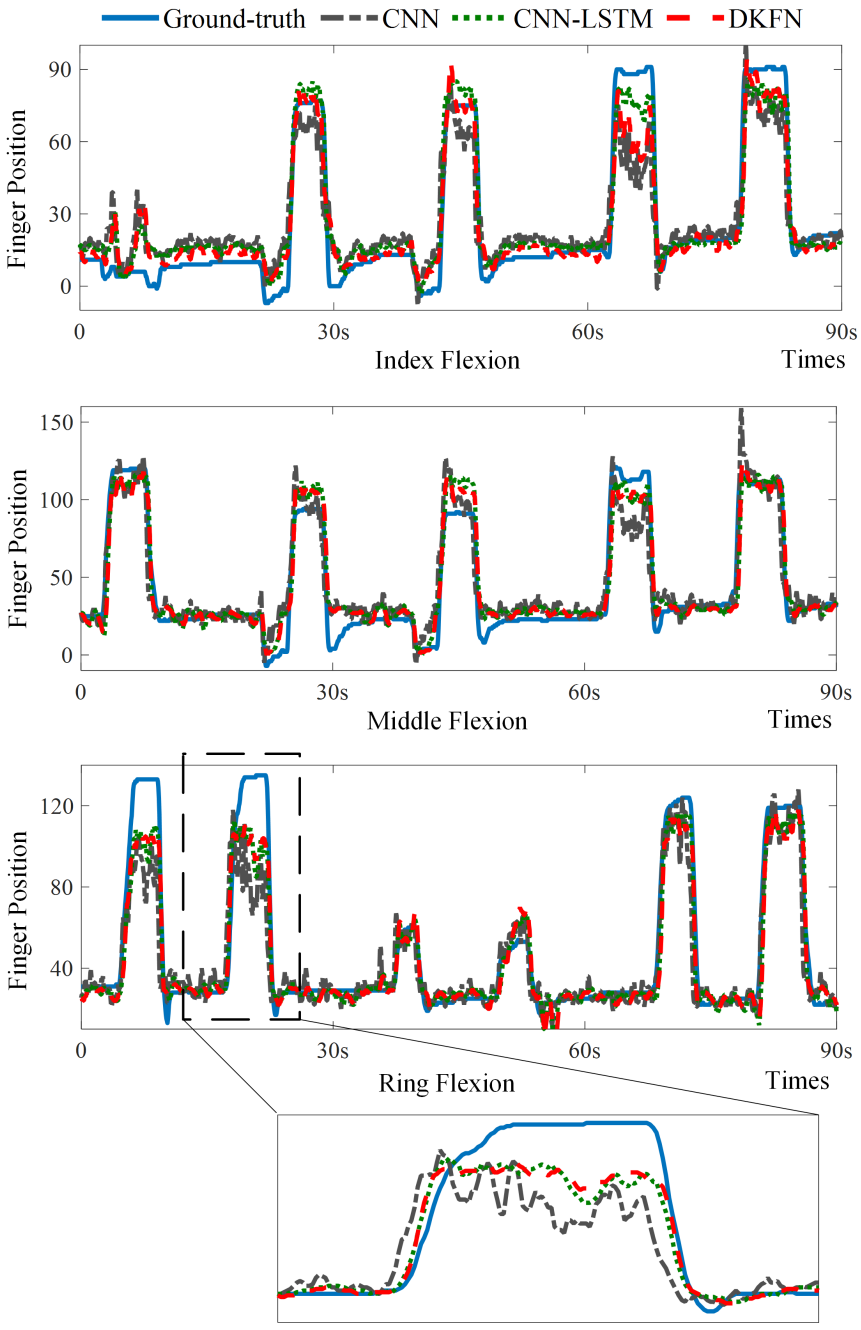


Figure 5.5: Performances of CNN, CNN-LSTM, and DKFN in finger kinematics estimation.

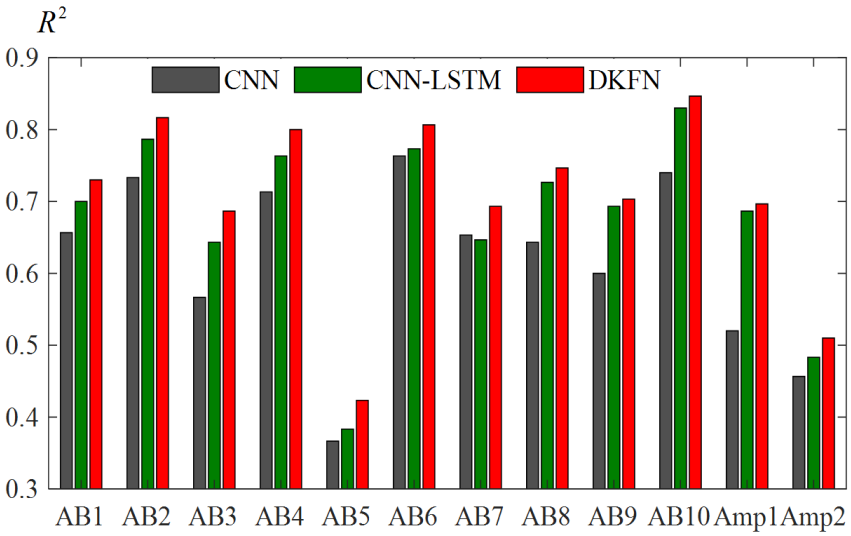


Figure 5.6: Average R^2 of CNN, CNN-LSTM, and DKFN among three DOFs in each subject. AB denotes the able-bodied subjects and Amp denotes amputees.

were trained/tested using kinematics data in each DOF (index/middle/ring flexion) separately. Figure 5.5 demonstrates the performances of CNN, CNN-LSTM and DKFN for estimating finger positions in an able-bodied subject (AB9). As we can see, three methods are capable of capturing measured trajectories in most cases. Similar to wrist kinematics estimations, herein LSTM and LSTM-KF can help to generate smoother trajectories and mitigate some abrupt fluctuations or deterioration of CNN, such as the last peak of curves in the middle flexion of AB9. Table 5.2 lists the R^2 of three methods in index/middle/ring flexion of each participant.

Figure 5.6 demonstrate the average estimation performances among all DOFs in each subject. The results of CNN, CNN-LSTM and DKFN among all DOFs in able-bodied subjects are 0.64±0.14, 0.69±0.15, 0.73±0.14, respectively. For finger kinematics of amputees, statistical results are 0.57±0.15, 0.66±0.17, 0.68±0.16, respectively. From this figure we can observe two interesting results: 1) compared with CNN, the sequence regression conducted by CNN-LSTM or DKFN improves estimation accuracy significantly; 2) through embedding LSTM modules in the

Table 5.2: R^2 of CNN, CNN-LSTM, and DKFN in Finger Kinematics Estimation.

Participant	Finger	CNN	CNN-LSTM	DKFN
Able-bodied 1	Index	0.53	0.58	***0.6***
	Middle	0.69	0.73	***0.76***
	Ring	0.75	0.79	***0.83***
Able-bodied 2	Index	0.67	0.69	***0.74***
	Middle	0.71	0.77	***0.8***
	Ring	0.82	0.9	***0.91***
Able-bodied 3	Index	0.57	0.64	***0.7***
	Middle	0.69	0.81	***0.84***
	Ring	0.44	0.48	***0.52***
Able-bodied 4	Index	0.63	0.71	***0.73***
	Middle	0.77	0.79	***0.83***
	Ring	0.74	0.79	***0.84***
Able-bodied 5	Index	0.29	0.38	***0.43***
	Middle	0.48	0.44	***0.46***
	Ring	0.33	0.33	***0.38***
Able-bodied 6	Index	0.75	0.75	***0.82***
	Middle	0.75	0.76	***0.78***
	Ring	0.79	0.81	***0.82***
Able-bodied 7	Index	***0.71***	0.67	0.7
	Middle	0.55	0.61	***0.63***
	Ring	0.7	0.66	***0.75***
Able-bodied 8	Index	0.53	0.53	***0.59***
	Middle	0.66	0.77	***0.79***
	Ring	0.74	***0.88***	0.86
Able-bodied 9	Index	0.47	0.61	***0.62***
	Middle	0.59	0.64	***0.68***
	Ring	0.74	***0.83***	0.81
Able-bodied 10	Index	0.59	0.67	***0.68***
	Middle	0.76	0.89	***0.91***
	Ring	0.87	0.93	***0.95***
Amputee 1	Index	0.52	***0.71***	0.69
	Middle	0.48	0.67	***0.69***
	Ring	0.56	0.68	***0.71***
Amputee 2	Index	0.42	***0.55***	0.5
	Middle	0.52	0.49	***0.57***
	Ring	0.43	0.41	***0.46***

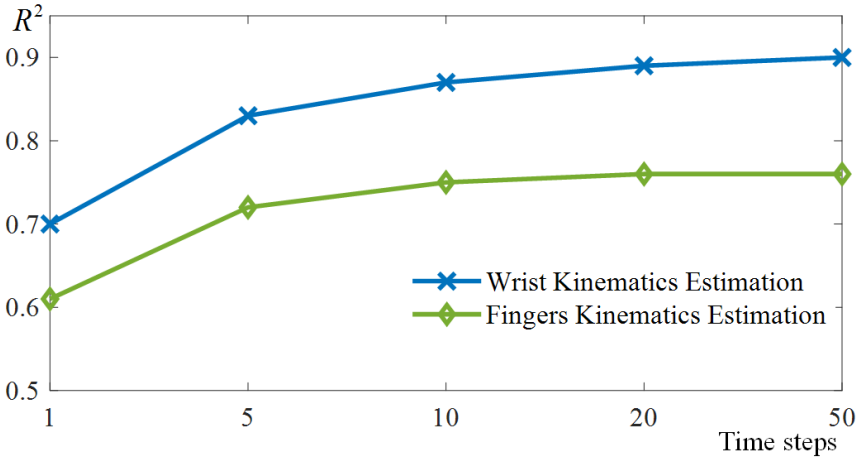


Figure 5.7: Average estimation results of DKFN among all subjects with varied length of sequences.

Kalman filter computational graph, LSTM-KF further enhances the accuracy of sequence regression. Besides, due to the complexity of finger motions, average performances of three models are inferior to those in wrist kinematics estimation. However, these models can still achieve high accuracy in many subjects, such as AB2, AB4, and AB6 (in ring flexion of AB2, R^2 of CNN, CNN-LSTM, and DKFN can reach 0.82, 0.90, and 0.91, respectively).

5.4.3 Effects of Sequence Length

The length of training/testing sequences determines the number of deep feature vectors involved in the sequence regression. In general, a too short sequence cannot provide sufficient information for model learning, whereas a too long sequence will significantly reduce the computational efficiency. Thus, a trade-off is of importance to enhance the practical utilisation of DKFN in myoelectric control. Herein, the influence of sequence length in wrist/finger kinematics estimation was validated. Five different time-steps were selected for comparison, i.e. 1, 5, 10, 20, 50. These time-steps correspond to the sequences lasting 120ms, 200ms, 300ms, 500ms, 1100ms in the wrist dataset and 250ms, 450ms, 700ms, 1200ms

2700ms in the finger dataset. Figure 5.7 illustrated the average estimation results of DKFN among all subjects using varied length of sequences. As we can see, in both datasets the model performances are overall improved with the increase of sequence length, but improvements will be limited when the length becomes comparatively larger. Therefore, it can be inferred that a sequence lasting 300-500ms may work properly in wrist kinematics estimation, whilst a length of 500-800ms might be suitable for the finger dataset.

5.5 Conclusions

This chapter proposed DKFN which utilises a novel structure, i.e. LSTM-KF, to conduct sequence regression for hand kinematics estimation in HMI. With LSTM modules integrated into the computational graph of Kalman filter, parameters of the transition/observation model and Kalman gain can be learned from data effectively. Since LSTM-KF combines the non-linear transform property of LSTM with the probabilistic fusion mechanism of Kalman filter, it outperforms conventional LSTM in the sequence regression of CNN features extracted from sEMG in wrist/finger kinematics estimation.

Chapter 6

Inter-Subject Domain Adaptation for CNN-Based Kinematics Estimation

Following most experimental protocols in related research, the above Chapters mainly focus on performances of DL in intra-subject scenario. However, as a type of bio-electricity, sEMG is inherently user-specific, causing the amplitude and frequencies to be highly variable among individuals even when signals are measured from the same location with the same motion. In this context, a pre-trained DL model often suffers from degradation when it is testing on a new subject. This issue can be regarded as a domain shift problem in DL applications, where the distributions of training and testing data differ significantly. Although retraining a new model from scratch can help to avoid this issue, considerable number of labelled data that is specific to the target subject need to be collected to enable sufficient learning, which is actually inconvenient in practice.

In this chapter, a novel regression scheme is proposed for supervised domain adaptation (SDA), based on which domain shift effects can be effectively reduced. Specifically, a two-stream CNN with shared weights is established to exploit source and target sEMG data simultaneously, such that domain-invariant

features can be extracted. To tune CNN weights, both regression losses and a domain discrepancy loss are employed, where the former enable supervised learning and the latter minimises distribution divergences between two domains. In this study, eight healthy subjects were recruited to perform wrist flexion-extension movements. Experiment results illustrated that the proposed regression SDA outperformed fine-tuning, a state-of-the-art transfer learning method, in both single-single and multiple-single scenarios of kinematics estimation. Unlike fine-tuning which suffers from catastrophic forgetting, regression SDA can maintain much better performances in original domains, which boosts the model reusability among multiple subjects.

6.1 Introduction

6.1.1 Motivation

Recently, DL techniques, especially CNN, has gained considerable attentions in joint kinematics estimation to shift the paradigm of artificial intelligence (AI) from conventional feature engineering to feature learning. For example, Ameri *et al.* proposed a CNN-based regression model which estimated wrist angles more accurately than support vector regression (SVR) and achieved better performances in the Fitts' law test [110]. Yang *et al.* investigated data-augmentation methods for CNN, and observed that CNN outperformed SVR significantly in the decoding of wrist kinetics [152, 175]. Moreover, CNN can also work as the deep feature extractor in the hybrid CNN-RNN (RNN denotes recurrent neural networks) scheme to further increase the estimation accuracy [14, 177]. However, these results are mainly obtained in laboratory settings which are simplistic and static. In fact, characteristics of sEMG can be easily influenced by external factors including muscle fatigue, electrode shift, impedance changes in electrodeâ€“skin interface, variations of contraction forces, and arm position effects, etc. [22, 56, 58, 201]. In particular, sEMG signals have a user-specific nature, causing a pre-trained CNN model to suffer from severe degradation when testing on a new individual.

Although it has been reported that features learned by deep neural networks may be able to share similar distributions across different subjects [152, 202], the inter-subject problem can still lead to a sharp decline in the estimation performances [152, 203].

Traditional DL approaches assume that training and testing data stem from the same underlying distribution. However, this assumption barely holds in practice [204], where the source domain D^S and target domain D^T have different feature spaces or marginal probability distributions [121], i.e. $D^T \neq D^S$. This issue is also known as domain shift. To this end, transfer learning (TL) has been investigated by exploiting knowledge learned in D^S and to effectively train DL models in D^T with insufficient labelled data. A simple but prevalent deep TL approach is fine-tuning (FT), where weights of a DL model developed in D^S are used as the starting points for the model to be trained in D^T. FT has also been reported to enhance model training or adaptability in sEMG-based hand motion estimation. For instance, Wang *et al.* utilised FT in the training of a multimodal recurrent CNN. In this study, D^S data came from the NinaPro project [139], and D^T was composed of multimodal data collected from experiments. Ameri *et al.* employed FT to enhance CNN performances under the condition of electrode shift [141]. Experiments in both hand gesture recognition and wrist kinematics estimation verified the outperformance of FT when compared with a simple aggregation of pre-shift and post-shift sets. In addition, Kim *et al.* also fine-tuned the supportive CNN classifiers in the proposed subject-transfer framework, such that the estimation model can be more robust in terms of intra-user variability [140]. However, FT is prone to be overfitting when too few labelled data are available in the target domain [205]. Besides, a fine-tuned network usually suffers from catastrophic forgetting which destroys the model reusability [206].

Apart from FT, another popular TL scheme is domain adaptation (DA) which improves the target predictive function $f(\bullet)$ by exploiting the knowledge in D^S and D^T simultaneously. Compared with FT, DA not only reduces the demand for labelled target data but also enables consistent performances on different domains [204]. The main idea of DA is to align feature distributions of D^S and

D^T in an embedding space. In practice, many efforts have followed the Siamese architecture [207], i.e. a two-stream CNN with shared weights. In this structure, one stream represents the source model and the other represents the target model. By adding additional discrepancy losses such as maximum mean discrepancy (MMD) [208–211], correlation alignment (CORAL) [212], or higher-order moments [213] etc. in model training, distribution divergences can be effectively minimised. Representative works include deep domain confusion (DDC) [208] and deep adaptation networks (DAN) [209]. In specific, DDC exploited a two-stream CNN and minimised MMD between outputs of the last fully-connected (FC) layer in each stream. DAN expanded DDC by employing multiple MMD terms to process outputs of several FC layers. Further advancements can be found in residual transfer network (RTN) [210] and joint adaptation networks (JAN) [211], etc. However, these approaches were mainly proposed to enhance CNN classification in computer vision (CV) tasks. To our best knowledge, few investigations have been conducted to address domain shift in sEMG-based kinematics estimation.

6.1.2 Contribution

A novel regression scheme for SDA is proposed to reduce domain shift effects on CNN-based wrist kinematics estimation in the inter-subject circumstance. In this study, the source domain D^S denotes the source subjects which provide sufficient labelled data for CNN training, and target domain D^T represents the target subject to be tested using the pre-trained model. Specifically, a Siamese architecture is established to exploit both source and target data simultaneously, such that the domain-invariant features can be extracted. To tune CNN weights effectively, three types of loss functions are employed, including the regression losses for supervised learning in D^S and D^T, a MMD loss to reduce distribution mismatches between two domains in the latent space, and a regression contrastive loss to learn more discriminative deep features for domain alignment.

6.1.3 Chapter Organisation

The remainder of this chapter is structured as follows. Section 6.2 describes the regression SDA, where the model structure, the CNN layers and the loss functions are presented elaborately. Section 6.3 introduces experimental setups for wrist kinematics estimation. Section 6.4 presents experimental results of regression SDA and baseline methods. The chapter is discussed in section 6.5 and then concluded in section 6.6.

6.2 Methodology

In general, DA can be divided into supervised (SDA) and unsupervised (UDA) approaches [122]. In UDA, there are no labelled data in D^T, and existed works mainly focus on the alignment of feature distributions between domains. As for SDA, a small number of labelled samples in D^T can be utilised to build a bridge from sources to targets. Herein, SDA is preferred since it can be more accurate in terms of the adaptation to large changes in sEMG signals [141]. The superiority in estimation performances have also been reported in CV tasks [214]. Specifically, in SDA we are given $D^S = \left\{\left(\boldsymbol{X}_m^S, \boldsymbol{y}_m^S\right)\right\}_{m=1}^{M}$, $D_{\text{train}}^T = \left\{\left(\boldsymbol{X}_n^T, \boldsymbol{y}_n^T\right)\right\}_{n=1}^{N}$ and $D_{\text{test}}^T = \left\{\left(\boldsymbol{X}_j^T, \boldsymbol{y}_j^T\right)\right\}_{j=1}^{J}$, where $\boldsymbol{X}$ denotes sEMG matrix extracted from raw signals and $\boldsymbol{y}$ is the related wrist angle (ground-truth). D_{train}^T participates in the model training together with D^S, whilst D_{test}^T is utilised to test SDA performances. It is noted that data in D_{train}^T are normally insufficient to train a conventional CNN, i.e. $N \ll J \approx M$.

6.2.1 Framework of Regression SDA

Following most efforts in DA [204, 208–211, 214], our regression SDA is also designed based on the two-stream CNN structure with shared weights. The basic structure of CNN is consistent with those in Chapter 4 and 5. As depicted in Figure 6.1, a pairwise sample $\left\{\left(\boldsymbol{X}_m^S, \boldsymbol{y}_m^S\right), \left(\boldsymbol{X}_n^T, \boldsymbol{y}_n^T\right)\right\}$ is imported into regression SDA, in which the first stream operates $\boldsymbol{X}_m^{\mathrm{S}}$ and the second operates $\boldsymbol{X}_n^{\mathrm{T}}$ separ-

ately. The construction of pairwise samples allows each target sample $\left(\boldsymbol{X}_n^T, \boldsymbol{y}_n^T\right)$ to be paired with all source samples $\left(\boldsymbol{X}_m^S, \boldsymbol{y}_m^S\right)$, which is able to effectively align the entire source data with the few target data [214]. This process can also be regarded as the Cartesian product of two datasets [215].

It is noted that the total number of pairwise samples, i.e. $M \times N$, will not be overlarge due to the size of D_{train}^T. In practice, the computational load can be further reduced by downsampling these pairwise samples for model training [214, 215]. Apart from the model structure, loss functions are also of vital importance to regression SDA. In this framework, regression losses and a domain discrepancy loss are combined to tune CNN, where the former is leveraged for supervised learning and the latter works to align feature distributions of source and target streams.

6.2.2 Design of Loss Functions

To tune CNN weights $\boldsymbol{W}$, the source label $\boldsymbol{y}_m^S$ and target label $\boldsymbol{y}_n^T$ are utilised to calculate the regression loss, i.e. mean square error (MSE), for each stream. Meanwhile, a domain discrepancy loss is also added to minimise the distribution divergence between two domains. Therefore, the optimal weights $\boldsymbol{W}^*$can be learned by reducing the total loss which is formulated as

$$\mathcal{L}\left(\boldsymbol{W} \mid \boldsymbol{X}^S, \boldsymbol{y}^S, \boldsymbol{X}^T, \boldsymbol{y}^T\right) = \mathcal{L}_S + \mathcal{L}_T + \mathcal{L}_d, \tag{6.1}$$

$$\mathcal{L}_S = MSE\left(\boldsymbol{W} \mid \boldsymbol{X}^S, \boldsymbol{y}^S\right), \tag{6.2}$$

$$\mathcal{L}_T = MSE\left(\boldsymbol{W} \mid \boldsymbol{X}^T, \boldsymbol{y}^T\right), \tag{6.3}$$

$$\begin{aligned} \mathcal{L}_d = \gamma_1 MMD^2\left(\boldsymbol{W} \mid \boldsymbol{X}^S, \boldsymbol{X}^T\right) + \\ \gamma_2 RContrastive\left(\boldsymbol{W} \mid \boldsymbol{X}^S, \boldsymbol{y}^S, \boldsymbol{X}^T, \boldsymbol{y}^T\right), \end{aligned} \tag{6.4}$$

where $\mathcal{L}_S$ denotes the regression loss calculated in D^S whilst $\mathcal{L}_T$ represents the loss in D_{train}^T. $\mathcal{L}_d$ is the domain discrepancy loss combined of a MMD loss and a regression contrastive loss ($RContrastive$). In particular, $RContrastive$ is an expansion of the classification contrastive loss ($CContrastive$) originally designed

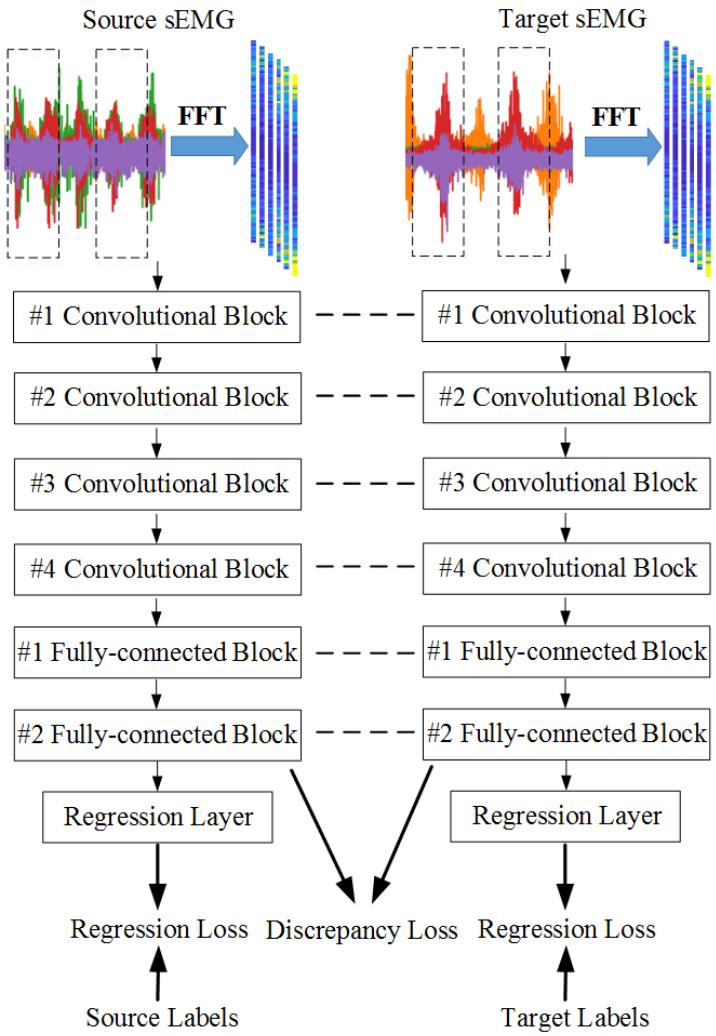


Figure 6.1: Framework of regression SDA for kinematics estimation.

to guarantee deep features with better intra-class compactness and inter-class separability in the latent space [214, 216]. Coefficients γ_1 and γ_2 are used to balance MMD loss and *RContrastive* in model training.

MSE Loss

MSE loss is one of the most commonly used regression loss functions for supervised learning. It is the sum of squared distances between ground-truth and predictions:

$$MSE\left(\boldsymbol{W} \mid \boldsymbol{X}^S, \boldsymbol{y}^S\right) = \frac{\sum_1^M \left\|\boldsymbol{y}_m^S - \widehat{\boldsymbol{y}}_m^S\right\|_2^2}{M}, \tag{6.5}$$

$$MSE\left(\boldsymbol{W} \mid \boldsymbol{X}^T, \boldsymbol{y}^T\right) = \frac{\sum_1^N \left\|\boldsymbol{y}_n^T - \widehat{\boldsymbol{y}}_n^T\right\|_2^2}{N}, \tag{6.6}$$

where $\widehat{\boldsymbol{y}}_m^S$ and $\widehat{\boldsymbol{y}}_n^T$ denote the predicted wrist angles in the source and target stream, respectively.

MMD Loss

Given two sets of data drawn from two distributions, MMD measures the distance between the mean of these two sets after mapping each sample to a Reproducing Kernel Hilbert Space (RKHS) [217]. The empirical estimate of squared MMD is as follows

$$MMD^2\left(\boldsymbol{X}^S, \boldsymbol{X}^T\right) = \left\| \sum_{m=1}^{M} \frac{\varphi\left(\boldsymbol{f}_m^S\right)}{M} - \sum_{n=1}^{N} \frac{\varphi\left(\boldsymbol{f}_n^T\right)}{N} \right\|_{\mathcal{H}}^2, \tag{6.7}$$

where $\boldsymbol{f}_m^S$ and $\boldsymbol{f}_n^T$ represent the feature vectors extracted in the 2$^{\text{nd}}$ FC Block of the source and target streams, respectively. $\varphi(\bullet)$ indicates the mapping of the feature vectors to RKHS, and $\| \bullet \|_{\mathcal{H}}$ denotes the RKHS norm. In practice, Eq. 6.7 is usually calculated using kernel tricks, and the MMD loss can be further expressed as

$$\begin{aligned} MMD^2\left(\boldsymbol{W} \mid \boldsymbol{X}^S, \boldsymbol{X}^T\right) = & \frac{\sum_{m,m*}^{M} k\left(\boldsymbol{f}_m^S, \boldsymbol{f}_{m^*}^S\right)}{M^2} \\ & - \frac{\sum_{m,n}^{M,N} k\left(\boldsymbol{f}_m^S, \boldsymbol{f}_n^T\right)}{M \times N} + \frac{\sum_{n,n*}^{N} k\left(\boldsymbol{f}_n^T, \boldsymbol{f}_{n^*}^T\right)}{N^2}, \end{aligned} \tag{6.8}$$

where $k(\bullet, \bullet)$ is a kernel function. Following most studies in DA, the standard RBF kernel is adopted such that MMD can compare all the orders of statistic moments [122]. As suggested in [205], the variance in RBF kernel is empirically set as 1.

Regression Contrastive Loss

RContrastive is to learn more discriminative deep features in regression tasks. The basic idea is that samples from different domains but with similar kinematics should be mapped nearby in the latent space. On the contrary, dissimilar samples should be distant from each other. Therefore, *RContrastive* is formulated as

$$\begin{aligned} RContrastive\left(\boldsymbol{W} \mid \boldsymbol{X}^S, \boldsymbol{y}^S, \boldsymbol{X}^T, \boldsymbol{y}^T\right) &= Y\left\|\boldsymbol{f}_m^S - \boldsymbol{f}_n^T\right\|_F^2 \\ +(1-Y)\left\{\max\left(0, \sigma - \left\|\boldsymbol{f}_m^S - \boldsymbol{f}_n^T\right\|_F\right)\right\}^2, \end{aligned} \tag{6.9}$$

where $\| \bullet \|_F$ denotes the Frobenius norm, and σ is a margin to specify the separation of feature vectors in the embedding space. Y is the label defined for the similarity of a pairwise sample. As mentioned before, *RContrastive* is the expansion of *CContrastive* which is designed for the classification tasks [214, 216]. In *CContrastive*, Y can be denoted as a binary value determined by the rule: $Y = 1$ if the source and target data are from the same category; otherwise $Y = 0$. However, in regression tasks $\boldsymbol{y}_m^S$ and $\boldsymbol{y}_n^T$ are continuous values that cannot be assigned to specific categories. To address this issue, the computation of Y in *RContrastive* is modified as

$$\mathrm{Y} = 1 - \frac{\left\|\boldsymbol{y}_m^S - \boldsymbol{y}_n^T\right\|_1}{\alpha}, \tag{6.10}$$

where α denotes a constant which normalises $\frac{\left\|\boldsymbol{y}_m^S - \boldsymbol{y}_n^T\right\|_1}{\alpha}$ into [0,1]. From Eq. 6.8 we can see that Y=1 if $\boldsymbol{y}_m^S = \boldsymbol{y}_n^T$. By contrast, Y will become smaller or even close to zero when $\boldsymbol{y}_m^S$ and $\boldsymbol{y}_n^T$ are dissimilar substantially.

To summarise, in the proposed method, both regression losses and discrepancy losses are employed to tune CNN weights $\boldsymbol{W}$: $\mathcal{L}_S$ and $\mathcal{L}_T$ utilise MSE losses to enable supervised learning in each domain, whilst $\mathcal{L}_d$ works as a regularisation term to ensure that CNN can perform well in both domains. Specifically, $\mathcal{L}_d$ is combined of MMD loss and *RContrastive*, where MMD loss minimises the distribution mismatch of two different domains in the latent space, and *RContrastive* provides more discriminative deep features to further boost domain alignment. With these losses, $\boldsymbol{W}$ can be updated effectively using the backpropagation al-

gorithm. The final weights $\boldsymbol{W}^*$ can be leveraged to estimate wrist kinematics in D^T_{test}. The overall process of SDA further is summarised in Algorithm 1.

Algorithm 1: The Proposed Regression SDA.

Input: Source domain dataset $D^S = \left\{\left(\boldsymbol{X}^S_m, \boldsymbol{y}^S_m\right)\right\}^M_{m=1}$, target domain dataset $D^T_{\text{train}} = \left\{\left(\boldsymbol{X}^T_n, \boldsymbol{y}^T_n\right)\right\}^N_{n=1}$, learning rate β, max training epochs T, loss coefficients γ_1 and γ_2, parameters σ and α.

Output: Optimal weights $\boldsymbol{W}^*$

1: Construct pairwise samples $\left\{\left(\boldsymbol{X}^S_m, \boldsymbol{y}^S_m\right), \left(\boldsymbol{X}^T_n, \boldsymbol{y}^T_n\right)\right\}$ based on D^S and D^T_{train}.

2: Initialise $\boldsymbol{W}$.

3: while epoch $t < T$ do

4: $\quad \widehat{\boldsymbol{y}}^S_m, \boldsymbol{f}^S_m \leftarrow CNN\left(\boldsymbol{X}^S_m, \boldsymbol{W}(t)\right)$

5: $\quad \widehat{\boldsymbol{y}}^T_n, \boldsymbol{f}^T_n \leftarrow CNN\left(\boldsymbol{X}^S_n, \boldsymbol{W}(t)\right)$

6: $\quad$ Calculate $\mathcal{L}_S$, $\mathcal{L}_T$, and $\mathcal{L}_d$ based on Eq. 6.5-6.10

7: $\quad \boldsymbol{W}(t+1) \leftarrow \boldsymbol{W}(t) - \beta \nabla_{\boldsymbol{W}} \left(\mathcal{L}_S + \mathcal{L}_T + \mathcal{L}_d\right)$

8: end while

9: Return $\boldsymbol{W}^*$

6.2.3 Baseline Methods

To demonstrate the effectiveness of regression SDA, it is further compared with several baseline methods. The descriptions of these methods are as follows.

Source Only (SO)

SO simulates the implementation of a pre-trained CNN in the TL process $D^S \rightarrow D^T$, where only data in D^S are utilised for supervised learning.

Target Only (TO)

TO represents the conventional training of CNN using D^T_{train}, in which the network weights are randomly initialised. Similar to regression SDA, SO, and TO are also trained using pairwise samples reconstructed from D^S and D^T_{train}. However, only $\mathcal{L}_S$ is adopted in SO, whereas TO utilises $\mathcal{L}_T$ instead. This strategy can also be regarded as a data augmentation approach for deep learning [218].

Joint Training (JT)

JT shares the same architecture with SDA but the discrepancy loss is excluded. It can be considered as a TL/DA approach which attempts to exploit information in both D^S and D^T.

Fine-tuning (FT)

As aforementioned, FT is the simplest but most prevalent TL approach in deep learning applications. Following previous research [141], convolutional layers are transferred from a CNN that is pre-trained in D^S as the initial values for a new model to be trained in D^T_{train}.

Ordinary least square (OLS)

Since the least square based approaches do not heavily rely on the size of training data and computation resources, the OLS model is also included for comparison. Similar to TO, OLS is also trained using D^T_{train}. As suggested by previous studies [26, 99, 193], several temporalâ€“spatial features are extracted from sEMG, including MAV, RMS, VAR, and 4^{th} AR. In case of over-fitting, PCA is applied to reduce redundant hand-crafted features.

6.3 Materials and Experimental Methods

In this study, SDA was evaluated using the customized dataset of wrist kinematics estimation. Details of experiment settings and data pre-processing can be found in Chapter 5.3.1. To extract samples for CNN, the size of sliding windows was set to be 100ms length with 50ms increment. To eliminate the impact of amplitude differences among subjects, sEMG data of each participant were normalised by dividing the peak value of each given muscle in the isometric maximum voluntary contraction [219].

6.3.1 Hyper-parameter Setting

The two-stream network was trained in a 32 sized mini-batch for 100 epochs via adaptive moment estimation (ADAM). The dynamic learning rate was 0.001. The slope scale of leaky ReLU layers was set as 0.1. The max-pooling layer used a pool size of 3, whilst the dropout rate was set to be 30%. Following [214], we also set $\sigma = 1$ for *RContrastive*. Based on experiment protocols, α in Eq. 6.10 was set to be 180 since wrist rotations were normally within $[-90°, 90°]$. In addition, we empirically set $\gamma_1 = 1000$ and $\gamma_2 = 0.1$. The training of the network was implemented using Pytorch backend.

6.3.2 Model Evaluation

Consistent with Chapter 4.5, R^2 was used to evaluate the performances of regression SDA. The mathematical expression can be found in Eq. 4.2. It is noted that R^2 of a perfect estimation is close to 1, and it becomes negative if the square sum of estimation errors are larger than the variance of ground-truth.

In this study, SDA was compared with baseline methods in both single-single and multiple-single scenarios of kinematics estimation. The dataset of each subject was categorised as either the source or target domain for each TL process, which thus resulted in 56 processes in the single-single scenario. For the sake of simplicity, we use $D^{Sa} \rightarrow D^{Tb}$ $(a, b = 1, 2, ..., 8, a \neq b)$ to define the TL process

from source subject a to target subject b. To guarantee a sufficient training, data in five experiment trials of a source subject were combined to construct a comparatively large D^S. Differently, in the multiple-single scenario, the inter-user data are leveraged for model training before testing on a new participant. Assuming that more general and informative features could be learned by CNN based on data aggregated from multiple individuals, this scenario is also prevalent in the inter-subject evaluations of TL approaches. In our experiment, for each D^{Tb}, the corresponding D^S was composed of data from the rest seven subjects.

According to the settings of regression SDA [122, 214], i.e. sufficient labelled training data in the source domain and sparse ones in target, D^T_{train} was composed of only 10% data collected in one experimental trial (about $2 \sim 3s$ to cover a wrist contraction circle from extension to flexion), whilst the rest data of this trial are stored in D^T_{test} for evaluation. Besides, to reduce the computational load, only the first 20% of the shuffled pairwise samples were utilised for model training. This procedure is similar to the ratio filter applied in [214, 215]. Empirically, the estimation accuracy is observed to be close to those when all $M \times N$ pairwise samples were involved.

6.4 Results

6.4.1 Domain Shift Effects on Inter-subject Estimation

Figure 6.2 demonstrates sEMG signals and related wrist angles of two subjects in kinematics estimation. Amplitudes of sEMG in each channel indicate the activation levels of the measured muscle. In particular, the measured muscles include Flexor Carpi Radialis (FCR), Flexor Carpi Ulnaris (FCU), Extensor Carpi Radialis Longus (ECRL), Extensor Carpi Radialis Brevis (ECRB), and Extensor Carpi Ulnaris (ECU). As we can see, in some channels (such as CH5) sEMG patterns can differ substantially among subjects even though wrist motions are similar. Performances of a CNN model were then validated in both intra-subject (D^S and D^T are from one subject) and inter-subject (D^S and D^T are from two

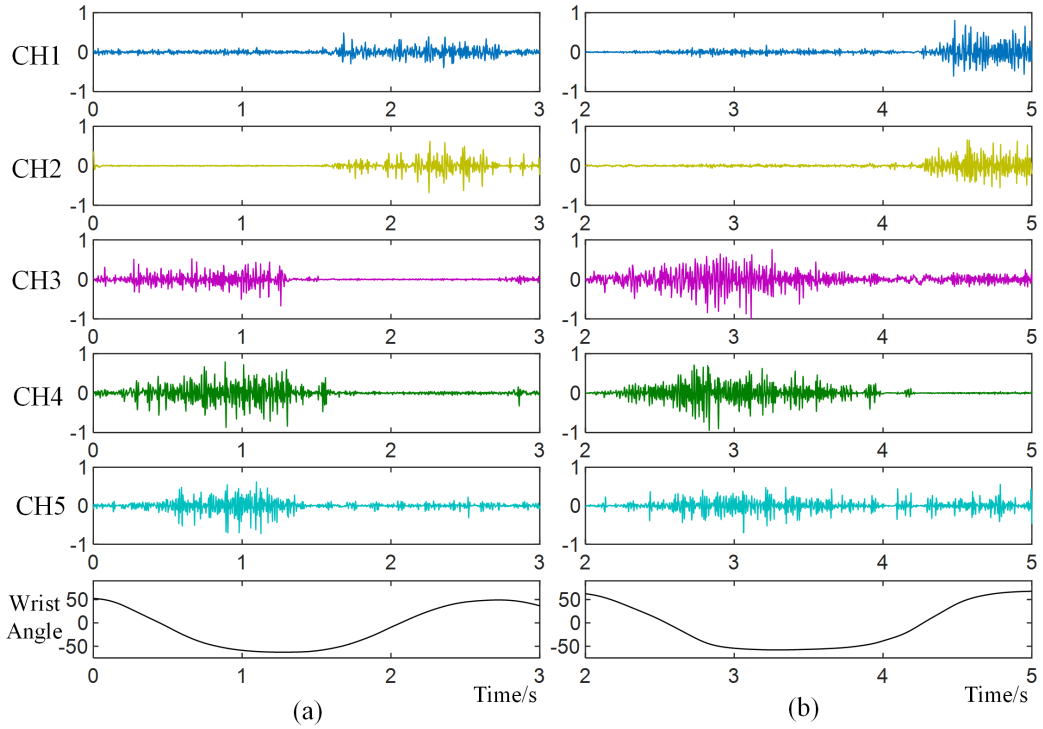


Figure 6.2: Normalised sEMG signals and wrist angles of (a) subject 6 and (b) subject 1 in a rotation cycle. The wrist angles are measured in degrees. The channel numbers and measured muscles were consistent among all subjects: CH1-FCR, CH2-FCU, CH3-ECRL, CH4-ECRB, CH5-ECU. As shown in this figure, muscle activation varied dramatically among two subjects. In particular, ECU of subject 6 was mainly activated during wrist extension. By contrast, high activation can be found in ECU of subject 1 during flexion.

different subjects) circumstances. As shown in Figure 6.3, the validation loss decreases effectively in the former circumstance but can hardly converge in the latter.

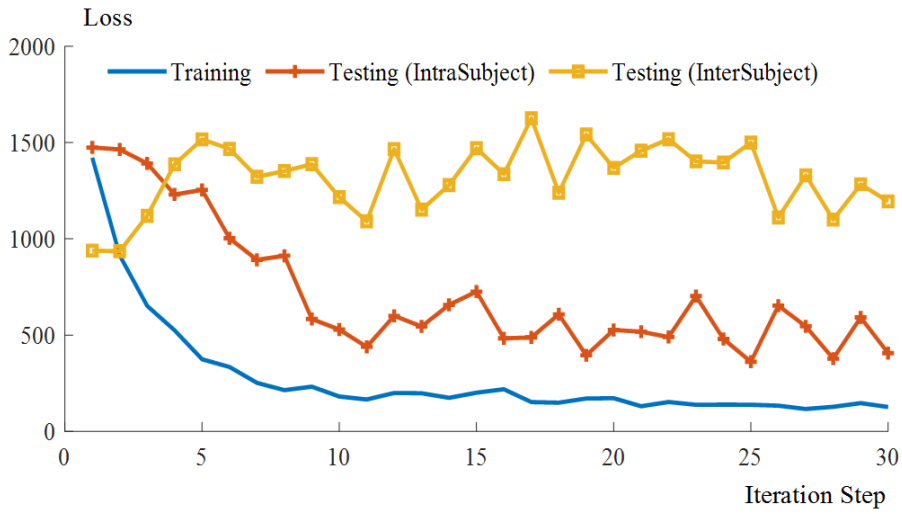


Figure 6.3: Loss performances of CNN during model learning in both intra-subject and inter-subject scenarios.

6.4.2 Learning Process of Regression SDA

In this section, learning process of regression SDA was investigated. Figure 6.4 illustrates convergences of $\mathcal{L}_S$, $\mathcal{L}_T$, and $\mathcal{L}_d$ in the TL process $D^{S6} \rightarrow D^{T1}$. From this figure it can be observed that two regression losses and the domain discrepancy loss could decrease simultaneously via backpropagation. Different from iteration performances in Figure 6.3, the convergence of $\mathcal{L}_S$ was substantially restricted due to the regularisation of $\mathcal{L}_d$, which helped CNN to avoid over-fitting to the low-error regions of D^S. Besides, we can also find that $\mathcal{L}_T$ decreased faster than $\mathcal{L}_S$ even though their coefficients were set to be the same. Similar observations can also be found in many other TL processes in our experiment. A possible reason is that, although samples in D^T_{train} were extremely augmented in the pairwise combinations, the information provided by D^T_{train} is much less than D^S due to its limited size.

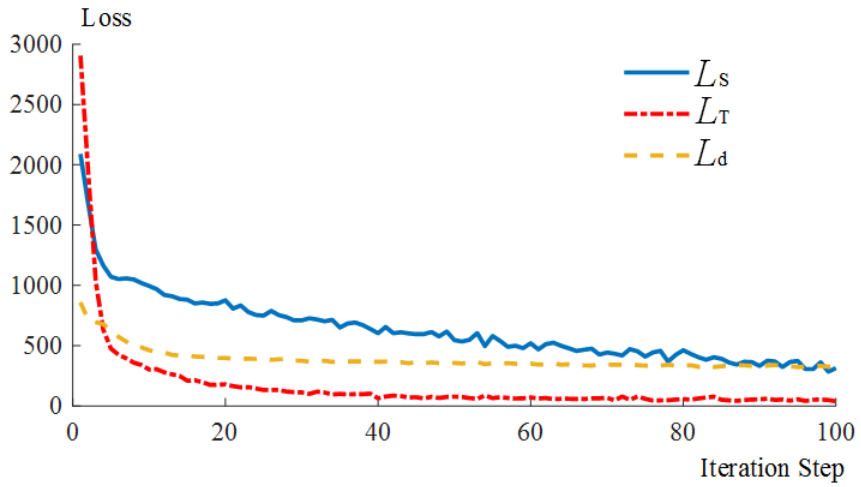


Figure 6.4: Loss performances of regression SDA during model learning in the TL process $D^{S6} \rightarrow D^{T1}$. Specifically, $\mathcal{L}_S$ denotes the regression loss calculated in D^S, $\mathcal{L}_T$ represents the loss in D^T_{train}, $\mathcal{L}_d$ is the domain discrepancy loss combined of a MMD loss and the *RContrastive* loss.

6.4.3 Estimation Performances in Single-Single TL Process

To illustrate the regression performances intuitively, Figure 6.5 plots the estimated trajectories of all listed methods in the transfer process $D^{S6} \rightarrow D^{T1}$. The absolute error of each method with regard to the ground-truth are also summarised in the histograms accordingly. From Figure 6.5 it can be observed that, due to domain shift effects in the inter-subject circumstance, the predicted trajectory of SO is quite distant from the ground-truth. The absolute errors of testing samples are substantially larger than other those of other methods. By contrast, trajectories of JT, FT, and regression SDA are much closer to the ground-truth, which can be also verified by the their better distributions of absolute errors. In particular, the trajectory of SDA can mostly fit the ground-truth, with absolute errors mainly smaller than 50 degrees.

For an explicit comparison, Table 6.1 summarises the regression performances

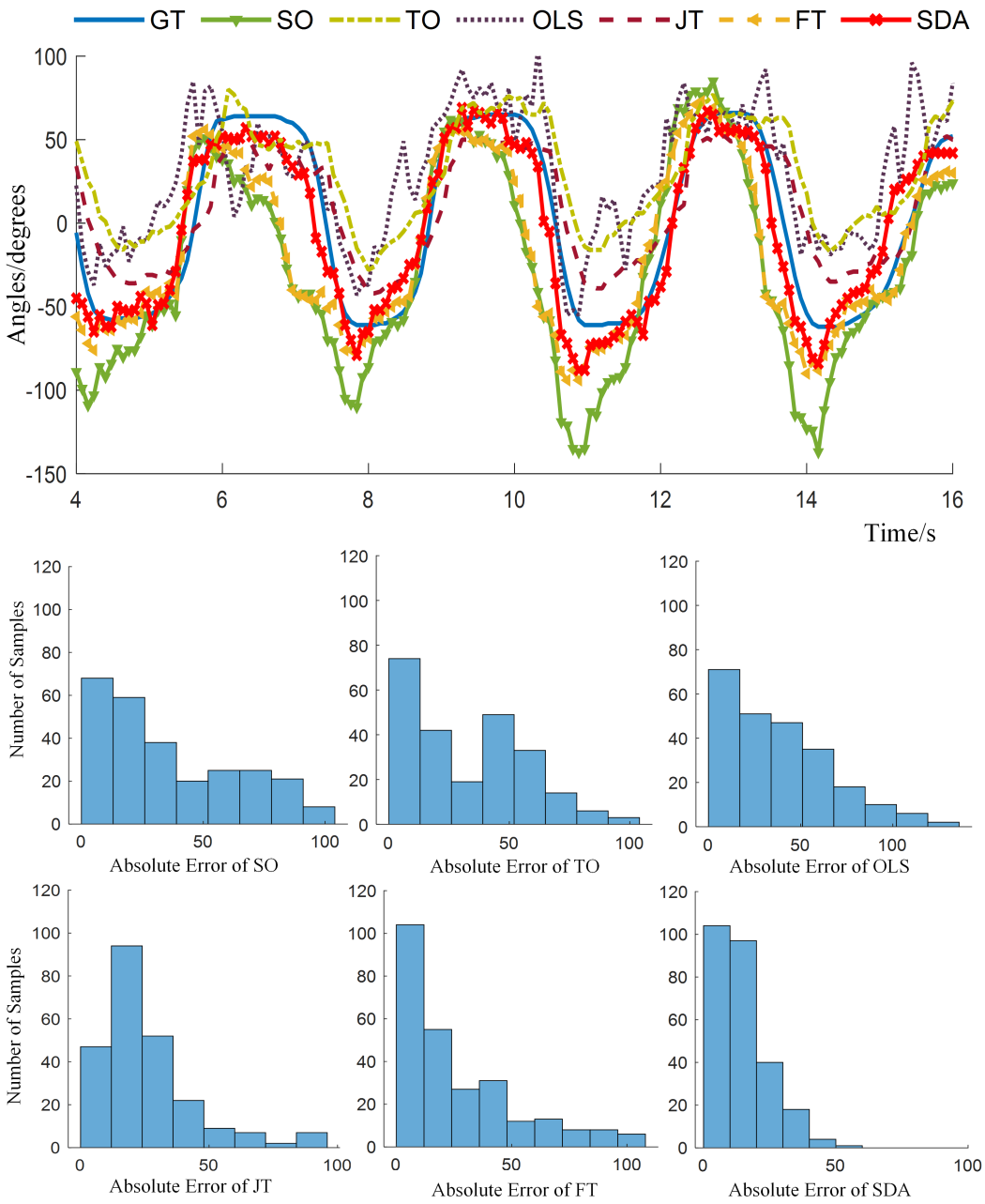


Figure 6.5: Estimation performances (predicted wrist angles and absolute errors with respect to the ground-truth) of all methods in the TL process $D^{S6} \rightarrow D^{T1}$. GT denotes the ground-truth. The R^2 of these four methods are 0.38, 0.59, 0.59, 0.72, respectively. The R^2 of TO and OLS in D^{T1} (TO and OLS are calculated once in each target subject) are 0.43 and 0.51.

of SO, JT, FT, and SDA in the single-single TL processes $D^{Sa} \rightarrow D^{Tb}$. Besides, according to the definition of TO and OLS, these two methods are calculated once in each target subject. The R^2 of TO and OLS for D^{Tb} can be found in Table 6.2. Since trajectories of SO usually differ a lot from the ground-truth (see Figure 6.5) due to the domain shift impact, the R^2 of SO in some TL processes can be negative, such as -0.24 in $D^{S4} \rightarrow D^{T1}$. From Table 6.1 it can be observed that SDA surpasses other methods, especially JT and FT in most cases. Another interesting observation is that in the same row TL performances also vary a lot. This is because the domain shift effects cannot be the same between every two subjects due to different similarities in their biochemical or physiological characteristics.

To better verify the effectiveness of regression SDA, ANOVA is applied for statistical analysis of SO, JT, FT, and SDA in each target subject, and the results can be found in Figure 6.6. As aforementioned, the performances of each method vary substantially among TL processes in each target subject, which can result in large standard deviations (Std). To this end, larger p-values were selected to indicate the significance in statistics (***p-value < 0.001, **p-value < 0.05, and *p-value < 0.1). Since TO and OLS was computed once in each target subject, statistical analysis of TO/OLS are only included in the multiple-single scenario (see Section 6.4.4).

6.4.4 Estimation Performances in Multiple-Single TL Process

In this subsection, performances of SO, TO, OLS, JT, FT, and SDA are compared in multiple-single processes. Table 6.2 lists R^2 of all listed methods in the multiple-single scenario, where for each target subject $D^{Tb}(b = 1 \cdots 8)$, the corresponding D^S is composed of data from the rest seven subjects. To fully exploit the capability of neural network methods, the leave-one-out cross-validation (LOOCV) is applied to tune hyper-parameters, where data of each source subject work alternatively as the validation subject in each TL process. Herein,

Table 6.1: R^2 of SO, JT, FT, and SDA in single-single TL processes $D^{Sa} \rightarrow D^{Tb}$ $(a, b = 1, 2, ..., 8, a \neq b)$.

Target	Method	D^{S1}	D^{S2}	D^{S3}	D^{S4}	D^{S5}	D^{S6}	D^{S7}	D^{S8}	Ave	Std
D^{T1}	SO	—	0.21	0.39	-0.24	0.46	0.38	0.46	0.24	0.27	0.25
	JT	—	0.52	0.55	0.66	0.7	0.59	0.6	0.66	0.61	0.06
	FT	—	0.51	0.55	0.62	0.71	0.59	0.62	0.66	0.61	0.07
	SDA	—	0.55	0.61	0.72	0.73	0.72	0.69	0.69	0.67	0.07
D^{T2}	SO	-0.34	—	-1.03	-0.61	-1.46	-1.66	-1.11	-0.34	-0.94	0.53
	JT	0.32	—	0.45	0.15	0.32	0.52	0.52	0.56	0.41	0.15
	FT	0.15	—	0.28	0.15	0.28	0.58	0.49	0.48	0.34	0.17
	SDA	0.48	—	0.64	0.51	0.41	0.58	0.60	0.61	0.55	0.08
D^{T3}	SO	-0.04	0.14	—	-0.05	0.28	-0.64	-0.98	-1.29	-0.37	0.60
	JT	0.49	0.32	—	0.44	0.38	0.32	0.53	0.52	0.43	0.09
	FT	0.58	0.30	—	0.41	0.30	0.25	0.53	0.72	0.44	0.17
	SDA	0.50	0.44	—	0.64	0.44	0.45	0.70	0.66	0.55	0.11
D^{T4}	SO	0.19	0.10	0.02	—	-0.44	0.31	0.22	0.09	0.07	0.24
	JT	0.44	0.42	0.47	—	0.48	0.39	0.42	0.43	0.43	0.03
	FT	0.41	0.43	0.31	—	0.48	0.39	0.35	0.48	0.41	0.06
	SDA	0.57	0.48	0.54	—	0.52	0.47	0.52	0.62	0.53	0.05
D^{T5}	SO	0.30	0.16	0.42	0.26	—	0.02	0.48	0.14	0.25	0.16
	JT	0.62	0.48	0.61	0.28	—	0.54	0.71	0.47	0.53	0.14
	FT	0.72	0.47	0.61	0.28	—	0.65	0.68	0.70	0.59	0.16
	SDA	0.70	0.65	0.68	0.47	—	0.72	0.72	0.68	0.66	0.09
D^{T6}	SO	0.14	0.17	0.32	0.14	-0.22	—	0.40	-0.63	0.05	0.36
	JT	0.58	0.31	0.22	0.46	0.51	—	0.66	0.54	0.47	0.15
	FT	0.59	0.18	0.45	0.49	0.52	—	0.47	0.38	0.44	0.13
	SDA	0.66	0.46	0.49	0.56	0.61	—	0.69	0.56	0.58	0.08
D^{T7}	SO	-0.68	0.31	0.47	0.43	0.42	0.51	—	0.44	0.27	0.42
	JT	0.69	0.46	0.59	0.63	0.69	0.58	—	0.7	0.62	0.09
	FT	0.69	0.47	0.55	0.68	0.70	0.74	—	0.55	0.63	0.10
	SDA	0.77	0.63	0.67	0.75	0.73	0.73	—	0.74	0.72	0.05
D^{T8}	SO	-0.46	0.19	0.30	0.06	0.38	-0.15	0.08	—	0.06	0.29
	JT	0.65	0.29	0.54	0.60	0.61	0.54	0.51	—	0.53	0.12
	FT	0.61	0.42	0.54	0.61	0.62	0.60	0.45	—	0.55	0.08
	SDA	0.71	0.64	0.58	0.67	0.69	0.71	0.78	—	0.68	0.06

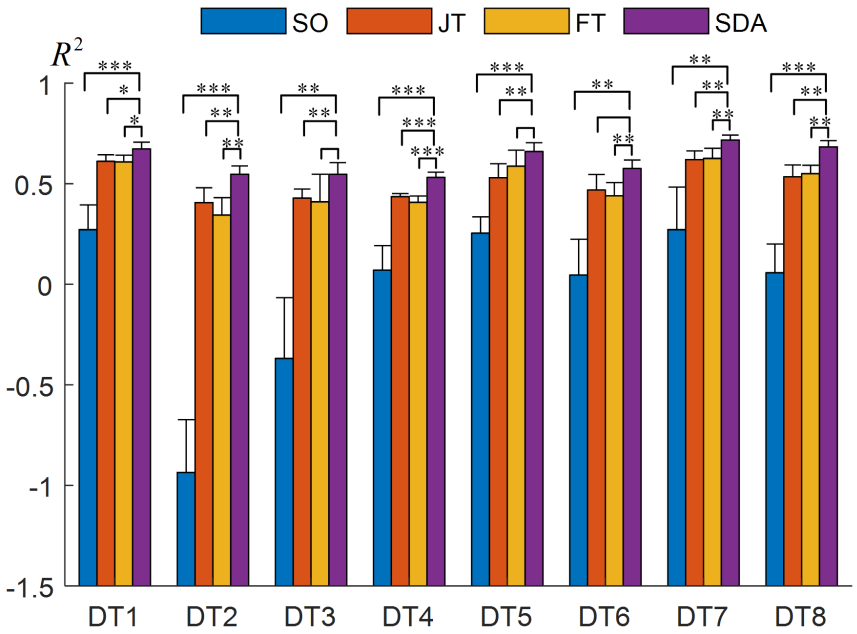


Figure 6.6: Statistical analysis of SO, JT, FT, and regression SDA for each target subject in the single-single transfer learning scenario (***p-value < 0.001, **p-value < 0.05, and *p-value < 0.1). TO and OLS are excluded in this figure since they are computed once in each target subject).

four hyper-parameters are selected to be optimised in SDA, including learning rate β, max training epochs T, loss coefficients γ_1 and γ_2. Each hyper-parameter is given five optional values, i.e. $\beta \in [0.1, 0.05, 0.001, 0.0005, 0.0001]$, $T \in [25, 50, 100, 150, 200]$, $\gamma_1 \in [100, 500, 1000, 2000, 5000]$, and $\gamma_2 \in [0.01, 0.05, 0.1, 0.5, 1.0]$. After LOOCV, the configured hyper-parameters are applied to SDA for the target subject. Statistical analysis is shown in Figure 6.7, from which it can be indicated that regression SDA still obtains the best estimation performance in each process.

Table 6.2: R^2 of all listed methods in TL processes of multiple-single scenario.

Target	SO	TO	OLS	JT	FT	SDA
D^{T1}	0.41	0.43	0.51	0.66	0.67	0.73
D^{T2}	-0.20	0.08	0.23	0.54	0.43	0.60
D^{T3}	-0.06	0.25	0.22	0.55	0.50	0.68
D^{T4}	0.33	0.35	0.46	0.39	0.56	0.60
D^{T5}	0.34	0.37	0.41	0.58	0.61	0.70
D^{T6}	0.34	0.31	0.44	0.46	0.50	0.62
D^{T7}	0.42	0.29	0.24	0.70	0.67	0.76
D^{T8}	0.17	0.41	0.58	0.52	0.52	0.66

6.4.5 Estimation Performances in Source Domains

Apart from better results in D^T, another main advantage of regression SDA over FT is that the former can maintain estimation performances in D^S. Theoretically, without extra guidance to tune CNN weights $\boldsymbol{W}$ for original tasks, FT only learns a final point $\boldsymbol{W}^*$ that yields a low error for D^T but not D^S. This issue is also known as the catastrophic forgetting [205, 206] which is prevalent in conventional deep neural networks when learning new tasks. On the contrary, due to the special design of loss functions in regression SDA, $\boldsymbol{W}^*$ can be kept in the low-error regions of both D^T but not D^S. The differences in the optimisation of CNN weights $\boldsymbol{W}$ via FT and regression SDA are illustrated in Figure 6.8.

In Figure 6.9 the overall performances of FT and regression SDA are further compared on each source domain D^{Sa} after conducting TL processes to the rest seven target domains following the single-single protocol. As we can see, performances of FT on D^{Sa} degraded substantially due to the catastrophic forgetting [206]. By contrast, performances of regression SDA can be maintained in a much better level for every D^{Sa}. Compared with FT, this advantage of regression SDA can effectively boost the model reusability among subjects. Although it is claimed that after FT a specific network could be stored for each subject separately, this strategy might be impractical in real-time applications since an

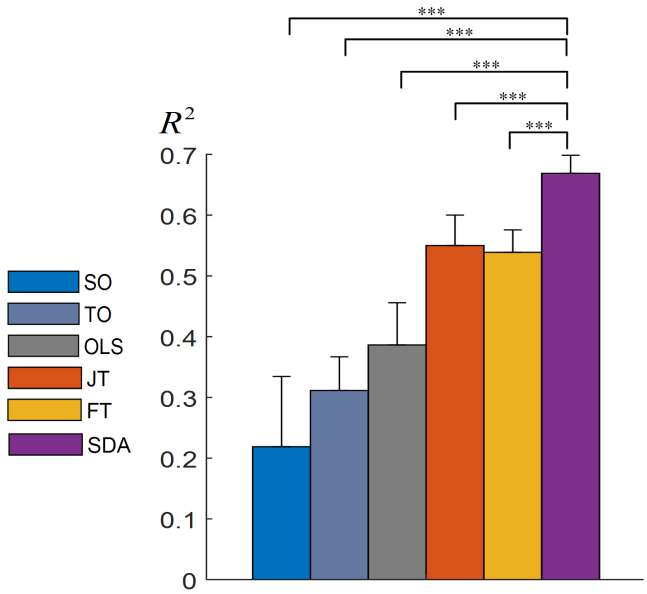
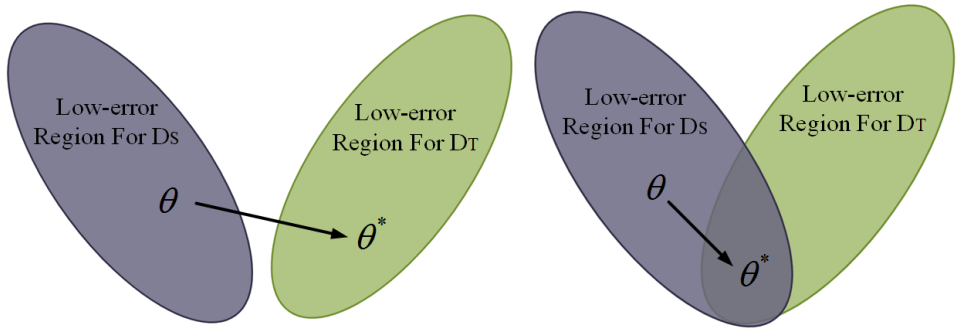


Figure 6.7: Statistical analysis of SO, TO, OLS, JT, FT, and SDA on eight target subjects in the multiple-single TL process (***p-value < 0.001, **p-value < 0.05, *p-value < 0.1).

extra step is then required to distinguish which subject the testing data should belong to.

6.5 Discussion

Domain shift issues are prevalent in sEMG-based motion estimation, particularly when DL models are implemented in the inter-subject circumstance. As illustrated in Figure 6.2, a main reason is that the physiological, anatomical and biochemical characteristics of muscles are highly variable among individuals. In addition, subjects may use different muscle control strategies to produce the same movement [219, 220]. Consequently, results of SO in our experiments (Figure 6.5-6.7, Table 6.1-6.2) indicate that models trained with sEMG from previous subjects may fail to predict accurately on a new subject, which results in a great challenge to the practical application of myoelectric control. In previous literat-



(a) FT (b) SDA

Figure 6.8: Optimisations of CNN weights via FT and regression SDA. The low-error region of two domains will become closer or more overlapped in regression SDA due to the reduction of domain discrepancy.

ures many efforts have been reported to enhance the model generalisation among individuals, including both machine learning approaches [125, 126, 134, 137, 221] and deep learning ones [80, 140, 147]. However, most of these works mainly focused on the hand gestures recognition rather than kinematics estimation, where specific designs were proposed for the classier or to match classification strategies.

To this end, the regression SDA is proposed to reduce domain shift effects on CNN performances in the inter-subject kinematics estimation. According to the experiment results in both single-single and multiple-single scenario it can be concluded that 1) by exploiting information of both source and target domains, the proposed SDA can outperform baseline methods significantly; 2) with help of the discrepancy losses, SDA can further surpass JT which simply combines labelled data of two domains; 3) due to the effectiveness of automatic feature extraction via CNN, deep learning methods, i.e. JT/FT/SDA, are better than OLS which depends heavily on the quality of hand-crafted features; 4) different from FT which suffers from the catastrophic forgetting in the source domains, SDA can maintain good performances in two domains and thus boost the model reusability among subjects.

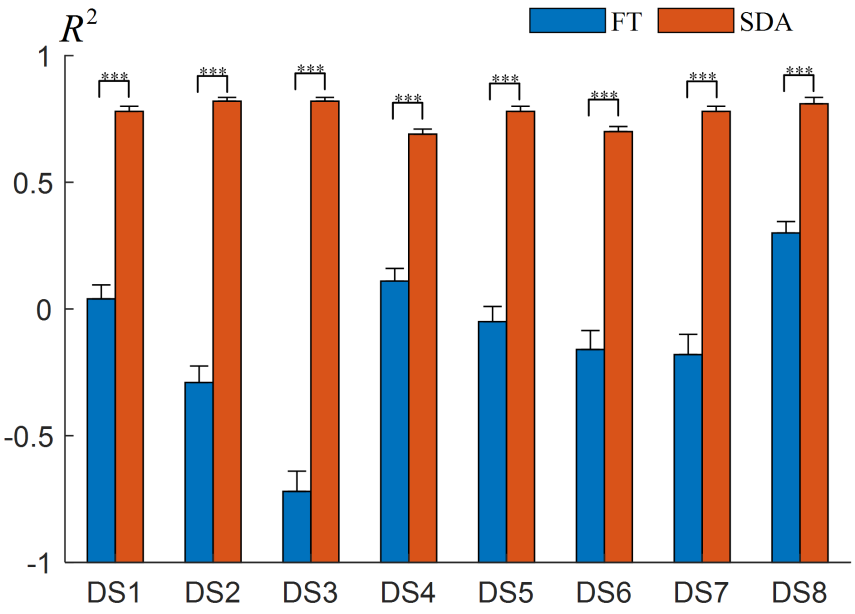


Figure 6.9: Statistical analysis of FT and regression SDA for each source subject in the single-single transfer learning scenario (***p-value < 0.001).

6.5.1 Application of SDA in Cross-Task Transfer

Another interesting observation of SDA is that D^T_{train} and D^T_{test} might be able to come from different tasks. In particular, participants were asked to perform two contractions: 1) wrist flexion (WF) to move the wrist towards to the palm side and then return to neutral position; 2) wrist extension (WE), which starts from neutral position, move the wrist towards to the back-hand side and then return to neutral position. Apparently, both features (muscle activation denoted by sEMG) and labels (wrist angles) are different between two tasks. Herein, both D^S and D^T_{test} are composed of data from WE, whilst D^T_{train} is obtained from WF. In our experiment, it was empirically found that TO, OLS, and FT all performed very poorly since they only utilised D^T_{train} which is irrelevant to D^T_{test}. Therefore, performances of SO, JT, and SDA were particularly compared. Figure 6.10 illustrates the statistical results in the multiple-single scenario, from which it can be found that SDA outperforms both JT and SO significantly (p-value < 0.05 for R^2). Interestingly, there is no significance between JT and SO, indicating that

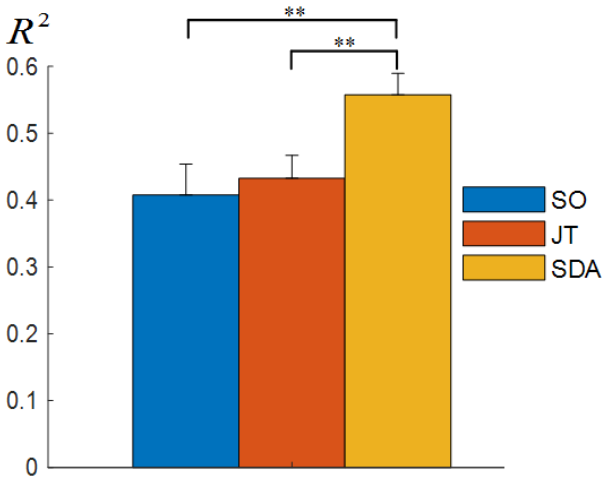


Figure 6.10: Statistical analysis of SO, JT, and SDA in multiple-single TL processes when D^T_{train} and D^T_{test} are from two tasks (**p-value < 0.05, *p-value < 0.1). Specifically, D^S and D^T_{test} are composed of data from wrist extension, whilst D^T_{train} is obtained from wrist flexion.

a simple addition of the irrelevant D^T_{train} may not benefit the transfer learning between different tasks.

6.5.2 Comparison of UDA and SDA

As aforementioned, domain adaptation can be divided into SDA and UDA approaches [122]. Different from SDA, UDA is also be of significance due to the exclusion of extra hardware and time for data relabelling. In fact, our method is a framework which is suitable for both SDA and UDA. Based on the proposed two-stream CNN architecture, a UDA model can be constructed when only the source regression loss $\mathcal{L}_S$ and MMD Loss are included. This setting is similar to those proposed in [208, 209]. To be specific, the loss function of UDA can be expressed as

$$\mathcal{L}\left(\boldsymbol{W} \mid \boldsymbol{X}^{\mathrm{S}}, \boldsymbol{y}^{\mathrm{S}}, \boldsymbol{X}^{\mathrm{T}}\right) = MSE\left(\boldsymbol{W} \mid \boldsymbol{X}^{S}, \boldsymbol{y}^{S}\right) + \gamma_1 MMD^2\left(\boldsymbol{W} \mid \boldsymbol{X}^{\mathrm{S}}, \boldsymbol{X}^{\mathrm{T}}\right). \tag{6.11}$$

The effectiveness of UDA was verified in the multiple-single scenario, and the LOOCV was applied to tune hyper-parameters β, T, and γ_1 (γ_2 was set to be zero in UDA) for each transfer learning process. Figure 6.11 compares the performances of SO, UDA, and SDA for each target subject. As we can see, by minimising the distribution mismatch of D^S and D^T in the latent space, UDA can continuously outperform SO in each target subject. However, due to the lack of target labels to provide discriminative information, the estimation results of UDA are significantly inferior to those of SDA (p-value < 0.01 for R^2 on the basis of ANOVA). Similarly, Xue *et al.* [137] found that the proposed supervised transportation map can be much more accurate than the unsupervised one in the multi-user gesture recognition, and Ameri *et al.* [141] stated that SDA (i.e. fine-tuning in this work) could be more useful than UDA methods in terms of the adaptation to large changes in sEMG signals. The outperformance of SDA over UDA have also been reported in other fields such as computer vision [214] and website detection [222]), etc.

6.6 Conclusions

In this study, the regression SDA is proposed to reduce domain shift effects on CNN performances in the inter-subject circumstance. Based on the two-stream structure, data in both source and target subject can be exploited simultaneously. By adding the discrepancy loss in model training, distribution divergences between two domains can be effectively minimised. The main merit of regression SDA compared with fine-tuning can be summarised as 1) it further improves the estimation accuracy with very limited data in the target domain; 2) it also maintains good performance in original domain and thus boosts the model reusability. Currently, the regression SDA was only examined using offline analysis. In the future work, the real-time usability will be further investigated via online exper-

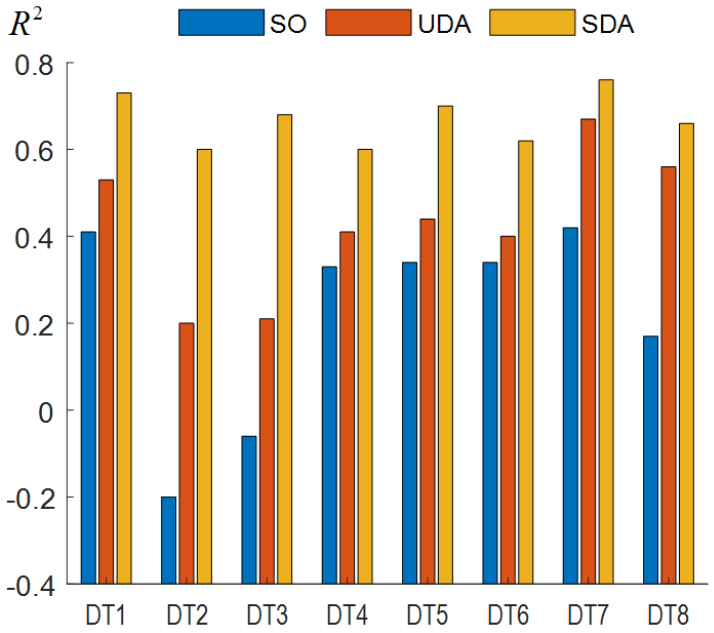


Figure 6.11: Comparison of SO, UDA, and SDA for each target subject in the multiple-single TL processes. The average R^2 for three approaches are 0.22±0.23, 0.43±0.16, and 0.67±0.06, respectively.

iments such as the Fitt's law test. Besides, it is also interesting to validate our method in addressing domain shift effects caused by other compounding factors such as muscle fatigue, arm position changes or electrode shift, etc.

Currently, the costs of hardware set-up and computations during recalibration is still a limitation of the proposed SDA, and there might be some solutions to address these disadvantages. Firstly, quantisation approaches [223] have been widely investigated in recent years to reduce the computation load of CNN models, and will be further explored to enhance SDA approaches in our future work. Secondly, the hardware setups in this experiment can also be further simplified by using the armband for sEMG detection [147] and Leap Motion Controller [45] which provides a cheap and efficient way to track the joint angles as labels. With the acceleration of computation and the simplification of hardware set-up, it can further benefit users by requesting fewer trials for recalibration via SDA.

Chapter 7

Conclusions and Future Work

Herein we summarise the research work conducted in this thesis, conclude the major contributions, and provide recommendations for further development of DL techniques in upper-limb myoelectric systems.

7.1 Conclusions

To develop intuitive and multi-functional myoelectric systems, DL is now becoming a research hotspot in the decoding of human movement intentions from sEMG. However, due to the complexity of upper-limb motions and the non-stability of sEMG signals, decoding performances of DL are still greatly hindered in practical applications. Following the objectives mentioned in Chapter 1, this thesis presented a number of efforts with respect to the usability of DL-based motion estimation, mainly focusing on the improvement of recognition reliability and user-safety, the building of effective regression schemes to better enable SPC, and the reduction of domain shift impacts on DL in the inter-subject scenario. Technical achievements of this thesis can be summarised as below:

1) Confidence estimation for CNN-based gesture recognition: Since user-safety is critical to myoelectric systems, it is highly desirable that unconfident recognition can be identified to avoid potential risks. To this end, Chapter 3 demon-

strated a confidence estimation model for CNN-based gesture recognition. Specifically, the proposed model worked to predict the probability of correctness, i.e. ConfScore, for each classification of CNN. A novel objective function was designed to train parameters of the model. After confidence estimation, a threshold-based rejection process was then applied to reject erroneous classifications. Considering that improper threshold selections can either result in too many recognition errors or a too high control lag, a comprehensive metric was proposed to consider both rejection efficiency and cost. The superiority of ConfScore to commonly used confidence features was firstly verified using data from both six public databases. To further evaluate ConfScore in real-time experiments, an online platform was specifically developed. To summarise, by using confidence-based rejection, the error rate of CNN was reduced significantly with limited loss of correct classifications, thereby enhancing the reliability of PR-based control effectively.

2) CNN-LSTM hybrid framework for joint kinematics estimation: Different from PR approaches that provide discrete predictions on upper-limb movements, regression methods focus on the establishment of a continuous mapping function between sEMG and joint kinematics, and thereby can help to enable a more fluent and natural control scheme. To further improve the regression performances, Chapter 4 introduced a CNN-LSTM hybrid model that combines deep feature extraction and sequence regression efficiently, such that the temporal-spatial correlations of sEMG can be fully exploited. Based on experiments in both intra-session and inter-session scenarios, the proposed hybrid framework was verified to outperform CNN, LSTM and several representative ML approaches, especially when complex wrist movements were activated in multi-DOFs. In contrast to conventional CNN-LSTM interfaces which have to train CNN and LSTM jointly, our framework exploited a separate training strategy such that the computational efficiency and model feasibility could be further improved. In addition, visual exploration was conducted to analyse the distribution of both CNN features and hand-crafted features, providing a deeper insight on the advantage of feature learning in joint kinematics estimation.

3) LSTM-based Kalman filter to enhance sequence regression: Estimation per-

formances of DL can be negatively impacted by noisy measurements of neural activities. By contrast, Kalman filter, which fuses the information of internal transition model and external observation model, can be less affected by measurement noise during the recursive process. In this context, Chapter 5 proposed LSTM-KF to further enhance the sequence regression on high-level features extracted by CNN. Specifically, LSTM-KF adopts the computational graph of Kalman filter but estimates parameters of the transition/observation model and the Kalman gain from data using LSTM modules. With this process, the advantages of Kalman filter and LSTM can be exploited jointly. Herein, the newly proposed two-step hybrid structure, i.e. DKFN, was shown to be more effective than CNN-LSTM in the experiments of both wrist and fingers kinematics estimation.

4) Inter-subject domain adaptation in CNN-based regression scheme: Characteristics of sEMG are highly variable among individuals even when signals are measured from the same location with the same motion. Therefore, a pretrained DL model often degrades substantially when testing on a new subject. To reduce the domain shift effect in the inter-subject scenario, Chapter 6 demonstrated a novel SDA method for CNN-based kinematics estimation. Specifically, a two-stream CNN with shared weights was established to exploit source and target sEMG data simultaneously, such that domain-invariant features could be extracted. To tune CNN weights, both regression losses and a domain discrepancy loss were employed. In particular, the latter is combined of MMD loss and *RContrastive* loss, where MMD minimises the distribution mismatch of two different domains in the latent space, and *RContrastive* provides more discriminative deep features to further boost domain alignment. Experiment results illustrated that regression SDA outperformed the commonly used fine-tuning method in both single-single and multiple-single TL scenarios. Meanwhile, it maintained much better performances than fine-tuning in the original domain.

7.2 Future Work

Despite the progresses achieved in this thesis to promote the robustness, adaptation and reliability of DL techniques, several emerging issues need to be further explored for advancing the development of myoelectric systems. In this section, some research interests in the future work will be discussed.

7.2.1 Data Augmentation

Compared with conventional ML, DL normally has more parameters to train and thereby requires an larger number of labelled data to avoid over-fitting. However, in the context of myoelectric control, generating an extensively large dataset for each subject is a complicated and even unpleasant experience for both patients and researchers due to tiredness in data acquisition, cleaning and labelling. Therefore, data augmentation can be employed to inflate the amount and diversity of available data based on existing datasets. Up to now, a few strategies have been proposed in the previous literature, including the addition of Gaussian noise [40, 42], shift of sEMG images [41], domain specific augmentation such as the electrode displacement [175, 224], and the generation of synthetic data [225]. As suggested by Tsinganos *et al.* [226], it is of vital importance to select appropriate augmentation methods and data sizes to provide adequate and diverse information for DL models. However, so far this topic has not been fully studied. In addition, the generative adversarial network (GAN) and its variations have been primarily investigated to generate EEG signals for data augmentation [227], which may inspire further attempts in DL-based myoelectric control.

7.2.2 Neuromorphic Computing

The success of DL is centred around long-term training and the use of dedicated GPU hardware [228]. However, computational load, associated with the power consumption, can be another critical issue for the wearability of myoelectric control. On the one hand, increases in model complexity, processing

steps (e.g. the post-processing and data augmentation), and the information size (multi-modal fusion, high-density data acquisition, etc.) result in a larger computational cost. On the other hand, wearable systems suffer significantly from limited processing resources and the space available for batteries. Furthermore, it is desirable that the processing time can be reduced as much as possible to produce timely commands for actuators. To address this contradiction, researchers start to investigate neuromorphic computing which exhibit desirable properties including analogue computation, low power consumption, fast inference, event-driven processing, online learning, and massive parallelism [229]. Some primary efforts of neuromorphic computing in myoelectric control can be found in [230–233]. Compared with the traditional ML pipeline, the proposed system exhibited increased inference time and a lower power consumption. Note that it is now possible to design mixed digital-analogue systems [232] to enable conventional ML/DL models in neuromorphic computing, the combination of two techniques can be further explored in myoelectric control.

7.2.3 Immersive Training

As aforementioned, it is always a complex and tedious work to build qualified dataset for sufficient training of ML/DL models. In addition, due to domain shift impacts, model re-calibration, or even re-training, is commonly urged to maintain the estimation accuracy in practical applications and dynamic scenarios. Apparently, it can cause non-negligible burdens for end-users that lacks experience and expertise to collect data properly. Therefore, the involvement of professionals in model training, particularly via a simple, inexpensive and convenient manner, is of importance. A possible solution to the above issue is the immersive training that provides an more engaging and interactive environment during in-home practice. The development of immersive training strategies have evolved from simple graphical user interfaces (GUI) to virtual avatars of real-time practice [234], including the display of sEMG signals as visual feedback [235], designs of computer games [236], and the control of prosthetic simulators [234, 237]. More recently, the implementation of virtual-reality (VR) [238] and augmented-reality

(AR) [239] have gained considerable attentions. By using immersive training, remarkable improvements of PR scheme have been reported in several real-time tests [234]. To summarise, immersive training can potentially increase the motivation of users for data collection and the experience of real-time control. Therefore, more effective immersive training strategies and related design of hardware can be further investigated.

Bibliography

[1] Roberto Merletti and Dario Farina. Surface electromyography for man-machine interfacing in rehabilitation technologies. 2016. xi, 8, 24, 25

[2] Ning Jiang, Hubertus Rehbaum, Ivan Vujaklija, Bernhard Graimann, and Dario Farina. Intuitive, online, simultaneous, and proportional myoelectric control over two degrees-of-freedom in upper limb amputees. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 22(3):501–510, 2013. xi, 24, 25

[3] Mads Jochumsen, Asim Waris, and Ernest Nlandu Kamavuako. The effect of arm position on classification of hand gestures with intramuscular emg. *Biomed Signal Process Control*, 43:1–8, 2018. xi, 17, 29

[4] Benjamin Paaßen, Alexander Schulz, Janne Hahne, and Barbara Hammer. Expectation maximization transfer learning and its application for bionic hand prostheses. *Neurocomputing*, 298:122–133, 2018. xi, 29, 31

[5] T Bao, Z Zhang, S Zaidi, and S Xie. Surface-emg based wrist kinematics estimation using convolutional neural network. In *Proceedings of BSN 2019*. IEEE, 2019. xiii, 72, 77

[6] Yihui Zhao, Zhiqiang Zhang, Zhenhong Li, Zhixin Yang, Abbas A Dehghani-Sanij, and Sheng Q Xie. An emg-driven musculoskeletal model for estimating continuous wrist motion. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 2020. xiv, 101

[7] Cody L McDonald, Sarah Westcott-McCoy, Marcia R Weaver, Juanita Haagsma, and Deborah Kartin. Global prevalence of traumatic non-fatal limb amputation. *Prosthet Orthot Int.*, page 0309364620972258, 2020. 1

[8] Vincent Mendez, Francesco Iberite, Solaiman Shokur, and Silvestro Micera. Current solutions and future trends for robotic prosthetic hands. *Annu. Rev. Control Robot. Auton. Syst.*, 4, 2020. 1

[9] Agamemnon Krasoulis, Sethu Vijayakumar, and Kianoush Nazarpour. Multi-grip classification-based prosthesis control with two emg-imu sensors. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 28(2):508–518, 2019. 1

[10] Alexander E Olsson, Nebojša Malešević, Anders Björkman, and Christian Antfolk. Learning regularized representations of categorically labelled surface emg enables simultaneous and proportional myoelectric control. *J. Neuroeng. Rehabil.*, 18(1):1–19, 2021. 2

[11] Miguel Simão, Nuno Mendes, Olivier Gibaru, and Pedro Neto. A review on electromyography decoding and pattern recognition for human-machine interaction. *IEEE Access*, 7:39564–39582, 2019. 2, 8

[12] Julian Maier, Adam Naber, and Max Ortiz-Catalan. Improved prosthetic control based on myoelectric pattern recognition via wavelet-based denoising. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 26(2):506–514, 2017.

[13] Infinite Biomedical Technologies. Sense. URL `https://www.i-biomed.com/sense.html`. Accessed: 11 May 2021. 2, 24

[14] Tianzhe Bao, Syed Ali Raza Zaidi, Shengquan Xie, Pengfei Yang, and Zhi-Qiang Zhang. A cnn-lstm hybrid model for wrist kinematics estimation using surface electromyography. *IEEE Trans Instrum Meas.*, 70:1–9, 2020. 2, 13, 113

[15] Rami N Khushaba, Erik Scheme, Ali H Al-Timemy, Angkoon Phinyomark, Ahmed Al-Taee, and Adel Al-Jumaily. A long short-term recurrent spatial-

temporal fusion for myoelectric pattern recognition. *Expert Syst. Appl*, 178: 114977, 2021. 2

[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 2, 3, 15

[17] Zhi-Hua Zhou and Ji Feng. Deep forest. *arXiv preprint arXiv:1702.08835*, 2017. 3

[18] Jason W Robertson, Kevin B Englehart, and Erik J Scheme. Effects of confidence-based rejection on usability and error in pattern recognition-based myoelectric control. *IEEE J. Biomed. Health. Inform.*, 23(5):2002–2008, 2018. 3, 23, 38, 39, 41, 66

[19] Ning Jiang, Strahinja Dosen, Klaus-Robert Muller, and Dario Farina. Myoelectric control of artificial limbsâ€"is there a need to change focus?[in the spotlight]. *IEEE Signal Process. Mag.*, 29(5):152–150, 2012. 3

[20] Dario Farina, Ning Jiang, Hubertus Rehbaum, Aleš Holobar, Bernhard Graimann, Hans Dietl, and Oskar C Aszmann. The extraction of neural information from the surface emg for the control of upper-limb prostheses: emerging avenues and challenges. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 22(4):797–809, 2014. 3, 24, 26, 68

[21] Xinyu Jiang, Berj Bardizbanian, Chenyun Dai, Wei Chen, and Edward A Clancy. Data management for transfer learning approaches to elbow emg-torque modeling. *IEEE. Trans. Biomed. Eng.*, 2021. 3, 30

[22] Asim Waris, Imran Khan Niazi, Mohsin Jamil, Kevin Englehart, Winnie Jensen, and Ernest Nlandu Kamavuako. Multiday evaluation of techniques for emg based classification of hand motions. *IEEE J. Biomed. Health Inform.*, 2018. 3, 28, 91, 113

[23] Shaoyang Hua, Congqing Wang, and Xuewei Wu. A novel semg-based force estimation method using deep-learning algorithm. *Complex Intell. Syst.*, pages 1–13, 2021. 8

[24] T Scott Saponas, Desney S Tan, Dan Morris, and Ravin Balakrishnan. Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 515–524, 2008. 8, 29

[25] Erik Scheme and Kevin Englehart. Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *J Rehabil. Res. Dev.*, 48(6), 2011. 10

[26] Angkoon Phinyomark, Franck Quaine, Sylvie Charbonnier, Christine Serviere, Franck Tarpin-Bernard, and Yann Laurillau. Emg feature evaluation for improving myoelectric pattern recognition robustness. *Expert Syst. Appl*, 40(12):4832–4840, 2013. 10, 11, 79, 122

[27] Adriano O Andrade and C Isabelle Andrade. On the relationship between features extracted from emg and force for constant and dynamic protocols. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3392–3395. IEEE, 2012. 10

[28] Silvestro Micera, Jacopo Carpaneto, and Stanisa Raspopovic. Control of hand prostheses using peripheral information. *IEEE Rev. Biomed. Eng.*, 3: 48–68, 2010. 11, 77, 103

[29] Jingpeng Wang. *Pattern recognition-based real-time myoelectric control for anthropomorphic robotic systems: a thesis presented in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Mechatronics at Massey University, Manawatū, New Zealand.* PhD thesis, Massey University, 2019. 11

[30] Daohui Zhang, Xingang Zhao, Jianda Han, and Yiwen Zhao. A comparative study on pca and lda based emg pattern recognition for anthropomorphic robotic hand. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4850–4855. IEEE, 2014. 11

[31] Bernard Hudgins, Philip Parker, and Robert N Scott. A new strategy for multifunction myoelectric control. *IEEE. Trans. Biomed. Eng.*, 40(1):82–94, 1993. 11

[32] Ganesh R Naik and Hung T Nguyen. Nonnegative matrix factorization for the identification of emg finger movements: Evaluation using matrix analysis. *IEEE J. Biomed. Health. Inform.*, 19(2):478–485, 2014. 11

[33] Jie Liu and Ping Zhou. A novel myoelectric pattern recognition strategy for hand function restoration after incomplete cervical spinal cord injury. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 21(1):96–103, 2012. 11

[34] Ganesh R Naik, Ali H Al-Timemy, and Hung T Nguyen. Transradial amputee gesture classification using an optimal number of semg sensors: an approach using ica clustering. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 24 (8):837–846, 2015. 11

[35] Han-Jeong Hwang, Janne Mathias Hahne, and Klaus-Robert Müller. Channel selection for simultaneous and proportional myoelectric prosthesis control of multiple degrees-of-freedom. *J. Neural Eng.*, 11(5):056008, 2014. 11, 27

[36] Edward A Clancy, Carlos Martinez-Luna, Marek Wartenberg, Chenyun Dai, and Todd R Farrell. Two degrees of freedom quasi-static emg-force at the wrist using a minimum number of electrodes. *J. Electromyogr. Kinesiol.*, 34:24–36, 2017. 11

[37] Ulysse Côté-Allard, Evan Campbell, Angkoon Phinyomark, François Laviolette, Benoit Gosselin, and Erik Scheme. Interpreting deep learning features for myoelectric control: A comparison with handcrafted features. *Front. Bioeng. Biotechnol.*, 8:158, 2020. 13, 34

[38] Wentao Wei, Yongkang Wong, Yu Du, Yu Hu, Mohan Kankanhalli, and Weidong Geng. A multi-stream convolutional neural network for semg-

based gesture recognition in muscle-computer interface. *Pattern Recognit. Lett.*, 2017. 13, 39, 49, 52

[39] Yann LeCun et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 20:5, 2015. 13, 41

[40] Weidong Geng, Yu Du, Wenguang Jin, Wentao Wei, Yu Hu, and Jiajun Li. Gesture recognition by instantaneous surface emg images. *Sci. Rep.*, 6: 36571, 2016. 13, 41, 142

[41] Yu Du, Wenguang Jin, Wentao Wei, Yu Hu, and Weidong Geng. Surface emg-based inter-session gesture recognition enhanced by deep domain adaptation. *Sensors*, 17(3):458, 2017. 34, 89, 142

[42] Manfredo Atzori, Matteo Cognolato, and Henning Müller. Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. *Front. Neurorobot.*, 10:9, 2016. 13, 41, 49, 52, 75, 82, 89, 142

[43] Wentao Wei, Qingfeng Dai, Yongkang Wong, Yu Hu, Mohan Kankanhalli, and Weidong Geng. Surface-electromyography-based gesture recognition by multi-view deep learning. *IEEE. Trans. Biomed. Eng.*, 66(10):2964–2973, 2019. 13, 39, 48, 52

[44] Zhen Ding, Chifu Yang, Zhihong Tian, Chunzhi Yi, Yunsheng Fu, and Feng Jiang. semg-based gesture recognition with convolution neural networks. *Sustainability*, 10(6):1865, 2018. 13, 39, 49

[45] Fernando Quivira, Toshiaki Koike-Akino, Ye Wang, and Deniz Erdogmus. Translating semg signals to continuous hand poses using recurrent neural networks. In *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pages 166–169. IEEE, 2018. 13, 69, 98, 138

[46] Teodor-Adrian Teban, Radu-Emil Precup, Elena-Cristina Lunca, Adriana Albu, Claudia-Adina Bojan-Dragos, and Emil M Petriu. Recurrent neural

network models for myoelectricbased control of a prosthetic hand. In *2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)*, pages 603–608. IEEE, 2018. 69, 91

[47] Yunan He, Osamu Fukuda, Nan Bu, Hiroshi Okumura, and Nobuhiko Yamaguchi. Surface emg pattern recognition using long short-term memory combined with multilayer perceptron. In *2018 40th Annual International Conference of the IEEE EMBC*, pages 5636–5639. IEEE, 2018. 70

[48] Ali Samadani. Gated recurrent neural networks for emg-based hand gesture classification. a comparative study. In *2018 40th Annual International Conference of the IEEE EMBC*, pages 1–4. IEEE, 2018. 13, 70, 98

[49] Muhammad Zia ur Rehman, Syed Omer Gilani, Asim Waris, Imran Khan Niazi, Gregory Slabaugh, Dario Farina, and Ernest Nlandu Kamavuako. Stacked sparse autoencoders for emg-based classification of hand motions: A comparative multi day analyses between surface and intramuscular emg. *Appl. Sci.*, 8(7):1126, 2018. 16

[50] Muhammad Zia ur Rehman, Asim Waris, Syed Omer Gilani, Mads Jochumsen, Imran Khan Niazi, Mohsin Jamil, Dario Farina, and Ernest Nlandu Kamavuako. Multiday emg-based classification of hand motions with deep learning techniques. *Sensors*, 18(8):2497, 2018. 16

[51] Domenico Buongiorno, Cristian Camardella, Giacomo Donato Cascarano, Luis Pelaez Murciego, Michele Barsotti, Irio De Feudis, Antonio Frisoli, and Vitoantonio Bevilacqua. An undercomplete autoencoder to extract muscle synergies for motor intention detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019. 16

[52] Xuhui Hu, Hong Zeng, Aiguo Song, and Dapeng Chen. Robust continuous hand motion recognition using wearable array myoelectric sensor. *IEEE Sens. J.*, 2021. 16

[53] Ivan Vujaklija, Vahid Shalchyan, Ernest N Kamavuako, Ning Jiang, Hamid R Marateb, and Dario Farina. Online mapping of emg signals into kinematics by autoencoding. *J. Neuroeng. Rehabil.*, 15(1):1–9, 2018. 16

[54] Guangyu Jia, Hak-Keung Lam, Junkai Liao, and Rong Wang. Classification of electromyographic hand gesture signals using machine learning techniques. *Neurocomputing*, 401:236–248, 2020. 16

[55] Mingde Zheng, Michael Crouch, and Michael S Eggleston. Surface electromyography as a natural human-machine interface: A review. *arXiv preprint arXiv:2101.04658*, 2021. 17

[56] Ethan C Hill, Terry J Housh, CM Smith, Kristen C Cochrane, NDM Jenkins, Joel T Cramer, Richard J Schmidt, and GO Johnson. Effect of sex on torque, recovery, emg, and mmg responses to fatigue. *J. Musculoskelet Neuronal Interact*, 16(4):310, 2016. 17, 113

[57] Jiayuan He, Dingguo Zhang, Ning Jiang, Xinjun Sheng, Dario Farina, and Xiangyang Zhu. User adaptation in long-term, open-loop myoelectric training: implications for emg pattern recognition in prosthesis control. *J. Neural Eng.*, 12(4):046005, 2015. 17

[58] Lizhi Pan, Dingguo Zhang, Ning Jiang, Xinjun Sheng, and Xiangyang Zhu. Improving robustness against electrode shift of high density emg for myoelectric control through common spatial patterns. *J. Neuroeng. Rehabil.*, 12(1):110, 2015. 17, 113

[59] Hao Xu and Anbin Xiong. Advances and disturbances in semg-based intentions and movements recognition: A review. *IEEE Sens. J.*, 2021. 18

[60] Roger M Enoka and Jacques Duchateau. Muscle fatigue: what, why and how it influences muscle function. *J. Physiol.*, 586(1):11–23, 2008. 18

[61] Mario Cifrek, Vladimir Medved, Stanko Tonković, and Saša Ostojić. Surface emg based muscle fatigue evaluation in biomechanics. *Clin. Biomech.*, 24 (4):327–340, 2009. 18

[62] Dennis Tkach, He Huang, and Todd A Kuiken. Study of stability of time-domain features for electromyographic pattern recognition. *J. Neuroeng. Rehabil.*, 7(1):1–13, 2010. 18

[63] Iris Kyranou, Sethu Vijayakumar, and Mustafa Suphi Erden. Causes of performance degradation in non-invasive electromyographic pattern recognition in upper limb prostheses. *Front. Neurorobot.*, 12:58, 2018. 18, 19

[64] L Hargrove, K Englehart, and Bernard Hudgins. The effect of electrode displacements on pattern recognition based myoelectric control. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2203–2206. IEEE, 2006. 18, 19

[65] Alexander Boschmann and Marco Platzner. Reducing classification accuracy degradation of pattern recognition based myoelectric control caused by electrode shift using a high density electrode array. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4324–4327. IEEE, 2012.

[66] Jiayuan He, Xinjun Sheng, Xiangyang Zhu, and Ning Jiang. Electrode density affects the robustness of myoelectric pattern recognition system with and without electrode shift. *IEEE J. Biomed. Health Inform*, 23(1): 156–163, 2018. 18, 19

[67] Yanjuan Geng, Oluwarotimi Williams Samuel, Yue Wei, and Guanglin Li. Improving the robustness of real-time myoelectric pattern recognition against arm position changes in transradial amputees. *Biomed Res. Int.*, 2017, 2017. 19

[68] Erik Scheme, A Fougner, Øyvind Stavdahl, Adrian DC Chan, and Kevin Englehart. Examining the adverse effects of limb position on pattern recognition based myoelectric control. In *2010 annual international conference of the IEEE engineering in medicine and biology*, pages 6337–6340. IEEE, 2010. 19

[69] Jianwei Liu, Dingguo Zhang, Xinjun Sheng, and Xiangyang Zhu. Quantification and solutions of arm movements effect on semg pattern recognition. *Biomed Signal Process Control*, 13:189–197, 2014. 19

[70] Dapeng Yang, Wei Yang, Qi Huang, and Hong Liu. Classification of multiple finger motions during dynamic upper limb movements. *IEEE J. Biomed. Health Inform*, 21(1):134–141, 2015. 19

[71] Anders Fougner, Erik Scheme, Adrian DC Chan, Kevin Englehart, and Øyvind Stavdahl. Resolving the limb position effect in myoelectric pattern recognition. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 19(6):644–651, 2011. 19

[72] Yanjuan Geng, Ping Zhou, and Guanglin Li. Toward attenuating the impact of arm positions on electromyography pattern-recognition based motion classification in transradial amputees. *J. Neuroeng. Rehabil.*, 9(1):1–11, 2012.

[73] Jianwei Liu, Dingguo Zhang, Jiayuan He, and Xiangyang Zhu. Effect of dynamic change of arm position on myoelectric pattern recognition. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1470–1475. IEEE, 2012. 19

[74] Kevin Englehart and Bernard Hudgins. A robust, real-time control scheme for multifunction myoelectric control. *IEEE. Trans. Biomed. Eng.*, 50(7): 848–854, 2003. 20

[75] Adrian DC Chan and Kevin B Englehart. Continuous myoelectric control for powered prostheses using hidden markov models. *IEEE. Trans. Biomed. Eng.*, 52(1):121–124, 2004. 20

[76] Seongjung Kim, Jongman Kim, Bummo Koo, Taehee Kim, Haneul Jung, Sehoon Park, Seunggi Kim, and Youngho Kim. Development of an armband emg module and a pattern recognition algorithm for the 5-finger myoelectric hand prosthesis. *Int. J. Precis. Eng. Man.*, 20(11):1997–2006, 2019.

[77] Ganesh R Naik, S Easter Selvan, and Hung T Nguyen. Single-channel emg classification with ensemble-empirical-mode-decomposition-based ica for diagnosing neuromuscular disorders. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 24(7):734–743, 2015.

[78] Rami N Khushaba, Sarath Kodagoda, Maen Takruri, and Gamini Dissanayake. Toward improved control of prosthetic fingers using surface electromyogram (emg) signals. *Expert Syst. Appl*, 39(12):10731–10738, 2012. 20

[79] Joel Falk-Dahlin. Evaluation of post-processing strategies for simultaneous pattern recognition based myoelectric prosthetic control. Master's thesis, 2013. 20

[80] Xiaolong Zhai, Beth Jelfs, Rosa HM Chan, and Chung Tin. Self-recalibrating surface emg pattern recognition for neuroprosthesis control based on convolutional neural network. *Front. Neurosci*, 11:379, 2017. 20, 39, 134

[81] Md Ferdous Wahid, Reza Tafreshi, and Reza Langari. A multi-window majority voting strategy to improve hand gesture recognition accuracies using electromyography signal. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 28 (2):427–436, 2019. 20

[82] Miguel Simao, Pedro Neto, and Olivier Gibaru. Emg-based online classification of gestures with recurrent neural networks. *Pattern Recognit. Lett.*, 128:45–51, 2019. 20

[83] Ali H Al-Timemy, Rami N Khushaba, and Javier Escudero. A comparison of post-processing techniques on the performance of emg based pattern recognition system for the transradial amputees. In *2016 3rd MECBME*, pages 46–49. IEEE, 2016. 21

[84] Ann M Simon, Levi J Hargrove, Blair A Lock, and Todd A Kuiken. A decision-based velocity ramp for minimizing the effect of misclassifications

during real-time pattern recognition control. *IEEE. Trans. Biomed. Eng.*, 58(8):2360–2368, 2011. 21

[85] Eric J Earley, Levi J Hargrove, and Todd A Kuiken. Dual window pattern recognition classifier for improved partial-hand prosthesis control. *Front. Neurosci*, 10:58, 2016. 21

[86] Kenneth R Lyons and Sanjay S Joshi. Upper limb prosthesis control for high-level amputees via myoelectric recognition of leg gestures. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 26(5):1056–1066, 2018.

[87] Lauren H Smith, Todd A Kuiken, and Levi J Hargrove. Real-time simultaneous and proportional myoelectric control using intramuscular emg. *J. Neural Eng.*, 11(6):066013, 2014.

[88] Lauren H Smith, Todd A Kuiken, and Levi J Hargrove. Evaluation of linear regression simultaneous myoelectric control using intramuscular emg. *IEEE. Trans. Biomed. Eng.*, 63(4):737–746, 2015. 21

[89] Osamu Fukuda, Toshio Tsuji, and Makoto Kaneko. A human supporting manipulator based on manual control using emg signals. *J. Robot. Society JPN*, 18(3):387–394, 2000. 21

[90] Osamu Fukuda, Toshio Tsuji, Makoto Kaneko, and Akira Otsuka. A human-assisting manipulator teleoperated by emg signals and arm motions. *IEEE Trans. Robot. Autom.*, 19(2):210–222, 2003. 21

[91] Jonathon W Sensinger, Blair A Lock, and Todd A Kuiken. Adaptive pattern recognition of myoelectric signals: exploration of conceptual framework and practical algorithms. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 17(3):270–278, 2009. 22, 35

[92] Erik J Scheme, Bernard S Hudgins, and Kevin B Englehart. Confidence-based rejection for improved pattern recognition myoelectric control. *IEEE Trans. Biomed. Eng.*, 60(6):1563–1570, 2013. 22, 38, 41

[93] Sebastian Amsüss, Peter M Goebel, Ning Jiang, Bernhard Graimann, Liliana Paredes, and Dario Farina. Self-correcting pattern recognition system of surface emg signals for upper limb prosthesis control. *IEEE. Trans. Biomed. Eng.*, 61(4):1167–1176, 2013. 22, 38

[94] Erik Scheme and K Englehart. A comparison of classification based confidence metrics for use in the design of myoelectric control systems. In *2015 37th annual international conference of the IEEE EMBC*, pages 7278–7283. IEEE, 2015. 23, 38, 41, 45, 47, 66

[95] Otto Bock. Myo plus pattern recognition. URL https://www.ottobock.co.uk/prosthetics/upper-limb-prosthetics/product-systems/myo-plus/. Accessed: 11 May 2021. 24

[96] Coapt. Coapt pattern recognition. URL https://coaptengineering.com/pattern-recognition. Accessed: 11 May 2021. 24

[97] J Ning, S Dosen, KR Muller, and D Farina. Myoelectric control of artificial limbsâ€"is there a need to change focus?[in the spotlight]. *IEEE Signal Process. Mag.*, 29(5):152–150, 2012. 24

[98] Chuang Lin, Binghui Wang, Ning Jiang, and Dario Farina. Robust extraction of basis functions for simultaneous and proportional myoelectric control via sparse non-negative matrix factorization. *J. Neural Eng.*, 15(2): 026017, 2018. 24, 27

[99] Silvia Muceli and Dario Farina. Simultaneous and proportional estimation of hand kinematics from emg during mirrored movements at multiple degrees-of-freedom. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 20(3):371–378, 2011. 26, 79, 122

[100] Ning Jiang, Silvia Muceli, Bernhard Graimann, and Dario Farina. Effect of arm position on the prediction of kinematics from emg in amputees. *Med. Biol. Eng. Comput.*, 51(1):143–151, 2013.

[101] Qichuan Ding, Jianda Han, and Xingang Zhao. Continuous estimation of human multi-joint angles from semg using a state-space model. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 25(9):1518–1528, 2016. 26

[102] Janne M Hahne, Felix Biessmann, Ning Jiang, Hubertus Rehbaum, Dario Farina, Frank C Meinecke, K-R Müller, and Lucas C Parra. Linear and nonlinear regression techniques for simultaneous and proportional myoelectric control. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 22(2):269–279, 2014. 26, 27, 78, 81

[103] Koushik Bakshi, M Manjunatha, and CS Kumar. Estimation of continuous and constraint-free 3 dof wrist movements from surface electromyogram signal using kernel recursive least square tracker. *Biomed Signal Process Control*, 46:104–115, 2018. 26, 76

[104] Agamemnon Krasoulis, Sethu Vijayakumar, and Kianoush Nazarpour. Effect of user practice on prosthetic finger control with an intuitive myoelectric decoder. *Front. Neurosci.*, 13:891, 2019. 26, 102, 103

[105] Qin Zhang, Te Pi, Runfeng Liu, and Caihua Xiong. Simultaneous and proportional estimation of multijoint kinematics from emg signals for myocontrol of robotic hands. *IEEE/ASME Trans. Mechatron.*, 25(4):1953–1960, 2020. 26, 27

[106] Ning Jiang, Kevin B Englehart, and Philip A Parker. Extracting simultaneous and proportional neural control information for multiple-dof prostheses from the surface electromyographic signal. *IEEE. Trans. Biomed. Eng.*, 56 (4):1070–1080, 2008. 26

[107] Chenyun Dai, Ziling Zhu, Carlos Martinez-Luna, Thane R Hunt, Todd R Farrell, and Edward A Clancy. Two degrees of freedom, dynamic, hand-wrist emg-force using a minimum number of electrodes. *J. Electromyogr. Kinesiol.*, 47:10–18, 2019. 26, 27

[108] Johnny LG Nielsen, Steffen Holmgaard, Ning Jiang, Kevin B Englehart, Dario Farina, and Phil A Parker. Simultaneous and proportional force estimation for multifunction myoelectric prostheses using mirrored bilateral training. *IEEE. Trans. Biomed. Eng.*, 58(3):681–688, 2010. 26

[109] Ning Jiang, Johnny LG Vest-Nielsen, Silvia Muceli, and Dario Farina. Emg-based simultaneous and proportional estimation of wrist/hand kinematics in uni-lateral trans-radial amputees. *J. Neuroeng. Rehabil.*, 9(1):1–11, 2012. 26

[110] Ali Ameri, Mohammad Ali Akhaee, Erik Scheme, and Kevin Englehart. Regression convolutional neural network for improved simultaneous emg control. *J. Neural Eng.*, 2019. 26, 39, 69, 75, 113

[111] Changmok Choi and Jung Kim. Synergy matrices to estimate fluid wrist movements by surface electromyography. *Med Eng Phys*, 33(8):916–923, 2011. 27

[112] Ali Ameri, Ernest N Kamavuako, Erik J Scheme, Kevin B Englehart, and Philip A Parker. Support vector regression for improved real-time, simultaneous myoelectric control. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 22 (6):1198–1209, 2014. 27

[113] Ali Ameri, Erik J Scheme, Kevin B Englehart, and Philip A Parker. Bagged regression trees for simultaneous myoelectric force estimation. In *2014 22nd Iranian Conference on Electrical Engineering (ICEE)*, pages 2000–2003. IEEE, 2014. 27, 81

[114] Ning Jiang, Ivan Vujaklija, Hubertus Rehbaum, Bernhard Graimann, and Dario Farina. Is accurate mapping of emg signals on kinematics needed for precise online myoelectric control? *IEEE Trans. Neural Syst. Rehabil. Eng.*, 22(3):549–558, 2013. 27

[115] Feiping Nie, Dong Xu, Ivor Wai-Hung Tsang, and Changshui Zhang. Flexible manifold embedding: A framework for semi-supervised and unsuper-

vised dimension reduction. *IEEE Trans. Image Process.*, 19(7):1921–1932, 2010. 27

[116] Janne M Hahne, Meike A Schweisfurth, Mario Koppe, and Dario Farina. Simultaneous control of multiple functions of bionic hand prostheses: Performance and robustness in end users. *Sci. Robot.*, 3(19), 2018. 27

[117] Ali Ameri, Erik J Scheme, Ernest Nlandu Kamavuako, Kevin B Englehart, and Philip A Parker. Real-time, simultaneous myoelectric control using force and position-based training paradigms. *IEEE. Trans. Biomed. Eng.*, 61(2):279–287, 2013. 27

[118] Janne M Hahne, Sven Dähne, Han-Jeong Hwang, Klaus-Robert Müller, and Lucas C Parra. Concurrent adaptation of human and machine improves simultaneous and proportional myoelectric control. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 23(4):618–627, 2015. 27

[119] Han-Jeong Hwang, Janne Mathias Hahne, and Klaus-Robert Müller. Real-time robustness evaluation of regression based myoelectric control against arm position change and donning/doffing. *Plos One*, 12(11):e0186318, 2017. 27

[120] Sebastian Amsuess, Ivan Vujaklija, Peter Goebel, Aidan D Roche, Bernhard Graimann, Oskar C Aszmann, and Dario Farina. Context-dependent upper limb prosthesis control for natural and robust use. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 24(7):744–753, 2015. 27

[121] Yuan-Pin Lin and Tzyy-Ping Jung. Improving eeg-based emotion classification using conditional transfer learning. *Front. Hum. Neurosci.*, 11:334, 2017. 28, 114

[122] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 29, 116, 119, 124, 136

[123] Francesco Orabona, Claudio Castellini, Barbara Caputo, Angelo Emanuele Fiorilla, and Giulio Sandini. Model adaptation with least-squares svm for

adaptive hand prosthetics. In *2009 IEEE International Conference on Robotics and Automation*, pages 2897–2903. IEEE, 2009. 30

[124] Rita Chattopadhyay, Jieping Ye, and Sethuraman Panchanathan. Transfer learning framework for early detection of fatigue using non-invasive surface electromyogram signals (semg). In *2011 AAAI Spring Symposium Series*, 2011. 30

[125] Takamitsu Matsubara and Jun Morimoto. Bilinear modeling of emg signals to extract user-independent features for multiuser myoelectric interface. *IEEE. Trans. Biomed. Eng.*, 60(8):2205–2213, 2013. 30, 134

[126] Yingwei Zhang, Yiqiang Chen, Hanchao Yu, Xiaodong Yang, and Wang Lu. Dual layer transfer learning for semg-based user-independent gesture recognition. *Pers. Ubiquitous Comput.*, pages 1–12, 2020. 30, 134

[127] Xinpu Chen, Dingguo Zhang, and Xiangyang Zhu. Application of a self-enhancing classification method to electromyography pattern recognition for multifunctional prosthesis control. *J. Neuroeng. Rehabil.*, 10(1):1–13, 2013. 30

[128] Jianwei Liu, Xinjun Sheng, Dingguo Zhang, Jiayuan He, and Xiangyang Zhu. Reduced daily recalibration of myoelectric prosthesis classifiers based on domain adaptation. *IEEE J. Biomed. Health. Inform.*, 20(1):166–176, 2014. 30

[129] Xiangyang Zhu, Jianwei Liu, Dingguo Zhang, Xinjun Sheng, and Ning Jiang. Cascaded adaptation framework for fast calibration of myoelectric control. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 25(3):254–264, 2016. 30

[130] Cosima Prahm, Benjamin Paassen, Alexander Schulz, Barbara Hammer, and Oskar Aszmann. Transfer learning for rapid re-calibration of a myoelectric prosthesis after electrode shift. In *Converging clinical and engineering research on neurorehabilitation II*, pages 153–157. Springer, 2017. 31

[131] Marina M-C Vidovic, Han-Jeong Hwang, Sebastian Amsüss, Janne M Hahne, Dario Farina, and Klaus-Robert Müller. Improving the robustness of myoelectric pattern recognition for upper limb prostheses by covariate shift adaptation. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 24(9):961–970, 2015. 31

[132] Suguru Kanoga and Atsunori Kanemura. Assessing the effect of transfer learning on myoelectric control systems with three electrode positions. In *2018 IEEE International Conference on Industrial Technology (ICIT)*, pages 1478–1483. IEEE, 2018. 31

[133] Myong Chol Jung, Rifai Chai, Jinchuan Zheng, and Hung Nguyen. Enhanced myoelectric control against arm position change with weighted recursive gaussian process. *Neural. Comput. Appl.*, pages 1–14, 2021. 31

[134] Rami N Khushaba. Correlation analysis of electromyogram signals for multiuser myoelectric interfaces. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 22 (4):745–755, 2014. 31, 35, 134

[135] Juan Cheng, Fulin Wei, Chang Li, Yu Liu, Aiping Liu, and Xun Chen. Position-independent gesture recognition using semg signals via canonical correlation analysis. *Comput. Biol. Med.*, 103:44–54, 2018. 31, 35

[136] Zhe Fan, Zhong Wang, Guanglin Li, and Ruomei Wang. A canonical correlation analysis based emg classification algorithm for eliminating electrode shift effect. In *2016 38th Annual International Conference of the IEEE EMBC*, pages 867–870. IEEE, 2016. 31

[137] Bo Xue, Le Wu, Kun Wang, Xu Zhang, Juan Cheng, Xiang Chen, and Xun Chen. Multiuser gesture recognition using semg signals via canonical correlation analysis and optimal transport. *Comput. Biol. Med.*, 130:104188, 2021. 31, 35, 134, 137

[138] Evan Campbell, Angkoon Phinyomark, and Erik Scheme. Deep cross-user

models reduce the training burden in myoelectric control. *Front. Neurosci*, 15, 2021. 31, 34

[139] Weiming Wang, Biao Chen, Peng Xia, Jie Hu, and Yinghong Peng. Sensor fusion for myoelectric control based on deep learning with recurrent convolutional neural networks. *Artif. Organs*, 42(9):E272–E282, 2018. 32, 35, 114

[140] Keun-Tae Kim, Cuntai Guan, and Seong-Whan Lee. A subject-transfer framework based on single-trial emg analysis using convolutional neural networks. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 28(1):94–103, 2019. 32, 114, 134

[141] Ali Ameri, Mohammad Ali Akhaee, Erik Scheme, and Kevin Englehart. A deep transfer learning approach to reducing the effect of electrode shift in emg pattern recognition-based control. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 28(2):370–379, 2019. 32, 35, 71, 114, 116, 122, 137

[142] Zhipeng Yu, Jianghai Zhao, Yucheng Wang, Linglong He, and Shaonan Wang. Surface emg-based instantaneous hand gesture recognition using convolutional neural network with the transfer learning method. *Sensors*, 21(7):2540, 2021. 32

[143] Fatih Demir, Varun Bajaj, Melih C Ince, Sachin Taran, and Abdulkadir Şengür. Surface emg signals and deep transfer learning-based physical action classification. *Neural. Comput. Appl.*, 31(12):8455–8462, 2019. 32

[144] Jordan J Bird, Jhonatan Kobylarz, Diego R Faria, Anikó Ekárt, and Eduardo P Ribeiro. Cross-domain mlp and cnn transfer learning for biological signal processing: Eeg and emg. *IEEE Access*, 8:54789–54801, 2020. 32

[145] Xiang Chen, Yu Li, Ruochen Hu, Xu Zhang, and Xun Chen. Hand gesture recognition based on surface electromyography using convolutional neural

network with transfer learning method. *IEEE J. Biomed. Health. Inform.*, 25(4):1292–1304, 2020. 34, 35

[146] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Campeau-Lecours, Clément Gosselin, François Laviolette, and Benoit Gosselin. Transfer learning for semg hand gestures recognition using convolutional neural networks. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1663–1668. IEEE, 2017. 34

[147] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette, and Benoit Gosselin. Deep learning for electromyographic hand gesture signal classification using transfer learning. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 27(4):760–771, 2019. 34, 134, 138

[148] István Ketykó, Ferenc Kovács, and Krisztián Zsolt Varga. Domain adaptation for semg-based gesture recognition with recurrent neural networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2019. 34

[149] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17 (1):2096–2030, 2016. 34

[150] Ulysse Côté-Allard, Gabriel Gagnon-Turcotte, Angkoon Phinyomark, Kyrre Glette, Erik J Scheme, François Laviolette, and Benoit Gosselin. Unsupervised domain adversarial self-calibration for electromyography-based gesture recognition. *IEEE Access*, 8:177941–177955, 2020. 34

[151] Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. *arXiv preprint arXiv:1803.08533*, 2018. 38

[152] Wei Yang, Dapeng Yang, Yu Liu, and Hong Liu. Decoding simultaneous multi-dof wrist movements from raw emg signals using a convolutional

neural network. *IEEE Trans. Hum. Mach. Syst.*, 49(5):411–420, 2019. 39, 69, 89, 98, 113, 114

[153] Yusuke Yamanoi, Yosuke Ogiri, and Ryu Kato. Emg-based posture classification using a convolutional neural network for a myoelectric hand. *Biomed Signal Process Control*, 55:101574, 2020. 39

[154] Vimal Shanmuganathan, Harold Robinson Yesudhas, Mohammad S Khan, Manju Khari, and Amir H Gandomi. R-cnn and wavelet feature extraction for hand gesture recognition with emg signals. *Neural Comput. Appl.*, 32 (21):16723–16736, 2020. 39

[155] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(1):121–135, 2017. 39, 41

[156] Ce Zhang, Xin Pan, Huapeng Li, Andy Gardiner, Isabel Sargent, Jonathon Hare, and Peter M Atkinson. A hybrid mlp-cnn classifier for very fine resolution remotely sensed image classification. *ISPRS J. Photogramm. Remote Sens.*, 140:133–144, 2018. 39

[157] Xin Wang, Yujia Luo, Daniel Crankshaw, Alexey Tumanov, Fisher Yu, and Joseph E Gonzalez. Idk cascades: Fast deep learning by learning not to overthink. *arXiv preprint arXiv:1706.00885*, 2017. 39, 41, 48

[158] Sheng Wan, Tung-Yu Wu, Wing H Wong, and Chen-Yi Lee. Confnet: Predict with confidence. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2921–2925. IEEE, 2018. 39, 43, 44, 48, 53, 62

[159] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. A convolutional neural network cascade for face detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5325–5334, 2015. 41

[160] Yu Hu, Yongkang Wong, Wentao Wei, Yu Du, Mohan Kankanhalli, and Weidong Geng. A novel attention-based hybrid cnn-rnn architecture for semg-based gesture recognition. *Plos One*, 13(10):e0206049, 2018. 41

[161] Zied Tayeb, Nicolai Waniek, Juri Fedjaev, Nejla Ghaboosi, Leonard Rychly, Christian Widderich, Christoph Richter, Jonas Braun, Matteo Saveriano, Gordon Cheng, et al. Gumpy: A python toolbox suitable for hybrid brain–computer interfaces. *J. Neural Eng.*, 15(6):065003, 2018. 41

[162] Yue-Jiao Gong, Jing-Jing Li, Yicong Zhou, Yun Li, Henry Shu-Hung Chung, Yu-Hui Shi, and Jun Zhang. Genetic learning particle swarm optimization. *IEEE Trans. Cybern.*, 46(10):2277–2290, 2015. 45

[163] Enrique F Schisterman, Neil J Perkins, Aiyi Liu, and Howard Bondell. Optimal cut-point and its corresponding youden index to discriminate individuals using pooled blood samples. *Epidemiology*, pages 73–81, 2005. 47

[164] Thiago Mazzu-Nascimento, Giorgio Gianini Morbioli, Luis Aparecido Milan, Fabiana Cristina Donofrio, Carlos Alberto Mestriner, and Emanuel Carrilho. Development and statistical assessment of a paper-based immunoassay for detection of tumor markers. *Anal. Chim. Acta*, 950:156–161, 2017. 47

[165] Stefano Pizzolato, Luca Tagliapietra, Matteo Cognolato, Monica Reggiani, Henning Müller, and Manfredo Atzori. Comparison of six electromyography acquisition setups on hand movement classification tasks. *Plos One*, 12(10): e0186132, 2017. 49

[166] Francesca Palermo, Matteo Cognolato, Arjan Gijsberts, Henning Müller, Barbara Caputo, and Manfredo Atzori. Repeatability of grasp recognition for robotic hand prosthesis control based on semg data. In *2017 International Conference on Rehabilitation Robotics (ICORR)*, pages 1154–1159. IEEE, 2017. 49

[167] Adrian Burns, Barry R Greene, Michael J McGrath, Terrance J O'Shea, Benjamin Kuris, Steven M Ayer, Florin Stroiescu, and Victor Cionca. Shimmerâ„¢–a wireless sensor platform for noninvasive biomedical research. *IEEE Sens. J.*, 10(9):1527–1534, 2010. 49, 76

[168] Yuanfang Wan, Zishan Han, Jun Zhong, and Guohua Chen. Pattern recognition and bionic manipulator driving by surface electromyography signals using convolutional neural network. *Int. J. Adv. Robot. Syst.*, 15(5): 1729881418802138, 2018. 51

[169] Manfredo Atzori, Arjan Gijsberts, Claudio Castellini, Barbara Caputo, Anne-Gabrielle Mittaz Hager, Simone Elsig, Giorgio Giatsidis, Franco Bassetto, and Henning Müller. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Sci. Data*, 1:140053, 2014. 52, 56

[170] Alessio Gallina, Roberto Merletti, and Marco Gazzoni. Uneven spatial distribution of surface emg: what does it mean? *Eur. J. Appl. Physiol.*, 113(4):887–894, 2013. 53

[171] Adrain DC Chan and Geoffrey C Green. Myoelectric control development toolbox. *CMBES Proceedings*, 30, 2007. 61

[172] Chris Wilson Antuvan. *Decoding human motion intention using myoelectric signals for assistive technologies*. PhD thesis, Nanyang Technological University, 2019. 68

[173] Simone Benatti, Fabio Montagna, Victor Kartsch, Abbas Rahimi, Davide Rossi, and Luca Benini. Online learning and classification of emg-based gestures on a parallel ultra-low power platform using hyperdimensional computing. *IEEE Trans. Biomed. Circuits Syst.*, 13(3):516–528, 2019. 69

[174] Ahmed W Shehata, Erik J Scheme, and Jonathon W Sensinger. Evaluating internal model strength and performance of myoelectric prosthesis control

strategies. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 26(5):1046–1055, 2018. 69

[175] Wei Yang, Dapeng Yang, Jiaming Li, Yu Liu, and Hong Liu. Emg dataset augmentation approaches for improving the multi-dof wrist movement regression accuracy and robustness. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1268–1273. IEEE, 2018. 69, 113, 142

[176] Dwaipayan Biswas, Luke Everson, Muqing Liu, Madhuri Panwar, Bram Verhoef, Shrishail Patrika, Chris H Kim, Amit Acharyya, Chris Van Hoof, Mario Konijnenburg, et al. Cornet: Deep learning framework for ppg based heart rate estimation and biometric identification in ambulant environment. *IEEE Trans. Biomed. Circuits Syst.*, 2019. 70

[177] Peng Xia, Jie Hu, and Yinghong Peng. Emg-based estimation of limb movement using deep learning with recurrent convolutional neural networks. *Artif. Organs*, 42(5):E67–E77, 2018. 70, 113

[178] Dawei Huang and Badong Chen. Surface emg decoding for hand gestures based on spectrogram and cnn-lstm. In *2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)*, pages 123–126. IEEE, 2019. 70

[179] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Adv. Neural Inf. Process. Syst.*, pages 2483–2493, 2018. 72

[180] Kartik Chitturi and Peter Onyisi. How easily can neural networks learn relativity? In *Journal of Physics: Conference Series*, volume 1085, page 042020. IOP Publishing, 2018. 72

[181] Utkarsh Singh, Jean-François Determe, François Horlin, and Philippe De Doncker. Crowd forecasting based on wifi sensors and lstm neural networks. *IEEE Trans. Instrum. Meas.*, 2020. 72

[182] Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *Plos One*, 12(7):e0180944, 2017. 73

[183] Baoxuan Zhao, Changming Cheng, Zhike Peng, Xingjian Dong, and Guang Meng. Detecting the early damages in structures with nonlinear output frequency response functions and the cnn-lstm model. *IEEE Trans. Instrum. Meas.*, 2020. 74

[184] Zuxuan Wu, Xi Wang, Yu-Gang Jiang, Hao Ye, and Xiangyang Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 461–470. ACM, 2015. 74, 75

[185] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015. 74

[186] Ali Ameri, Ernest N Kamavuako, Erik J Scheme, Kevin B Englehart, and Philip A Parker. Real-time, simultaneous myoelectric control using visual target-based training paradigm. *Biomed Signal Process Control*, 13:8–14, 2014. 76

[187] Sebastian OH Madgwick, Andrew JL Harrison, and Ravi Vaidyanathan. Estimation of imu and marg orientation using a gradient descent algorithm. In *2011 IEEE international conference on rehabilitation robotics*, pages 1–7. IEEE, 2011. 76

[188] Mamun Bin Ibne Reaz, MS Hussain, and Faisal Mohd-Yasin. Techniques of emg signal analysis: detection, processing, classification and applications. *Biol. Proced. Online*, 8(1):11, 2006. 77

[189] Carlo J De Luca, L Donald Gilmore, Mikhail Kuznetsov, and Serge H Roy. Filtering the surface emg signal: Movement artifact and baseline noise contamination. *J. Biomech.*, 43(8):1573–1579, 2010. 77, 103

[190] Zhaoke Huang, Chunhua Yang, Xiaojun Zhou, and Tingwen Huang. A hybrid feature selection method based on binary state transition algorithm and relieff. *IEEE J. Biomed. Health Inform*, 2018. 77, 103

[191] Andrea d'Avella, Alessandro Portone, Laure Fernandez, and Francesco Lacquaniti. Control of fast-reaching movements by muscle synergy combinations. *J. Neurosci.*, 26(30):7791–7810, 2006. 78

[192] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *J Mach Learn Res*, 9(Nov):2579–2605, 2008. 79

[193] Bin Yu, Xu Zhang, Le Wu, Xiang Chen, and Xun Chen. A novel postprocessing method for robust myoelectric pattern-recognition control through movement pattern transition detection. *IEEE Trans. Human-Mach. Syst.*, 50(1):32–41, 2019. 79, 122

[194] Kazuhiko Seki and Mitsuo Narusawa. Firing rate modulation of human motor units in different muscles during isometric contraction with various forces. *Brain Res.*, 719(1-2):1–7, 1996. 89

[195] Wing-kin Tam, Tong Wu, Qi Zhao, Edward Keefer, and Zhi Yang. Human motor decoding from neural signals: a review. *BMC Biomed. Eng.*, 1(1): 22, 2019. 94, 96

[196] David J Warren, Spencer Kellis, Jacob G Nieveen, Suzanne M Wendelken, Henrique Dantas, Tyler S Davis, Douglas T Hutchinson, Richard A Normann, Gregory A Clark, and V John Mathews. Recording and decoding for neural prostheses. *Proceedings of the IEEE*, 104(2):374–391, 2016. 94, 96

[197] Elizaveta Okorokova, Mikhail Lebedev, Michael Linderman, and Alex Ossadtchi. A dynamical model improves reconstruction of handwriting from

multichannel electromyographic recordings. *Front. Neurosci.*, 9:389, 2015. 94

[198] Huseyin Coskun, Felix Achilles, Robert DiPietro, Nassir Navab, and Federico Tombari. Long short-term memory kalman filters: Recurrent neural estimators for pose regularization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5524–5532, 2017. 94, 98, 99

[199] Cheng Zhao, Li Sun, Zhi Yan, Gerhard Neumann, Tom Duckett, and Rustam Stolkin. Learning kalman network: A deep monocular visual odometry for on-road driving. *Rob Auton Syst*, 121:103234, 2019. 95, 99

[200] Ce Ju, Zheng Wang, Cheng Long, Xiaoyu Zhang, Gao Cong, and Dong Eui Chang. Interaction-aware kalman neural networks for trajectory prediction. *arXiv preprint arXiv:1902.10928*, 2019. 95

[201] Roman Kusche and Martin Ryschka. Combining bioimpedance and emg measurements for reliable muscle contraction detection. *IEEE Sens. J.*, 19 (23):11687–11696, 2019. 113

[202] Lingfeng Xu, Xiang Chen, Shuai Cao, Xu Zhang, and Xun Chen. Feasibility study of advanced neural networks applied to semg-based force estimation. *Sensors*, 18(10):3226, 2018. 114

[203] Dezhen Xiong, Daohui Zhang, Xingang Zhao, and Yiwen Zhao. Deep learning for emg-based human-machine interaction: A review. *IEEE/CAA J. Autom. Sin.*, 8(3):512–533, 2021. 114

[204] Chaohui Yu, Jindong Wang, Chang Liu, Tao Qin, Renjun Xu, Wenjie Feng, Yiqiang Chen, and Tie-Yan Liu. Learning to match distributions for domain adaptation. *arXiv preprint arXiv:2007.10791*, 2020. 114, 116

[205] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(4):801–814, 2018. 114, 119, 132

[206] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2935–2947, 2017. 114, 132

[207] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005. 115

[208] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 115, 116, 136

[209] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105. PMLR, 2015. 115, 136

[210] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Adv Neural Inf Process Syst*, pages 136–144, 2016. 115

[211] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017. 115, 116

[212] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016. 115

[213] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*, 2017. 115

[214] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In

Proceedings of the IEEE International Conference on Computer Vision, pages 5715–5725, 2017. 116, 117, 118, 120, 123, 124, 137

[215] Lukas Hedegaard, Omar Ali Sheikh-Omar, and Alexandros Iosifidis. Supervised domain adaptation using graph embedding. *arXiv preprint arXiv:2003.04063*, 2020. 117, 124

[216] Chao Chen, Zhihong Chen, Boyuan Jiang, and Xinyu Jin. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3296–3303, 2019. 118, 120

[217] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006. 119

[218] Bing Liu, Xuchu Yu, Pengqiang Zhang, Anzhu Yu, Qiongying Fu, and Xiangpo Wei. Supervised deep feature extraction for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.*, 56(4):1909–1921, 2017. 122

[219] Mark Halaki and Karen Ginn. Normalization of emg signals: to normalize or not to normalize and what to normalize to. *Computational Intelligence in Electromyography Analysis - A Perspective on Current Applications and Future Challenges*, pages 175–194, 2012. 123, 133

[220] Carlo J De Luca. Surface electromyography: Detection and recording. *DelSys Incorporated*, 10(2):1–10, 2002. 133

[221] Xinjun Sheng, Bo Lv, Weichao Guo, and Xiangyang Zhu. Common spatial-spectral analysis of emg signals for multiday and multiuser myoelectric interface. *Biomed Signal Process Control*, 53:101572, 2019. 134

[222] Fuzhen Zhuang, Xiaohu Cheng, Ping Luo, Sinno Jialin Pan, and Qing He. Supervised representation learning with double encoding-layer autoencoder

for transfer learning. *ACM Trans. Intell. Syst. Technol.*, 9(2):1–17, 2017. 137

[223] Jian Cheng, Jiaxiang Wu, Cong Leng, Yuhang Wang, and Qinghao Hu. Quantized cnn: A unified approach to accelerate and compress convolutional networks. *IEEE Trans. Neural Netw. Learn. Syst.*, 29(10):4730–4743, 2017. 138

[224] Boyu Fan, Xuefeng Liu, Xiang Su, Pan Hui, and Jianwei Niu. Emgauth: An emg-based smartphone unlocking system using siamese network. In *2020 IEEE International Conference on PerCom*, pages 1–10. IEEE, 2020. 142

[225] Akira Furui, Hideaki Hayashi, Go Nakamura, Takaaki Chin, and Toshio Tsuji. An artificial emg generation model based on signal-dependent noise and related application to motion classification. *Plos One*, 12(6):e0180112, 2017. 142

[226] Panagiotis Tsinganos, Bruno Cornelis, Jan Cornelis, Bart Jansen, and Athanassios Skodras. Data augmentation of surface electromyography for hand gesture recognition. *Sensors*, 20(17):4892, 2020. 142

[227] Qiqi Zhang and Ying Liu. Improving brain computer interface performance by data augmentation with conditional deep convolutional generative adversarial networks. *arXiv preprint arXiv:1806.07108*, 2018. 142

[228] Chao Duan, Steffen Junginger, Jiahao Huang, Kairong Jin, and Kerstin Thurow. Deep learning for visual slam in transportation robotics: a review. *Transp. Saf. Environ.*, 1(3):177–184, 2019. 142

[229] Michael Pfeiffer and Thomas Pfeil. Deep learning with spiking neurons: opportunities and challenges. *Front. Neurosci*, 12:774, 2018. 143

[230] J Camilo Vasquez Tieck, Sandro Weber, Terrence C Stewart, Arne Roennau, and Rüdiger Dillmann. Triggering robot hand reflexes with human emg data using spiking neurons. In *Int. Conf. Intell. Auton. Syst.*, pages 902–916. Springer, 2018. 143

[231] Jan Behrenbeck, Zied Tayeb, Cyrine Bhiri, Christoph Richter, Oliver Rhodes, Nikola Kasabov, Josafath I Espinosa-Ramos, Steve Furber, Gordon Cheng, and Jörg Conradt. Classification and regression of spatio-temporal signals using neucube and its realization on spinnaker neuromorphic hardware. *J. Neural Eng.*, 16(2):026014, 2019.

[232] Elisa Donati, Melika Payvand, Nicoletta Risi, Renate Krause, and Giacomo Indiveri. Discrimination of emg signals using a neuromorphic implementation of a spiking neural network. *IEEE Trans. Biomed. Circuits Syst.*, 13 (5):795–803, 2019. 143

[233] Enea Ceolini, Charlotte Frenkel, Sumit Bam Shrestha, Gemma Taverni, Lyes Khacef, Melika Payvand, and Elisa Donati. Hand-gesture recognition based on emg and event-based camera sensor fusion: A benchmark in neuromorphic computing. *Front. Neurosci*, 14, 2020. 143

[234] Levi Hargrove, Laura Miller, Kristi Turner, and Todd Kuiken. Control within a virtual environment is correlated to functional outcomes when using a physical prosthesis. *J. Neuroeng. Rehabil.*, 15(1):1–7, 2018. 143, 144

[235] Bernhard Terlaak, Hanneke Bouwsema, Corry K van der Sluis, and Raoul M Bongers. Virtual training of the myosignal. *Plos One*, 10(9):e0137161, 2015. 143

[236] Cosima Prahm, Ivan Vujaklija, Fares Kayali, Peter Purgathofer, and Oskar C Aszmann. Game-based rehabilitation for myoelectric prosthesis control. *JMIR Serious Games*, 5(1):e3, 2017. 143

[237] Hanneke Bouwsema, Corry K van der Sluis, and Raoul M Bongers. Effect of feedback during virtual training of grip force control with a myoelectric prosthesis. *Plos One*, 9(5):e98301, 2014. 143

[238] Richard B Woodward and Levi J Hargrove. Adapting myoelectric control

in real-time using a virtual environment. *J. Neuroeng. Rehabil.*, 16(1):1–12, 2019. 143

[239] Alexander Boschmann, Dorothee Neuhaus, Sarah Vogt, Christian Kaltschmidt, Marco Platzner, and Strahinja Dosen. Immersive augmented reality system for the training of pattern classification control with a myoelectric prosthesis. *J. Neuroeng. Rehabil.*, 18(1):1–15, 2021. 144