



MASTER THESIS

ESTIMATION OF LOWER LIMB JOINT KINEMATICS USING ELECTROMYOGRAPHY AND MACHINE LEARNING

Joost Herijgers

FACULTY OF ELECTRICAL ENGINEERING, MATHEMATICS AND COMPUTER SCIENCE
BIOMEDICAL SIGNALS AND SYSTEMS

EXAMINATION COMMITTEE
dr.ir. B.J.F. van Beijnum (Chair)
ir. R.V. Schulte
dr. E.C. Prinsen
M. Sartori, PhD

29-06-2020

Preface

Dear reader,

With this document, I conclude my Master's education in Biomedical Engineering. Since November 2019, I have been working on my thesis titled: *Estimation of lower limb joint kinematics using electromyography and machine learning*. This project proved to be a perfect opportunity for me to combine the topics I enjoyed most within the field of Biomedical Engineering: data science and signal analysis of human movements. Due to the COVID-19 situation, I had to make adaptations to the original research plan. Unfortunately, only three of the measurements we set up and designed could be carried out. However, I consider myself fortunate that I managed to think of a workaround in these strange times, so that the work in my thesis still aligned with the original ideas I had for the analysis.

This thesis was carried out at Roessingh Research and Development (RRD) within the MyLeg project. From the moment I started working at RRD, I have felt welcome and involved. Within the MyLeg team, I really enjoyed the involvement I personally had within the project. With all meetings we had, both in person as via Skype, it was both interesting and encouraging to discuss my work, as well as projects other members of the MyLeg team were working on. Therefore, I want to thank Erik, Robert, Eline, Parham, Rutger and Jelle for their contributions as part of the MyLeg team. Herein, special thanks go out to both Erik and Robert, for their great supervision during the past nine months. Special thanks also go out to Parham, as we collaborated frequently in setting up and carrying out the measurements.

Furthermore, I want to thank Bert-Jan and Massimo for their involvement and supervision within my thesis. We only spoke several times for the duration of my thesis, but I really appreciated the feedback I have received during the meetings we had. The feedback of both Bert-Jan and Massimo really helped me in challenging myself to improve and extend the analysis I have done.

Additionally, I want to thank my friends for showing interest in my thesis. Over the final months of my thesis, I have really appreciated all the support and involvement shown by my family, which helped me in making the most out of the thesis. A final word of thanks goes out to Lian, who was always there for me and enabled me to do the best I could have done.

I am proud of the work my thesis represents, so I want to wish you pleasant reading of the report!

Joost Herijgers

Samenvatting¹

In deze studie zijn verschillende algoritmen vergeleken die gebruikt kunnen worden voor het schatten van gewrichts kinematica van de onderste extremiteit. Deze schattingen zijn gemaakt door middel van elektromyografie (EMG) voor verschillende activiteiten. Het ultieme doel van deze methode is om mensen met een bovenbeenamputatie intuïtieve controle te geven over een actieve prothese in een niet-gewichtsdragende situatie. Doelen binnen deze studie waren het analyseren van de machine learning aanpak in (niet-)gewichtsdragende activiteiten, alsmede het bepalen van de performance wanneer de aanpak toegepast werd op mensen met een bovenbeenamputatie. Daarnaast is de invloed van het toevoegen van (historische) informatie van inertial measurement units (IMUs) onderzocht. Drie datasets zijn geanalyseerd om de verschillende onderzoeksdoelen te behalen. De geanalyseerde activiteiten waren niet-gewichtsdragende activiteiten, zitten-staan transities, lopen op een vlakke ondergrond (voor gezonde proefpersonen en proefpersonen met een bovenbeenamputatie), trap oplopen en trap aflopen. Acht verschillende machine learning algoritmen zijn gebruikt om te testen hoe geschikt ze zijn voor het schatten van gewrichts kinematica van het been. Per activiteit is er voor ieder algoritme een optimale combinatie van hyperparameters gevonden, gebruikmakende van een Bayesiaanse optimalisatie strategie. De performance van de verschillende algoritmen werd bepaald door middel van een 5-fold cross-validatie op een proefpersoon-specifiek niveau. Een convolutional neural network gaf de beste performance bij de meeste activiteiten ten aanzien van de R^2 en $RMSE$ maatstaven, wanneer enkel EMG gebruikt werd. Het toevoegen van historische IMU data zorgde voor een significante stijging van de performance ($p < 0.05$) bij de meeste activiteiten. Het gebruik van enkel de historische informatie van IMUs resulteerde in een significante afname van de performance in verschillende activiteiten. De ontwikkelde machine learning aanpak is toepasbaar op mensen met een bovenbeenamputatie, omdat er een vergelijkbare performance gevonden is voor gezonde proefpersonen en mensen met een bovenbeenamputatie bij lopen op een vlakke ondergrond. Resultaten voor de niet-gewichtsdragende activiteiten bij gezonde proefpersonen zijn veelbelovend (R^2 van 0.956 ± 0.13 voor de kniehoek). Daarom kunnen vervolgstudies zich richten op het bepalen van de toepasbaarheid van de methode op niet-gewichtsdragende activiteiten in mensen met een bovenbeenamputatie. Verdere vervolgstudies kunnen gericht worden op het gebruiken van verschillende methoden om features uit de EMG en IMU signalen te verkrijgen, om zo de performance van de algoritmen te verbeteren.

¹English summary is provided as an abstract on page 5

Contents

Abstract	5
1 Introduction	5
2 Methods	6
2.1 Data collection	6
2.1.1 MLK dataset	7
2.1.2 ENABL3S dataset	7
2.1.3 TIPS dataset	7
2.2 Data pre-processing	7
2.3 Machine learning approach	8
2.3.1 Algorithms	8
2.3.2 Hyperparameter optimisation	9
2.4 Performance metrics	10
2.5 Analysis strategy	10
2.5.1 EMG-only analysis	12
2.5.2 (EMG and) IMU analysis	12
2.5.3 Robustness analysis	12
2.6 Statistical analysis	12
3 Results	13
3.1 ENABL3S dataset	13
3.1.1 EMG-only input	13
3.1.2 EMG and IMU input	15
3.1.3 IMU input	15
3.1.4 Robustness analysis	16
3.2 MLK dataset	17
3.3 TIPS dataset	18
4 Discussion	19
4.1 Non-weight-bearing tasks & sit-to-stand transitions	19
4.1.1 Non-weight-bearing tasks	19
4.1.2 Sit-to-stand transitions	20
4.1.3 Sitting-generic models	21
4.2 ADLs	21
4.3 Influence IMUs	22
4.4 Transfemoral amputees	22
4.5 General findings	23
4.6 Recommendations	24
5 Conclusion	25
References	25
A Hyperparameter search	28
A.1 ENABL3S EMG-only	28
A.2 ENABL3S IMU	30
A.3 MLK	32
A.4 TIPS	35

Estimation of lower limb joint kinematics using electromyography and machine learning

J. Herijgers

Abstract

This study compared different algorithms that estimate lower limb joint kinematics from electromyography (EMG) in different activities. The ultimate goal is to give transfemoral amputees intuitive control over a powered prosthesis in a non-weight-bearing situation. Objectives of this study included an analysis of the machine learning approach in (non-) weight-bearing tasks and determining performance of the approach when applied to transfemoral amputees. Additionally, the influence of inclusion of (historic) information from inertial measurement units (IMUs) was studied. Three datasets were analysed to complete the different research objectives. The analysed activities included non-weight-bearing tasks, sit-to-stand transitions, level ground walking (for able-bodied subjects and transfemoral amputees), stair ascent and stair descent. Eight different algorithms were tested on their ability to estimate lower limb joint angles. An optimal set of hyperparameters for each algorithm was found using a Bayesian optimisation routine, on an activity-generic level. Performance of the different algorithms was analysed using a 5-fold cross-validation routine on a subject-specific level. A convolutional neural network gave the best performance in terms of R^2 and $RMSE$ using only EMG data in most tested activities. Including historic information from IMUs significantly increased performance ($p < 0.05$) for most of the studied activities. Exclusively using the same historic data from IMUs resulted in a significant decrease in performance for several of the studied activities. The developed approach showed to be feasible to apply to transfemoral amputees, as comparable performance is seen for amputees and able-bodied subjects in level ground walking. Results for non-weight-bearing tasks in able-bodied subjects were promising (R^2 of 0.956 ± 0.013 for the knee angle). Therefore, further research could focus on studying the applicability of non-weight-bearing tasks in transfemoral amputees. Additional future research could focus on using different methods to extract features from both the EMG and IMU signals to increase performance of the algorithms.

1 Introduction

In the Netherlands, 7.7 in 100,000 persons undergo a major amputation of the lower limb due to dysvascularity, which is the main cause for amputations [1]. An amputation of the lower limb drastically changes an individual's life and functional abilities, resulting in a lower quality of life [2]. To improve physical functioning, amputees can be fitted a prosthesis that makes it possible to regain mobility and independence during daily life [3]. Prostheses can be passive, semi-passive or active (also known as powered) devices [4], which are controlled differently. Passive prostheses are often used for both rehabilitation purposes and daily life settings. However, passive prostheses require the amputee to have adequate hip extensor strength and positional awareness of their knee flexion angle [5]. Furthermore, tasks like standing up or climbing stairs require generation of additional energy at the joints, which can not be provided by passive prostheses [4]. As a result, amputees show higher oxygen consumption compared to non-amputees, which can reduce the mobility of an amputee. To overcome the limitations of passive prosthetic devices, recent years have shown an increasing research interest into powered prosthetics for the lower limb [4].

The EU Horizon 2020 Research and Innovation Project MyLeg [6] aims to develop a new generation of powered transfemoral prosthetic legs. One of the main objectives of the MyLeg project is to use implantable myoelectric sensors on Targeted Reinnervated Mus-

cles [6]. Targeted Muscle Reinnervation (TMR) is a surgical technique that relocates nerves, that originally innervated the amputated limb, onto muscle sites that no longer have a biomechanical function due to the amputation [7]. Hence, this surgical procedure makes it possible to measure the intended motor commands of muscles that were lost due to the amputation. In the case of transfemoral amputations, measuring muscle activity of for instance the m. gastrocnemius could still be possible with TMR.

This thesis relates to the MyLeg objective to develop a high-level, intuitive, control scheme for the prosthesis user. The aim of the high-level control is to detect user intention to perform different activities. A state-machine (mid-level controller) can then be used to select appropriate low-level control of the prosthesis in different activities of daily living (ADLs). Previous work in the MyLeg project includes the development of classifiers that use electromyography (EMG) and Inertial Measurement Units (IMUs) to predict activities that are performed by the user. Previous research has shown that classification errors are reduced when a combination of EMG and IMU data is used as input [8].

Prediction of discrete activities can give an intuitive control scheme by transitioning between states of the state machine, but it limits the possibility to apply voluntary proportional control to the prosthesis [9]. Especially in the sitting state, it can be beneficial for an amputee to apply direct voluntary control to intuitively reposition the prosthesis instead of relying

solely on the mid-level control of the sitting state. Voluntary control of the prosthesis can be realised by presenting information on required joint kinematics corresponding to intended movements. If joint kinematics are to be used as a control signal, an estimation of the required kinematics has to be done prior to execution of the movement. This requires detection of movement onset, which is possible by measuring muscle activity using EMG, up to 138 ms in advance when the prosthesis leads [10]. As the MyLeg project aims to employ myoelectric sensors in the prosthesis, EMG signals of the lower limb musculature can be used as a predictor for the lower limb joint kinematics.

There are several data-driven approaches that estimate joint angles in the lower extremity from EMG. Earlier work on this topic used Neural Networks (NNs) [11, 12, 13] and Support Vector Regression (SVR) [13] to estimate joint angles in the lower extremity. However, literature on the topic of joint kinematics estimation from EMG is mainly focussed on the upper extremity, where elbow, wrist and finger kinematics are estimated [14, 15, 16]. Although upper extremity kinematics are estimated in these studies, the techniques employed could also be feasible for estimation of knee and ankle kinematics. As with the lower extremity, most studies have used neural networks or SVR to estimate the desired angles [14, 15, 17]. However, different approaches are also seen in the upper extremity: Gaussian Process regressors (GPs) [16, 18], Kernel Ridge Regression (KRR) [19], Linear Regression (LR) [19], Convolutional NNs (CNNs) [20], or Long Short Term Memory (LSTM) layers in conjunction with a CNN (CNN-LSTM) [21].

The different algorithms have shown their applicability to estimate joint angles from EMG signals. Direct comparison, on the same data, of all methods can demonstrate which algorithm is best suited for application within the MyLeg project. Therefore, the main research question of this study is: *What is the best method for estimation of lower limb joint kinematics based on electromyography for online control of a transfemoral prosthesis?*

The first objective of this thesis is to find the best algorithm for the non-weight-bearing tasks and the sit-to-stand transitions (and stand-to-sit transitions). The intended use of the algorithms is to give an amputee the ability to voluntary control his/her prosthesis in the sitting state as selected by the mid-level controller. As such, a primary interest is taken in performance for non-weight-bearing situations. Furthermore, it is helpful for the amputee if assistance is provided by the prosthesis in standing up from a sitting position. Earlier studies have focussed on assessing knee flexion and extension tasks under different movement speeds and loads [11, 13]. This thesis assesses whether different types of non-weight-bearing movements can be estimated accurately by a single model. No previous work has been found that estimates joint angles from EMG during sit-to-stand transitions.

A second objective is to assess how accurate estimations of the required kinematics are for different

weight-bearing ADLs. The studied ADLs are level ground walking, stair ascent and stair descent. The ADLs are studied, as amputees could prefer voluntary control of their prosthesis over the selected low level control of the state machine, if the estimation errors are not too large. Of the studied ADLs, estimation of joint angles has previously only been done for level ground walking [12].

The third objective is to assess the influence of historic information from IMUs on the predictions, so that information of the prosthesis' movement is available. Hu et al. [8] showed that a combination of EMG and IMUs reduce the activity classification error. Possibly, a combination of EMG and historic IMU data can reduce the joint angles' estimation error. This reduction is expected due to having more information available on the prosthesis' movements in, for example, a sit-to-stand transition. However, no previous work has studied the combination of using both types of sensing modalities for the prediction of lower limb joint kinematics.

The fourth and final objective is to determine the performance of the algorithms on transfemoral amputees. Assessment of performance in transfemoral amputees is an important analysis to do since this analysis gives an indication of the performance of the algorithms when applied to the expected users. Currently, it remains unanswered whether the prediction of joint angles is feasible based on EMG acquired from muscles in the stump of a transfemoral amputee.

This thesis is structured as follows. In section 2, the experimental procedure and data analysis are presented. Section 3 presents the results obtained from the conducted experiments. Section 4 discusses these results and provides an interpretation on the relevance of the results. Recommendations for further research are also discussed in this section. Section 5 concludes this paper.

2 Methods

The methods section is divided in six parts: data collection, data pre-processing, machine learning approach, performance metrics, analysis strategy and statistical analysis.

2.1 Data collection

Three different datasets were used to complete the research objectives: The MyLeg Kinetics (MLK) dataset that was collected as part of this study, the online available ENcyclopedia of Able-bodied Bilateral Lower Limb Locomotor Signals (ENABL3S) dataset [22] and the The Ideal Prosthesis Selection (TIPS) dataset, which was collected (but remained unpublished) earlier at Roessingh Research and Development (RRD). This section presents the different datasets, along with a brief description of what analyses were done with the respective datasets.

2.1.1 MLK dataset

Performance for non-weight-bearing tasks and sit-to-stand (and stand-to-sit) transitions was assessed with the MLK dataset. Three able-bodied subjects (sex: one male, two female, age: 21.3 ± 1.5 years, length: 174.7 ± 10.0 cm, weight: 69.6 ± 8.6 kg) were recruited who performed several activities. Prior to the measurements all subjects gave their informed consent. The activities were split into four categories: Non-weight-bearing tasks, sit-to-stand transitions, level ground walking and stair ascent/descent. The non-weight-bearing tasks consisted of knee flexion and extension separately, or in combination with ankle plantar- and dorsiflexion. This gave three combinations: 1. Knee flexion and extension, 2. Extended knee with ankle plantar- and dorsiflexion, 3. Flexed knee with ankle plantar- and dorsiflexion. In sit-to-stand tasks, subjects were asked to stand up from a stool, stay in position, and sit down again. In the stairs task, a subject had to ascent two steps of 15 cm in height. This task also included the subject descending these same steps. All above-mentioned tasks were repeated thirty times. Level ground walking was performed fifty times, where a subject walked approximately four metres in one repetition.

Bipolar EMG was recorded from eight different muscles on the right leg: rectus femoris (RF), vastus lateralis (VL), biceps femoris (BF), gluteus maximus (Gmax), gastrocnemius medialis (GM), tibialis anterior (TA), semitendinosus (ST) and adductor magnus (AM). All EMG electrodes were placed according to SENIAM guidelines [23]. The signals were acquired using the Sessantaquattro (OT Bioelettronica, Turin, Italy) at a sample frequency of 2000 Hz.

Joint angles of the knee and ankle were determined using eight IMUs (Xsens Link, Enschede, The Netherlands), which were placed on the sternum, the pelvis and bilaterally on the thigh, shank and foot of the subject. The IMUs had a sample frequency of 240 Hz. Joint angles were reconstructed from Xsens MVN software. Only joint angles in the sagittal plane were used in this study. The IMUs output 3D accelerations and 3D angular velocities.

2.1.2 ENABL3S dataset

Performance for ADLs and the influence of IMUs were assessed using the publicly-available ENABL3S dataset [22]. The ENABL3S dataset contained simultaneously collected kinematics and EMG data of the lower limb, for ten able-bodied subjects (sex: seven male, three female, age: 25.5 ± 2 years, length 174 ± 12 cm, weight: 70 ± 14 kg) [22]. EMG of seven muscles were recorded (TA, GM, RF, VL, BF, ST and soleus (SOL)). Knee and ankle kinematics were recorded using goniometers (sagittal plane only). IMUs were placed bilaterally on the thigh and shank. The ENABL3S database contained sit-to-stand transitions (± 50 repetitions per subject), level ground walking (± 100 repetitions per subject), stair ascent (± 25

repetitions per subject) and stair descent (± 25 repetitions per subject), which correspond to the chosen ADLs of research objective 2. Details of the data collection procedure for the ENABL3S dataset can be found in the publication [22].

2.1.3 TIPS dataset

Performance in transfemoral amputees was assessed using the TIPS dataset, which was collected at RRD. The TIPS dataset contained level ground walking of six transfemoral amputees (age: 49.2 ± 3.8 years, length: 183.5 ± 8.3 cm, weight: 80.7 ± 10.6 kg, time since amputation: 9.8 ± 13.2 years), who were fitted five passive prosthetic knees. The prosthetic knees varied in complexity and are listed along increasing complexity: Ottobock 3R20, Ottobock 3R106, Ottobock 3R60, Össur Mauch SNS and a micro-processor-controlled Adaptive Knee (either Össur Rheo Knee or Ottobock's C-Leg). The simplest knee had no varying resistance and the most complex knee had varying resistance and adaptive control in both the stance and swing phase. The subjects completed ± 30 trials of level ground walking using each prosthetic knee (approximately four meters per trial). Bipolar EMG was collected using the Glonner BioTel EMG system at a sample frequency of 1000 Hz. Five muscles were measured on the amputated side: RF, VL, Gmax, ST and Tensor Fascia Latae (TFL). Kinematics were obtained using a six-camera motion capture system (Vicon Nexus, Oxford, Great-Britain) with sixteen reflective markers, using Vicon Nexus' Plug-in-Gait model.

No dataset was available for transfemoral amputees in a non-weight-bearing situation. Therefore, this dataset (containing level ground walking only) was chosen to give an indication of the performance of the developed approach for prosthesis users.

2.2 Data pre-processing

The ENABL3S dataset and TIPS dataset were time-synchronised and had been resampled to a sample frequency of 1000 Hz. The EMG system and IMU system used in the MLK dataset had to be synchronised. Synchronisation was done using the method developed by Schulte et al. [24], using an additional accelerometer connected to the EMG system. All signals were resampled to a sample frequency of 1000 Hz.

The smoothed rectified envelope (SRE) of the EMG was used as the only input feature to reduce model complexity. The SRE was obtained from the raw EMG by combining three pre-processing steps [11, 12, 16]. First the EMG was high-pass filtered with a second order zero-lag Butterworth filter with a cut-off frequency of 20 Hz. Secondly, the signal was rectified. Finally, the signal was low-pass filtered with a second order zero-lag Butterworth filter with a cut-off frequency of 6 Hz. Both the SREs and the joint angles were windowed to prepare the input data for the algorithms. Data windows of 128 ms with 50% overlap were created. This window size was chosen to accommodate

for the detection of the movement onset using EMG [10], in combination with having a computationally-efficient size of the data windows. The SRE values in the window were retained to generate an image as input for the CNN- and LSTM-based neural networks (see section 2.3.1). Hence, one input image had size $(128, num_EMG)$, with num_EMG 8, 7 and 5 for the MLK dataset, ENABL3S dataset and TIPS dataset respectively. In all other algorithms the input was the mean value of each muscle’s SRE in the given window (size $(num_EMG, 1)$), to compare the model performances with respect to the presented data pre-processing strategy. The target data were the mean joint angles in the analysis window.

A final pre-processing step was scaling the data. Machine learning models tend to perform better on scaled data, as each input feature has equal weight when all features are scaled to fall within a specified range [25]. The data was scaled to the maximal absolute value in the data (per channel), so that all data points fell within the range $(-1, 1)$. Data was standardized for the GPs (see section 2.3.1) as a zero mean of the data was assumed for this type of model [26].

Historic information from the IMUs was used in models where IMU data was included. Accelerations and angular velocities measured one non-overlapping data window back in time (128 ms) were used. Norm vectors were computed and used as input for both the accelerations and angular velocities. The norm vectors were chosen, because general movement of a body segment (such as the thigh) was the parameter of interest. Furthermore, the use of the norm vectors ensured that the input still primarily consisted of EMG signals. The use of the norm vector as IMU feature, reduced the model complexity as less model parameters needed to be learned. The norm vector of an acceleration signal was computed as in equation 1.

$$a_{norm} = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (1)$$

In figure 1, the different pre-processing steps can be seen, next to the other processing steps.

2.3 Machine learning approach

This section describes the different utilised algorithms, as well as the procedure to find the optimal hyperparameter settings. All algorithms were implemented in the open-source software Python v3.7 [27]. The selected algorithms were chosen based on their earlier application for joint angle estimation problems, as described in the introduction.

2.3.1 Algorithms

Three classes of algorithms were used in this study: neural networks, kernel-based algorithms and a linear method. The neural networks estimated all joint angles simultaneously, for the kernel-based methods separate models were created for each joint. This approach

was taken based on the results of a preliminary analysis on a different dataset, which was comparable to the ENABL3S dataset.

Neural Networks

The first class consisted of several types of neural networks. Neural networks are models that consist of multiple layers which contain neurons. The first type was the fully-connected neural network (FC-NN). The FC-NN is the simplest form of a neural network. In a FC-NN, the weights are learned that connect all neurons in one layer with all neurons in the next layer. Each neuron has its own activation function, which is applied to the sum of all incoming neurons, multiplied by their weights, including a bias term. The weights and biases are found by minimising an error function, which can be done using several optimisation techniques (optimisers) such as Stochastic Gradient Descent. Due to the model structure, a neural network is able to approximate any given function [26].

The second type of neural networks were CNNs. Compared with regular neural networks, CNNs try to learn features from input data, which makes the network more robust to distortions in the input. It does so by making use of local receptive fields and weight sharing. From the convolutional layer(s), a feature map is obtained. To further reduce the model complexity, a sub-sampling layer can be used that combines the information of several, pre-defined, samples in the feature map [26].

A third type of neural networks included LSTM layers. An LSTM layer is a type of recurrent neural network, that tries to learn from the temporal sequence in which data is processed. The model can learn from sequential data by recursively looping over the previous states. Recursively looping over previous states is done by using several gates: the input gate, the forget gate and the output gate. These gates control what information is retained and which information can be dropped from the model [21].

The activation function for all fully-connected (FC) layers was the Rectified Linear Unit (ReLU) activation function, except for the output layer, which used a linear activation function. Furthermore, a dropout layer was used which decreases the probability of overfitting, by randomly detaching a fraction of the connections during model training [28]. The dropout layer was used for the FC layers.

1D convolutional layers were used in this study and a MaxPooling layer was chosen as sub-sampling layer. The size of the local receptive field followed from the chosen kernel size. ReLU activation functions were used in all convolutional layers. Several FC layers were connected to the output of the final convolutional layer, so that the feature map was essentially used as input to an FC-NN to estimate the joint angles.

The LSTM layer was implemented in two settings: firstly as input layer, with FC layers connected to the LSTM layer (referred to as LSTM) and secondly as an intermediate layer between convolutional layers

and FC layers (CNN-LSTM). The recurrent activation function in the LSTM layer was the sigmoid function. The activation function was the LeakyReLU function, which is a ReLU function with a small slope (0.1 is the chosen slope) for negative inputs. The LeakyReLU function helps in preventing finding a zero gradient for negative inputs [29]. A preliminary analysis showed this type of activation function was necessary for the LSTM models.

All neural networks were implemented using Python’s software package Keras (v2.3.1), with a TensorFlow backend (v2.0.0) [30]. Several settings for compilation were chosen: the type of optimiser, the batch size and the number of epochs. Other compilation settings used their respective default values. The optimiser is an algorithm that helps finding the model parameters (weights and biases) in the neural network [26]. The batch size controls the number of training samples that are passed before the model parameters are updated. The number of epochs sets the repetitions that the training data is passed through the network, so that the weights can be learned.

Kernel-based algorithms

The kernel-based algorithms comprised of Support Vector Regression (SVR), Kernel Ridge Regression (KRR) and Gaussian Processes (GPs). In SVR, data points are transformed to a high-dimensional space using (non-linear) kernels. To optimise the use of SVR, an ϵ -insensitive error function is employed, where $\epsilon > 0$. All errors smaller than ϵ are treated as zero error. This gives a range of values which are treated as zero error by the model, which is known as the tube. In the model, all training data points that lie outside (or on) the tube are treated as support vectors. Using support vectors, estimations can be made of the data points in the test set [26].

Another technique that is somewhat similar to SVR is KRR [19]. This technique is similar in the sense that the data is mapped into a higher dimensional space using a kernel, so that a linear solution can be found in the transformed feature space. Compared with SVR, a different cost function is used, which is applied in the learning phase of the model [31]. Another difference is that KRR gives a non-sparse model, whilst SVR creates a sparse model. Due to this characteristic, a KRR model is slower in making predictions than an SVR model.

A final kernel-based approach was GPs for regression [16, 18]. In GP regression, the target value (joint angles) is found by finding the probability distribution of functions that maps the stochastic process (the target values) in a consistent manner [26]. The probability distribution consists of a mean function and a covariance function. As the data used for the GPs was standardised, it was ensured that the data has a zero mean. Therefore, only the covariance function needed to be constructed, which can be found directly by employing different kernels. A GP regressor can become infeasible for large datasets as too many datapoints

have to be used to learn the optimal covariance function, as it is a non-sparse model [26].

All kernel-based algorithms were implemented using Python’s scikit-learn library (v0.21.3) [32].

Linear regression

Performance of a linear method was also studied in this model comparison, using a Linear Regression (LR) model [19]. In LR models, a mapping is learned from the input data to the output data based on a weight vector that minimizes the sum of squared errors between the observed and predicted outputs. No hyperparameters were optimised for this model. Implementation was done using Python’s scikit-learn library [32].

2.3.2 Hyperparameter optimisation

It is a necessary step to optimise the hyperparameters in a machine learning model in order to get the best performance out of each algorithm. The optimal hyperparameters differ per dataset, as different relations between input and output can be learned from different datasets. The way a model learns these relations depends on the chosen model settings (i.e. hyperparameters). In this study, hyperparameter optimisation was done by using a Bayesian optimisation strategy [33]. In a Bayesian optimisation strategy, a probability distribution is learned over the set of hyperparameters along with the performance on the validation set. Therefore, more time could be spent searching hyperparameter spaces that had a higher probability of returning good model performance. This decreased the computational cost of running the optimisation routines and increased the probability of finding a good set of hyperparameters for the given model. Per model, ranges of the possible hyperparameters were used as input to the search algorithm.

In this study, the optimal hyperparameters for each model were found per activity (in a given dataset), based on 5-fold cross-validation (5-cv) on all available data for that activity. All subject data of an activity was stacked for neural networks to create the most-generalisable hyperparameter settings. It was ensured that the splits for the 5-cv were made so that both the training set and test set contained data from each subject in the dataset. 75 evaluations were performed on the hyperparameter space per optimisation. Activity-generic hyperparameter settings were found for each individual algorithm described in section 2.3.1. The kernel-based approaches required too much memory when all data was stacked together. Therefore, a slightly different approach was taken as a 5-cv was performed for each subject separately. The results for all subjects were then averaged in the loss function. The minimised loss function by the optimisation routine was $1 - R^2$, with R^2 one of the performance metrics described in section 2.4.

The type of optimiser used for the neural networks was considered a hyperparameter that required optimisation. All optimisers were used with their default settings. The number of epochs and batch size were not optimised. Three epochs were chosen for the hyperparameter optimisation procedure to limit computation time of the optimisations. Testing neural network model performance was done with fifteen epochs, so that the neural networks were allowed to minimise the error function further. The maximum number of epochs was chosen to prevent overfitting to the training data. The chosen error function to be minimised during training was the mean squared error function. The chosen batch size was 64.

Optimised hyperparameters of the FC-NN were the number of layers, number of neurons in a layer and the dropout ratio. Optimised hyperparameters of the CNN were the kernel size, number of neurons in a convolutional layer, number of convolutional layers and pool size in the MaxPooling layer. Furthermore, the fully-connected layers in the CNN had the same hyperparameters that required optimisation as in the FC-NN. Optimised hyperparameters of the neural networks including an LSTM layer were similar to the FC-NN and CNN, with one additional hyperparameter which was the number of neurons in the LSTM layer.

Hyperparameters of the kernel-based methods followed from the arguments that could be given to each model’s function in Python’s scikit-learn library.

A list of optimised hyperparameters per algorithm is presented in table 1. The optimisation routine was implemented by using Python’s software package hyperopt (v0.2.2) [34], in combination with hyperas (v0.4.1). A complete list of optimised hyperparameters per dataset is presented in Appendix A.

2.4 Performance metrics

To compare the performance of the algorithms, two different, frequently-used, performance metrics were chosen [11, 12, 13, 14, 19]. The chosen performance metrics were the coefficient of determination (R^2) and the Root Mean Square Error ($RMSE$). The R^2 gives an indication on the goodness-of-fit of the prediction, whilst the $RMSE$ gives an indication on the prediction error. Equations 2 and 3 present the mathematical notation for both performance metrics. \hat{x} denotes the predicted data point. \bar{x} is the mean value for all data points.

$$R^2 = 1 - \frac{\sum_{t=0}^N (\hat{x}_t - x_t)^2}{\sum_{t=0}^N (x_t - \bar{x})^2} \quad (2)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=0}^N (\hat{x}_t - x_t)^2} \quad (3)$$

Table 1: Optimised hyperparameters per algorithm.

Algorithm	Optimised hyperparameters
FC-NN	Number of hidden layers Number of neurons in hidden layer Dropout ratio Optimiser
CNN	Number of convolutional layers Number of neurons in convolutional layer Kernel size of the convolutional layer Pool size of the MaxPooling layer Number of hidden FC layers Number of neurons in FC layer Dropout ratio Optimiser
LSTM	Number of neurons in LSTM layer Number of hidden FC layers Number of neurons in FC layer Dropout ratio Optimiser
CNN-LSTM	Number of convolutional layers Number of neurons in convolutional layer Kernel size of the convolutional layer Pool size of the MaxPooling layer Number of neurons in LSTM layer Number of hidden FC layers Number of neurons in FC layer Dropout ratio Optimiser
SVR	Kernel type Regularisation parameter ('C') ϵ value for the ϵ -tube Independent term kernel function ('coef0') Degree of the polynomial kernel
KRR	Kernel type Conditioning parameter (α) Independent term kernel function ('coef0') Degree of the polynomial kernel
GP	Kernel type
LR	-

2.5 Analysis strategy

A graphic overview of the processing pipeline (for a single type of algorithm and activity) is presented in figure 1. Subject-specific models were created per activity, because implementation for amputees will be amputee-specific. The model’s hyperparameter settings were found from the activity-generic hyperparameter optimisation results. A 5-cv was done to determine performance for each subject-specific model per activity. 5-cv was chosen as the number of repetitions for all activities in the different datasets were limited. Due to the training of five different models

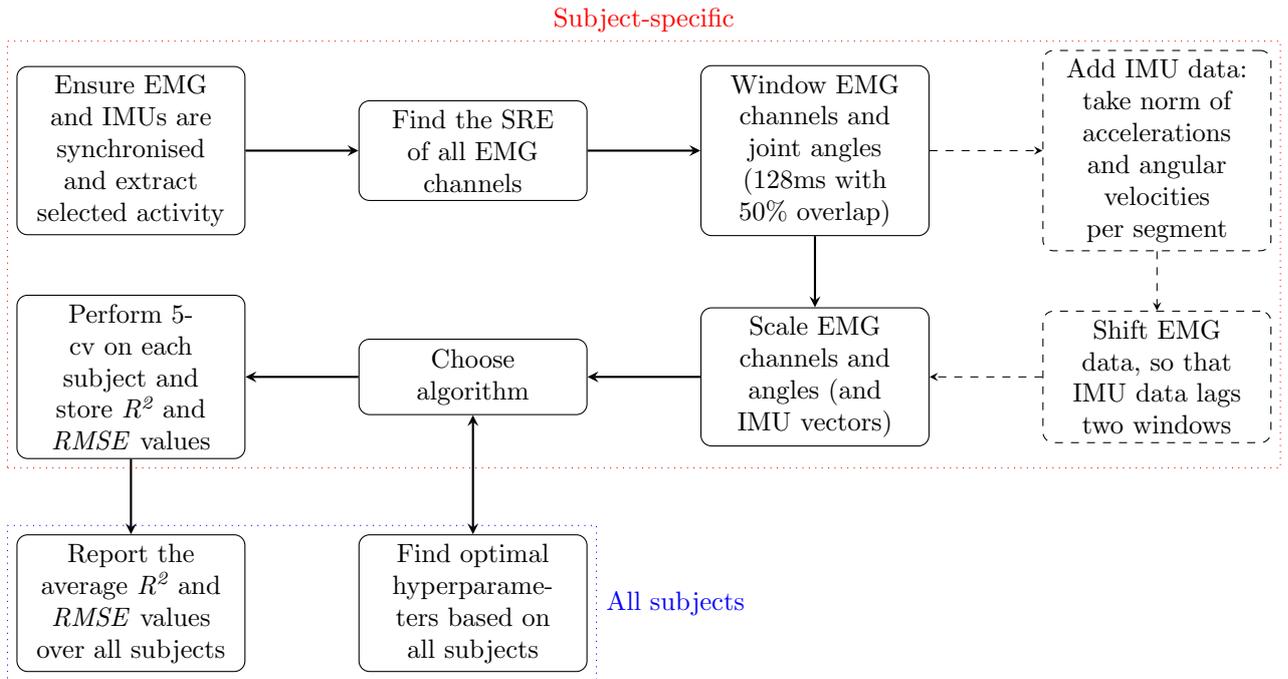


Figure 1: Graphical overview of the processing pipeline for one type of algorithm and one kind of activity. Dashed lines are optional steps in the pre-processing, depending on what data is included in the models.

(on different training sets), a measure on the generalisability per subject could be found. k-fold cross-validation is an approach commonly seen for reporting subject-specific performance, with k set to four or five [13, 15, 16, 17, 19]. Models were created per activity, as different states per activity could be selected by the mid-level controller. Per subject, the performance was determined using the mean and standard deviation of the performance metrics over the five folds. General algorithm performance was determined using the mean and standard deviation of the performance metrics over all subjects for each activity.

The three described datasets (in section 2.1) were used to study the different objectives. The MLK dataset was used to study the first objective, which was to determine performance in non-weight-bearing tasks and sit-to-stand transitions. Sit-to-stand transitions were also studied using the ENABL3S dataset.

Performance in different ADLs, which was the second objective, was determined using the ENABL3S dataset. The third objective, studying the influence of historic IMU data, was also studied with the ENABL3S dataset as the ENABL3S dataset was more extensive than the MLK dataset. The fourth objective was to study performance of the algorithms when used on transfemoral amputees, which could be studied with the TIPS dataset. A final analysis was conducted on the ENABL3S dataset to determine robustness of the different algorithms. A general overview of the different analyses is summarised in figure 2.

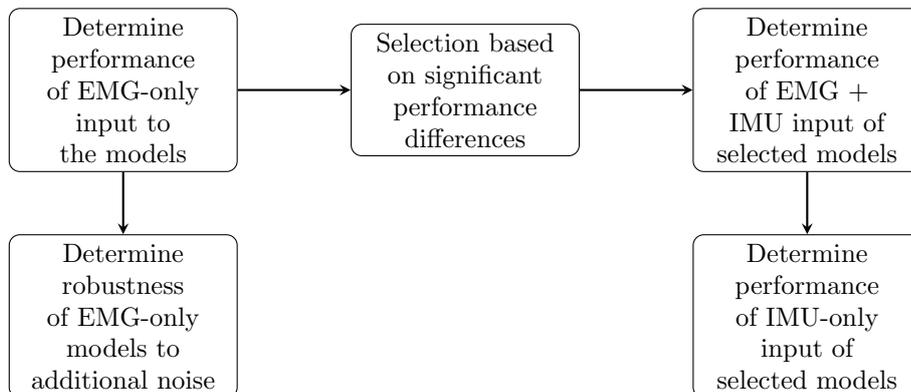


Figure 2: Graphical overview of the data analysis pipeline for the ENABL3S dataset. Analysis of the MLK dataset and TIPS dataset was done exclusively for EMG-only input.

2.5.1 EMG-only analysis

All algorithms described in section 2.3.1 were tested to assess the performance on the different datasets. The ENABL3S dataset made it possible to analyse four activities: sit-to-stand transitions, level ground walking, stair ascent and stair descent. The two analysed activities in the MLK dataset were the combined non-weight-bearing activities (as described in section 2.1.1) and sit-to-stand transitions (sitting-specific models). Furthermore, models were created that were trained on data from both non-weight-bearing tasks and sit-to-stand transitions (sitting-generic models). This approach was taken, as a preliminary analysis revealed that training a model on one of both activities, and testing on the other activity, gave poor performance (negative R^2 , relatively large $RMSE$) for both joints. The analysis of combined activities was done to determine whether a generic model could be trained for providing intuitive control in the sitting state. The TIPS dataset was used to study level ground walking of transfemoral amputees. As different prostheses were used in the trials, different knee angles were obtained per prosthesis. Therefore, for each subject all algorithms were trained for each type of prosthetic knee. Joint angles in the sagittal plane were estimated. Ankle and knee joint angles were estimated for the ENABL3S dataset and MLK dataset. Knee joint angles were estimated for the TIPS dataset.

2.5.2 (EMG and) IMU analysis

A selection was made for the algorithms to be used for the IMU analysis. The selection was made to reduce the computational load, by dropping the worst-performing models on the EMG-only input for the analysis on different types of input. The selection procedure is described in section 2.6. The selected algorithms were tested using historic IMU information in conjunction with the EMG data. Additionally, models containing merely historic IMU data were tested, to be able to objectively study the influence of historic IMU data.

2.5.3 Robustness analysis

The ENABL3S dataset was used to assess the robustness of the various algorithms to noise. The robustness analysis was done to assess whether performance of the different models changed when testing the models on different circumstances than that were used for training the models. The sit-to-stand activity was selected for the robustness analysis, as this activity belonged to the main application for the MyLeg project. Testing robustness of EMG-based controllers to various sources of noise was previously done in several arm prosthetics' control tasks [35, 36, 37]. The reasoning for addition of noise was that the EMG signal could differ due to small placement differences of the prosthesis after donning and doffing, a change in skin condition or the time of testing (on for instance a different day) [36]. As testing on different days was not

possible, the change in signal was simulated by contaminating all raw EMG signals with white Gaussian noise. This condition simulated an unstable reference electrode [37]. To perform a sit-to-stand transition, the studied muscles require muscle activity that was 21.77% to 37.41% of the Maximum Voluntary Contraction (MVC) [38]. Four different noise levels were tested, namely 2%, 4%, 6% and 8% of the Root Mean Square (RMS) value of the MVC per muscle. These noise levels were chosen to ensure that the noise corresponds to at most 10%, 20%, 30% and 40% of the MVC percentage required for the sit-to-stand transition. The 10%, 20%, 30% and 40% levels were studied in a prosthetics' control task [37]. The models were trained on uncorrupted data and tested on the four noise levels, using a 5-cv. Pre-processing was done in the same manner as for the original data. Models are considered robust to an additional noise level when no significant performance difference is found.

2.6 Statistical analysis

The selection procedure for the ENABL3S dataset was based on a statistical analysis of the model goodness-of-fit performance and was carried out for each individual activity. The R^2 was chosen as selection criterion as this metric was used to find the optimal hyperparameters. All algorithms are compared with the best-performing algorithm (per joint), which was determined by taking the overall average R^2 value over all subjects.

The selection procedure determined which algorithms did not significantly differ ($p > 0.05$) from the best-performing algorithm. The ENABL3S dataset contained data from ten subjects, so ten R^2 values were used for the algorithm comparison. The ten R^2 values were the averaged results over the different folds in the 5-cv per subject. Eight different algorithms were tested, thus seven statistical tests were done. A Wilcoxon signed-rank test was used to compare the best-performing algorithm with every other algorithm. As seven statistical tests were done, a post-hoc analysis was applied. The Holm-Bonferroni correction was used for the post-hoc analysis [39]. All algorithms that had no significant performance difference, compared to the best-performing model, were selected for (EMG and) IMU analysis. If only a single joint showed a not significant performance difference, the algorithm was selected for further analysis for both joints.

Statistical comparison of the best-performing EMG-only model (per activity and metric) with the best-performing EMG and IMU input and the IMU-only input models was also done using the Wilcoxon signed-rank test. Performances were deemed significantly different at the level $p < 0.05$. Statistical analysis of the robustness analysis compared the performance at a given noise level with the baseline conditions (0% MVC). As four statistical tests were done, a post-hoc analysis was applied using the Holm-Bonferroni correction to assess which conditions showed significant different performance. The MLK

dataset had insufficient subjects (three) to perform a statistical analysis. Statistical analysis of the TIPS dataset was done using the Wilcoxon signed-rank test combined with the Holm-Bonferroni correction, to compare the different model performances.

Statistical analysis was performed using Python’s software package SciPy (v1.4.1) [40].

3 Results

In this section, the results are presented. Each dataset has its own subsection.

3.1 ENABL3S dataset

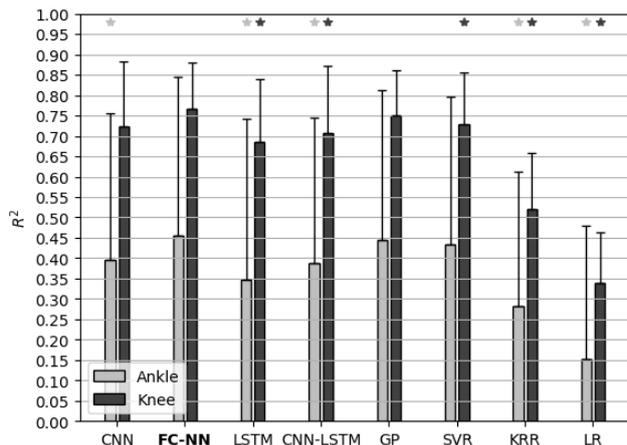
Results for the different analyses (figure 2) of the ENABL3S dataset are presented in individual subsections.

3.1.1 EMG-only input

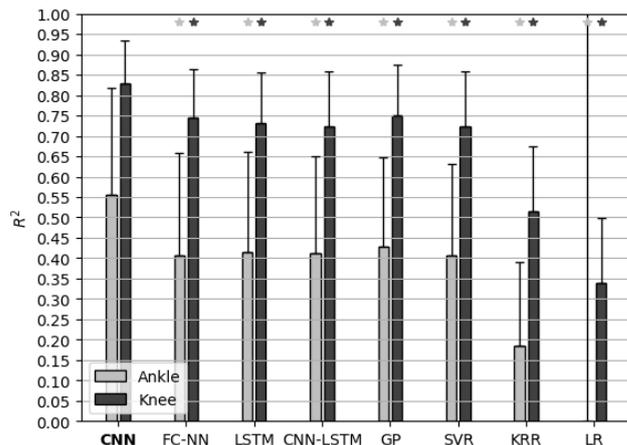
Figures 3a, 3b, 3c, 3d show the results of the R^2 for the different models in the EMG-only situation. The

FC-NN has the highest R^2 for sit-to-stand transitions for both the ankle and knee. In terms of R^2 , the CNN performs best for level ground walking, stair ascent and stair descent, for both the ankle and knee joint. The R^2 values found with the best model per activity are listed. For sit-to-stand (and stand-to-sit) transitions, the R^2 values are 0.456 ± 0.390 and 0.765 ± 0.115 for the ankle and knee joint respectively. For level ground walking, the R^2 values are 0.555 ± 0.264 and 0.830 ± 0.106 for the ankle and knee joint respectively. For stair ascent, the R^2 values are 0.749 ± 0.213 and 0.902 ± 0.070 for the ankle and knee joint respectively. For stair descent, the R^2 values are 0.863 ± 0.091 and 0.884 ± 0.066 for the ankle and knee joint respectively.

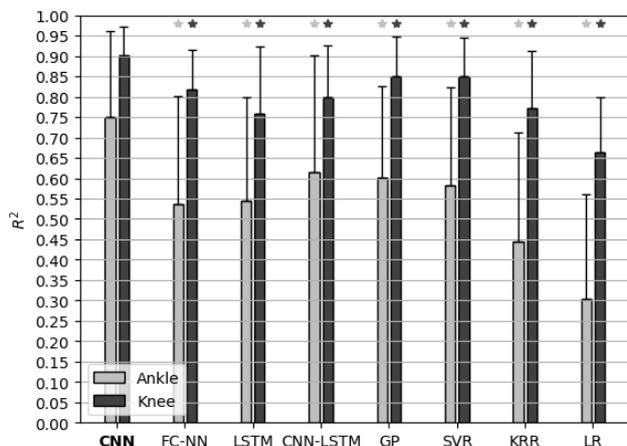
The lowest $RMSE$ values in sit-to-stand transitions are obtained with the FC-NN model. The lowest $RMSE$ values are obtained using the CNN model for level ground walking, stair ascent and stair descent. The joint angle ranges differ for the ankle and knee joint, which influences the magnitude of the $RMSE$. The differences are an important consideration in the comparison of the $RMSE$ values. The ankle angle



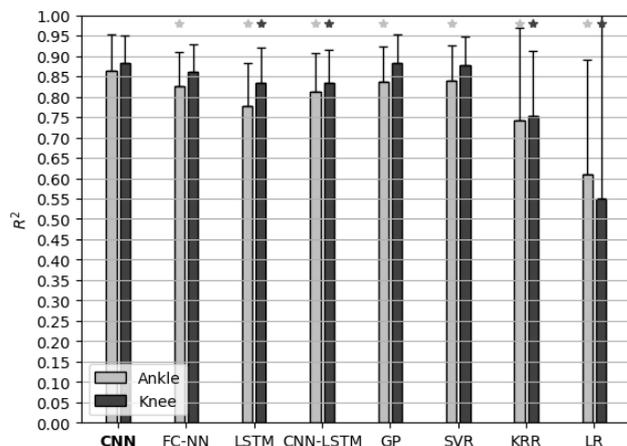
(a) Sit-to-stand transitions in the ENABL3S dataset.



(b) Level ground walking in the ENABL3S dataset.



(c) Stair ascent in the ENABL3S dataset.



(d) Stair descent in the ENABL3S dataset.

Figure 3: R^2 values (mean and standard deviation) of the activities in the ENABL3S dataset for the EMG-only models. The best-performing model is marked bold. Statistically significant different performance between the best-performing model and the other models is shown by an asterisk, for both the ankle (light) and knee (dark).

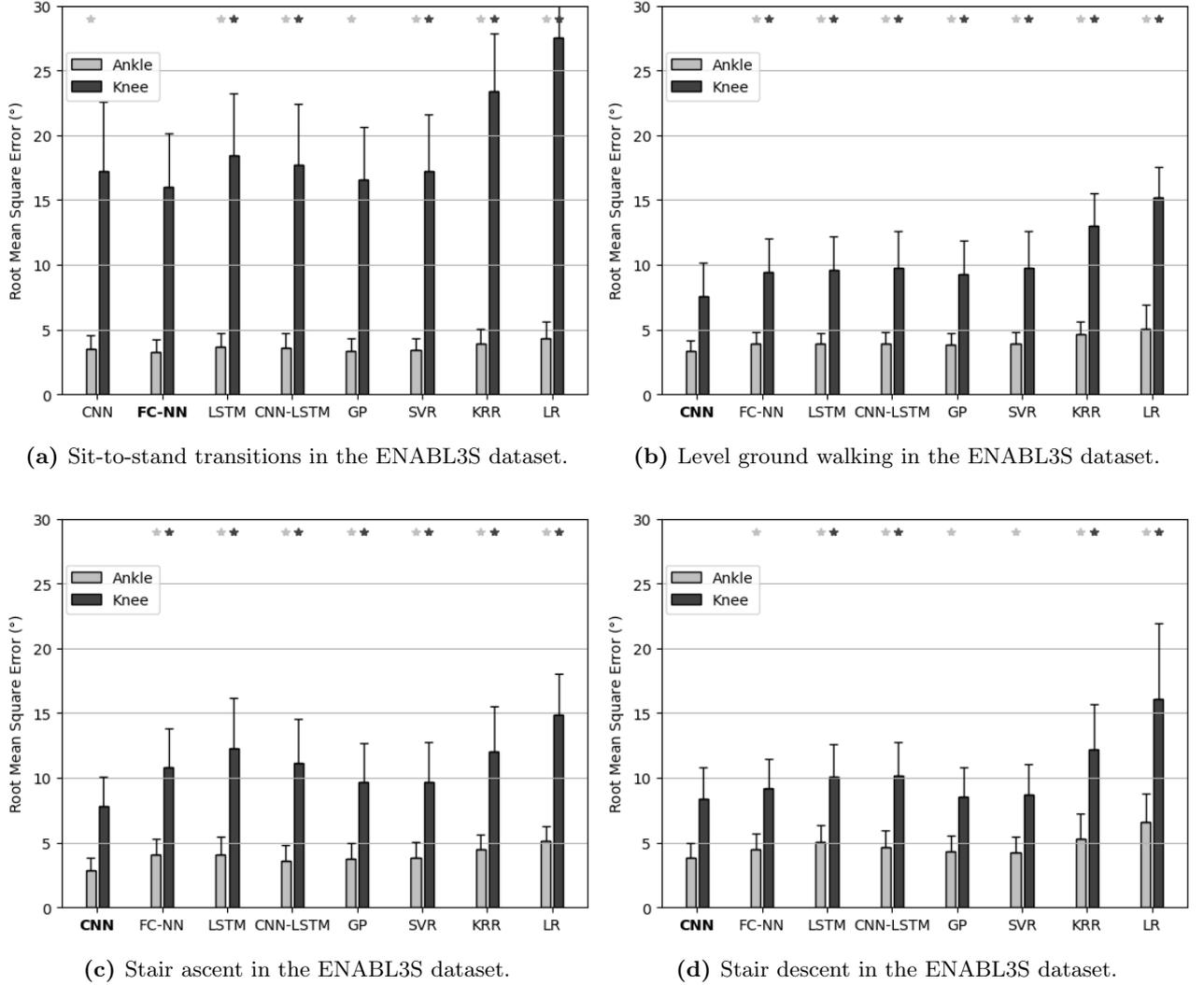


Figure 4: *RMSE* values (mean and standard deviation) of the activities in the ENABL3S dataset for the EMG-only models. The best-performing model is marked bold. Statistically significant different performance between the best-performing model and the other models is shown by an asterisk, for both the ankle (light) and knee (dark).

ranges approximately from -5° to 25° for the sit-to-stand transitions. In the other activities, the ankle angle ranges from approximately -15° to 25° . More differences are observed in the knee angle ranges, with level ground walking having a range of approximately 0° to 65° . Sit-to-stand transitions have a knee joint angle ranging from approximately 0° to 90° . In stair ascent and stair descent, the knee joint angle ranges from approximately 0° to 80° .

Figures 4a, 4b, 4c, 4d show the results of the *RMSE* for the different models in the EMG-only situation. For sit-to-stand (and stand-to-sit) transitions, the *RMSE* values are $3.29 \pm 0.94^\circ$ and $15.99 \pm 4.15^\circ$ for the ankle and knee joint respectively. For level ground walking, the *RMSE* values are $3.35 \pm 0.86^\circ$ and $7.60 \pm 2.54^\circ$ for the ankle and knee joint respectively. For stair ascent, the *RMSE* values are $2.88 \pm 0.97^\circ$ and $7.80 \pm 2.32^\circ$ for the ankle and knee joint respectively. For stair descent, the *RMSE* values are $3.88 \pm 1.12^\circ$ and $8.42 \pm 2.36^\circ$ for the ankle and knee joint respectively.

The selection based on the statistical analysis compares the FC-NN with all other models for sit-to-stand transitions. The CNN is compared with all other models for level ground walking, stair ascent and stair descent. For sit-to-stand transitions, no significant performance difference (after applying the post-hoc analysis) is found for the SVR ($p_{ankle} = 0.093$), the GP ($p_{ankle} = 0.047$ & $p_{knee} = 0.028$) and the CNN ($p_{knee} = 0.059$) models for at least one of the studied joints. For level ground walking and stair ascent, the performance of all models significantly differs from the best-performing model. For stair descent, no significant performance differences (after applying the post-hoc analysis) are found for the GP ($p_{knee} = 0.721$), SVR ($p_{knee} = 0.333$) and FC-NN ($p_{knee} = 0.037$), compared with the CNN for the knee joint. Therefore, the models tested with IMU input are the CNN model for all activities and the FC-NN, SVR and GP models for stair descent and sit-to-stand transitions.

Although not used for the selection procedure, the statistical analysis is also done for the *RMSE* performance. For sit-to-stand transitions, the CNN

($p_{knee} = 0.074$) and GP ($p_{knee} = 0.028$) models have no significant different performance for the knee joint (after applying the post-hoc analysis), compared to the FC-NN. For level ground walking and stair ascent, all models significantly differ from the best-performing model. For stair descent, no significant performance differences are found for the GP ($p_{knee} = 0.721$), SVR ($p_{knee} = 0.169$) and FC-NN ($p_{knee} = 0.028$), compared with the CNN for the knee joint after applying the post-hoc analysis.

3.1.2 EMG and IMU input

Table 2 shows the performance for the models containing EMG and IMU input. Different models perform best using a combination of EMG and historic IMU data. The CNN performs best on both metrics for the knee joint in sit-to-stand transitions. For the ankle joint in sit-to-stand transitions, the best performance in terms of R^2 is found with the GP, whilst the best performance in terms of $RMSE$ is found using the CNN. The CNN performs best (on both metrics) for level ground walking and stair ascent for both joints. Finally, for stair descent the best performance on both metrics is found using the SVR model for the ankle joint and the GP model for the knee joint.

Performance in terms of R^2 for sit-to-stand transitions significantly increased for both the ankle and

knee ($p_{ankle} = 0.047$ & $p_{knee} = 0.005$). For level ground walking, a significant increase of the R^2 is seen for the ankle and knee joints ($p_{ankle} = 0.005$ & $p_{knee} = 0.005$). For stair ascent, a significant increase in R^2 is observed for the ankle ($p_{ankle} = 0.013$ & $p_{knee} = 0.074$). Finally, for stair descent, a significant performance increase in terms of R^2 is seen for the ankle and knee joint ($p_{ankle} = 0.013$ & $p_{knee} = 0.005$).

For sit-to-stand transitions, a significant performance increase (lower $RMSE$) is seen for both the ankle and knee joint ($p_{ankle} = 0.022$ & $p_{knee} = 0.005$). For level ground walking, a significantly lower $RMSE$ is seen for the ankle and knee joint ($p_{ankle} = 0.005$ & $p_{knee} = 0.005$). For stair ascent, a significantly lower $RMSE$ is only seen for the ankle joint ($p_{ankle} = 0.017$ & $p_{knee} = 0.059$). Finally, for stair descent the $RMSE$ is significantly lower for both the ankle and knee joint ($p_{ankle} = 0.007$ & $p_{knee} = 0.005$).

3.1.3 IMU input

Table 2 shows the performance for the models containing IMU-only input. The CNN model performs best on both performance metrics when only historic IMU data is used for all activities. This is seen for both the ankle and knee joint.

Table 2: Results of the selected models for IMU-testing. Both the EMG and IMU and IMU-only results are presented for the ankle and knee joint. R^2 and $RMSE$ (in $^\circ$) are reported using mean (standard deviation). Activities are sit-to-stand transitions (sts), level ground walking (lgw), stair ascent (sa) and stair descent (sd). EMG-only results for the different models are included for the activities that are studied with the IMU analysis.

Ankle								
	sts		lgw		sa		sd	
	R^2	$RMSE$	R^2	$RMSE$	R^2	$RMSE$	R^2	$RMSE$
CNN								
EMG	0.396 (0.360)	3.51 (1.03)	0.555 (0.264)	3.35 (0.86)	0.749 (0.213)	2.88 (0.97)	0.863 (0.091)	3.88 (1.12)
EMG+IMU	0.519 (0.464)	3.01 (1.00)	0.651 (0.245)	2.85 (0.85)	0.786 (0.187)	2.62 (0.95)	0.881 (0.135)	3.34 (1.27)
IMU	0.233 (0.460)	4.07 (1.28)	0.588 (0.248)	3.12 (0.78)	0.672 (0.247)	3.32 (1.04)	0.813 (0.117)	4.38 (1.19)
FC-NN								
EMG	0.456 (0.390)	3.29 (0.94)	-	-	-	-	0.826 (0.084)	4.49 (1.22)
EMG+IMU	0.487 (0.447)	3.11 (0.91)	-	-	-	-	0.882 (0.112)	3.34 (1.21)
IMU	0.136 (0.389)	4.43 (1.49)	-	-	-	-	0.730 (0.180)	5.19 (1.43)
GP								
EMG	0.444 (0.369)	3.35 (0.94)	-	-	-	-	0.836 (0.087)	4.33 (1.18)
EMG+IMU	0.522 (0.408)	3.05 (0.94)	-	-	-	-	0.890 (0.093)	3.29 (1.05)
IMU	0.191 (0.402)	4.24 (1.38)	-	-	-	-	0.774 (0.167)	4.63 (1.49)
SVR								
EMG	0.433 (0.364)	3.40 (0.95)	-	-	-	-	0.840 (0.086)	4.27 (1.20)
EMG+IMU	0.512 (0.403)	3.11 (0.94)	-	-	-	-	0.893 (0.092)	3.24 (1.07)
IMU	0.153 (0.387)	4.36 (1.42)	-	-	-	-	0.768 (0.165)	4.71 (1.42)

Knee								
	sts		lgw		sa		sd	
	R^2	$RMSE$	R^2	$RMSE$	R^2	$RMSE$	R^2	$RMSE$
CNN								
EMG	0.724 (0.159)	17.24 (5.33)	0.830 (0.106)	7.60 (2.54)	0.902 (0.070)	7.80 (2.32)	0.884 (0.066)	8.42 (2.36)
EMG+IMU	0.878 (0.056)	11.36 (2.50)	0.888 (0.125)	5.95 (3.09)	0.913 (0.226)	6.57 (5.14)	0.913 (0.092)	6.97 (2.79)
IMU	0.559 (0.163)	21.93 (4.43)	0.829 (0.156)	7.38 (3.56)	0.854 (0.204)	9.35 (4.78)	0.751 (0.144)	12.22 (3.48)
FC-NN								
EMG	0.765 (0.115)	15.99 (4.15)	-	-	-	-	0.862 (0.068)	9.23 (2.25)
EMG+IMU	0.842 (0.106)	12.75 (3.67)	-	-	-	-	0.909 (0.091)	7.18 (2.74)
IMU	0.426 (0.172)	25.16 (4.30)	-	-	-	-	0.525 (0.232)	16.90 (4.36)
GP								
EMG	0.751 (0.111)	16.56 (4.06)	-	-	-	-	0.882 (0.070)	8.52 (2.29)
EMG+IMU	0.842 (0.063)	13.06 (2.69)	-	-	-	-	0.935 (0.055)	6.11 (2.22)
IMU	0.493 (0.130)	23.75 (3.80)	-	-	-	-	0.702 (0.203)	13.06 (4.36)
SVR								
EMG	0.730 (0.125)	17.26 (4.34)	-	-	-	-	0.876 (0.071)	8.73 (2.36)
EMG+IMU	0.828 (0.071)	13.62 (3.02)	-	-	-	-	0.933 (0.056)	6.20 (2.20)
IMU	0.444 (0.137)	24.89 (4.02)	-	-	-	-	0.683 (0.222)	13.46 (4.46)

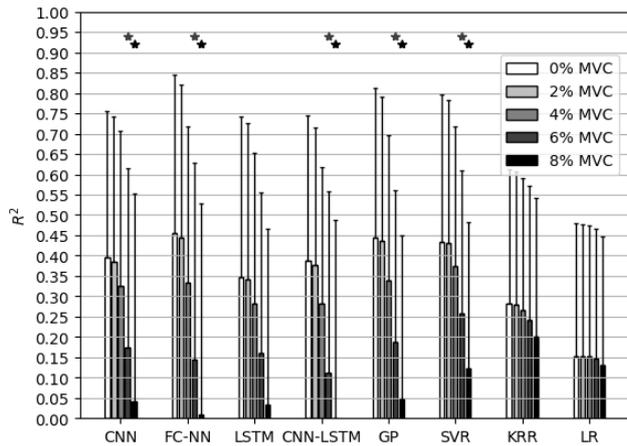
For sit-to-stand transitions, a significant decrease of the R^2 values is found for both the ankle and knee joint ($p_{ankle} = 0.007$ & $p_{knee} = 0.013$). For level ground walking, no significant difference in R^2 values is seen for both the ankle and knee joint ($p_{ankle} = 0.241$ & $p_{knee} = 0.799$). For stair ascent, the R^2 values significantly decrease for the ankle joint ($p_{ankle} = 0.005$ & $p_{knee} = 0.074$). Finally, for stair descent a significant decrease is seen for the R^2 values of both the ankle and knee joint ($p_{ankle} = 0.007$ & $p_{knee} = 0.007$).

Sit-to-stand transitions have a significant increase of the $RMSE$ for both the ankle and knee joint ($p_{ankle} = 0.005$ & $p_{knee} = 0.013$). For level ground walking, the $RMSE$ metric does not significantly differ for the ankle and knee joint ($p_{ankle} = 0.074$ & $p_{knee} = 0.575$). For stair ascent, a significant increase of the $RMSE$ is seen for both the ankle and knee joint ($p_{ankle} = 0.005$ & $p_{knee} = 0.028$). Finally, for stair descent a significant increase of the $RMSE$ is found for both the ankle and knee joint ($p_{ankle} = 0.017$ & $p_{knee} = 0.007$).

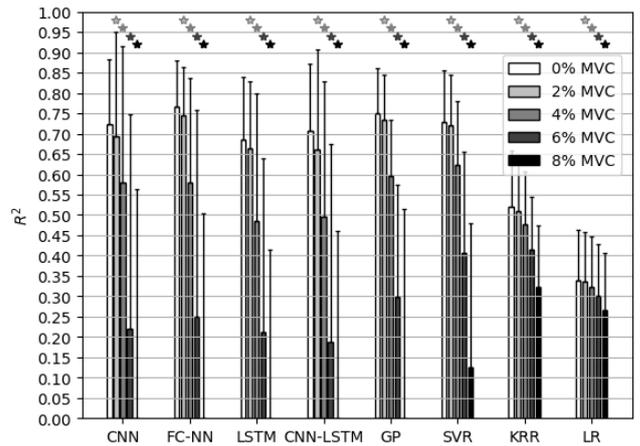
3.1.4 Robustness analysis

Figures 5a and 5b show the R^2 values in the selected noise levels for the ankle and knee joint respectively. Performance decreases with higher noise levels for all types of models. At the 2% and 4% MVC noise levels, the post-hoc analysis shows the decrease in performance is not significant ($p > 0.05$) for all models for the ankle joint. At the 6% and 8% MVC levels, the CNN, FC-NN, CNN-LSTM, GP and SVR models have a significant performance decrease. The post-hoc analysis shows that the LSTM, LR and KRR models have no significant decrease in performance in terms of R^2 for the ankle joint. A significant decrease in performance in terms of R^2 is found with all models in all tested noise conditions for the knee joint.

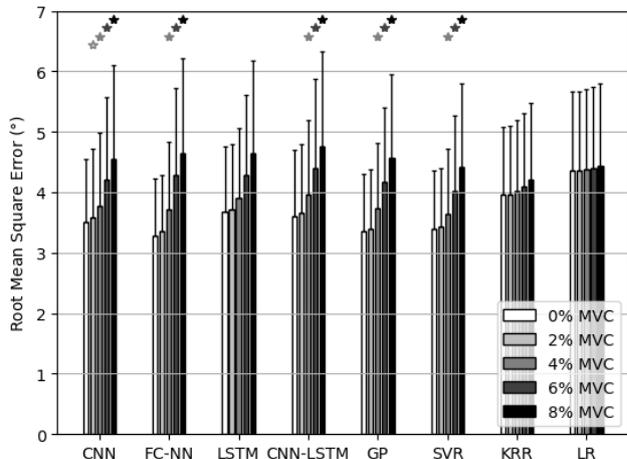
The $RMSE$ values for the ankle and knee joint in the tested noise conditions are shown in figures 5c and 5d. Performance differences on the $RMSE$ metric for the various noise levels differ slightly with the R^2 metric. The CNN has a significant difference in performance at the 2% MVC level for the ankle joint.



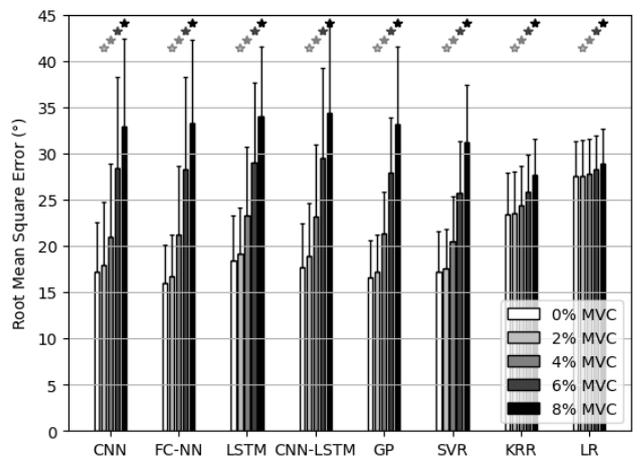
(a) R^2 for the ankle joint at different noise levels.



(b) R^2 for the knee joint at different noise levels.



(c) $RMSE$ for the ankle joint at different noise levels.



(d) $RMSE$ for the knee joint at different noise levels.

Figure 5: R^2 (a,b) and $RMSE$ (c,d) values (mean and standard deviation) of the different noise levels in the sit-to-stand transitions of the ENABL3S dataset. Statistically significant different performance between the 0% MVC level and the other tested noise levels is shown by an asterisk.

All models except for the LSTM, LR and KRR models have a significantly different $RMSE$ starting at the 4% MVC level. The post-hoc analysis shows that the LSTM, LR and KRR models have no significant performance difference in terms of $RMSE$ for the ankle joint. A significant decrease in performance in terms of $RMSE$ is found with all models in all tested noise conditions for the knee joint.

3.2 MLK dataset

Figures 6a and 6b show the averaged stand-sit-stand repetition based on all repetitions for a single subject, using the activity-specific model. Both the true and estimated stand-sit-stand repetitions are averaged. Comparison of figures 6a and 6b shows that the ankle angle has relatively more variation than the knee angle. The degree of variation can be seen from the standard deviations associated with the averaged true joint angles.

Figures 7a and 7b show the performance in terms of R^2 values for the different models, for both the sitting-specific and sitting-generic models. Highest R^2 values in non-weight-bearing tasks are obtained using the CNN model. Highest R^2 values in sit-to-stand transitions are obtained using the SVR model and CNN model for the ankle and knee joint respectively. For the non-weight-bearing tasks, the R^2 values are 0.839 ± 0.038 and 0.956 ± 0.013 for the ankle and knee joint respectively. For sit-to-stand transitions, the highest R^2 values are 0.476 ± 0.356 and 0.952 ± 0.025 for the ankle and knee joint respectively.

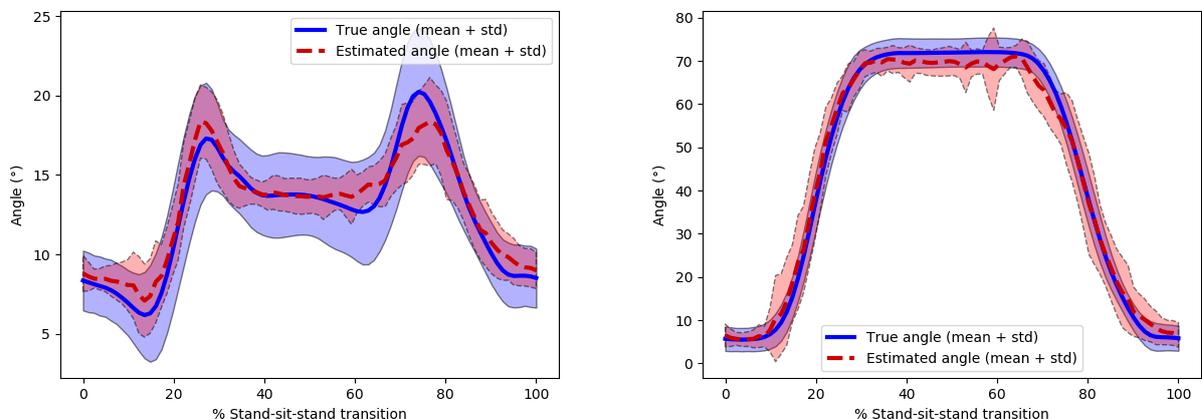
For the sitting-generic models, the CNN model gives the highest R^2 values for the ankle joint in non-weight-bearing activities and both joints in sit-to-stand transitions; the FC-NN model gives the highest R^2 value for the knee joint in the non-weight-bearing tasks. The sitting-generic models yield R^2 values of 0.842 ± 0.040 and 0.955 ± 0.010 for the ankle and knee

joint in the non-weight-bearing tasks. The sitting-generic models have highest R^2 values of 0.040 ± 0.296 and 0.942 ± 0.017 for the ankle and knee joint in the sit-to-stand transitions.

Figures 7c and 7d show the $RMSE$ values for the two activities in the MLK dataset. The ankle angle ranges from approximately -60° to 15° for the non-weight bearing tasks and approximately 5° to 25° for the sit-to-stand transitions. The knee angle ranges from approximately 0° to 90° for the non-weight bearing tasks and approximately 0° to 80° for the sit-to-stand transitions.

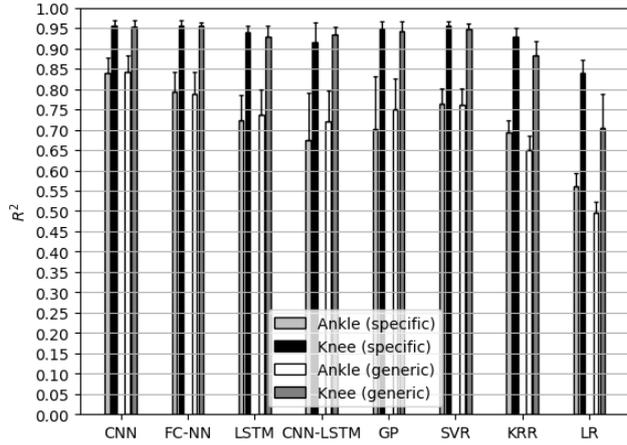
The lowest $RMSE$ values in the non-weight-bearing activities are obtained using the CNN model. The lowest $RMSE$ values in sit-to-stand transitions are obtained using the SVR model and CNN model for the ankle and knee joint respectively. For the non-weight-bearing tasks, the lowest $RMSE$ values are $8.62 \pm 1.48^\circ$ and $7.21 \pm 1.18^\circ$ for the ankle and knee joint respectively. For the sit-to-stand transitions, the lowest $RMSE$ values are $2.44 \pm 0.80^\circ$ and $5.97 \pm 1.24^\circ$ for the ankle and knee joint respectively.

For the sitting-generic models, the CNN model gives the lowest $RMSE$ values for the ankle joint in non-weight-bearing activities and both joints in sit-to-stand transitions; the FC-NN model gives the lowest $RMSE$ value for the knee joint in the non-weight-bearing tasks. The lowest $RMSE$ values for the sitting-generic models are $8.56 \pm 1.60^\circ$ and $7.34 \pm 0.93^\circ$ for the ankle and knee joint respectively in the non-weight-bearing tasks. The lowest $RMSE$ values for sitting-generic models are $3.52 \pm 1.19^\circ$ and $6.84 \pm 1.45^\circ$ for the ankle and knee joint in the sit-to-stand transitions.

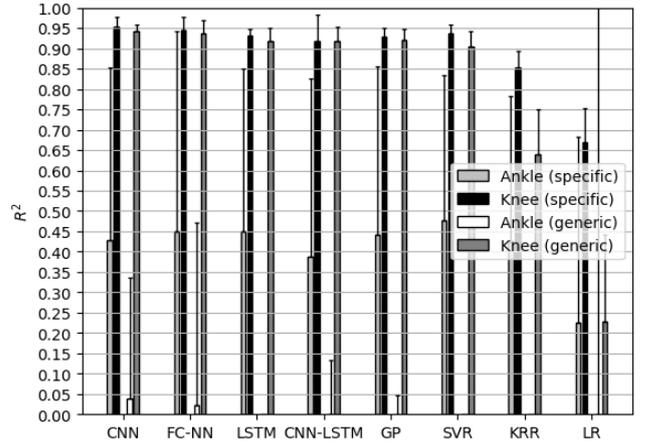


(a) Averaged ankle angle in a stand-sit-stand transition in the MLK dataset. (b) Averaged knee angle in a stand-sit-stand transition in the MLK dataset.

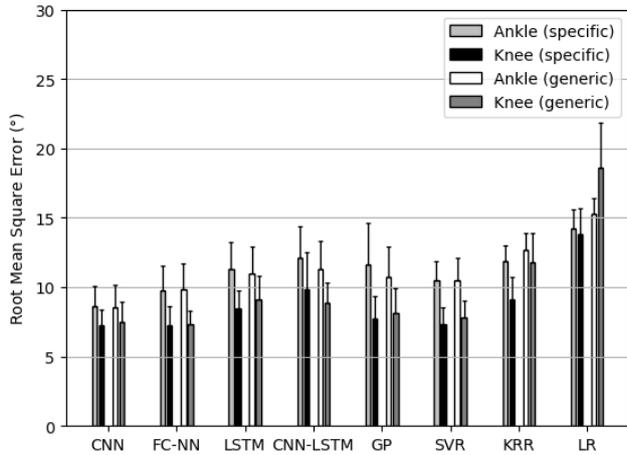
Figure 6: Average of all repetitions for the ankle (a) and knee (b) angles during a stand-sit-stand task for a subject in the MLK dataset.



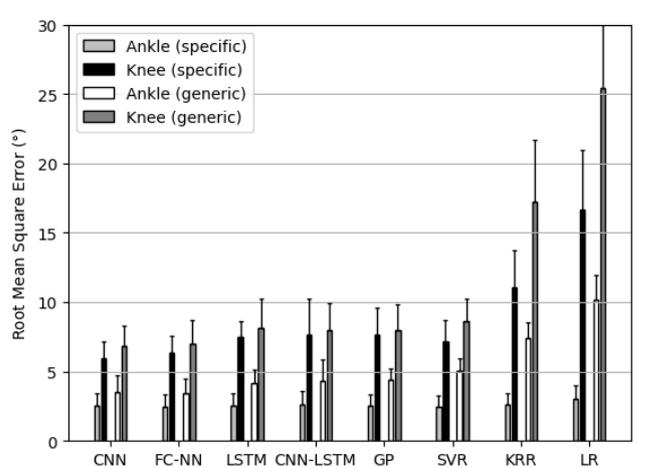
(a) R^2 for non-weight-bearing tasks in the MLK dataset.



(b) R^2 for sit-to-stand transitions in the MLK dataset.



(c) $RMSE$ for non-weight-bearing tasks in the MLK dataset.



(d) $RMSE$ for sit-to-stand transitions in the MLK dataset.

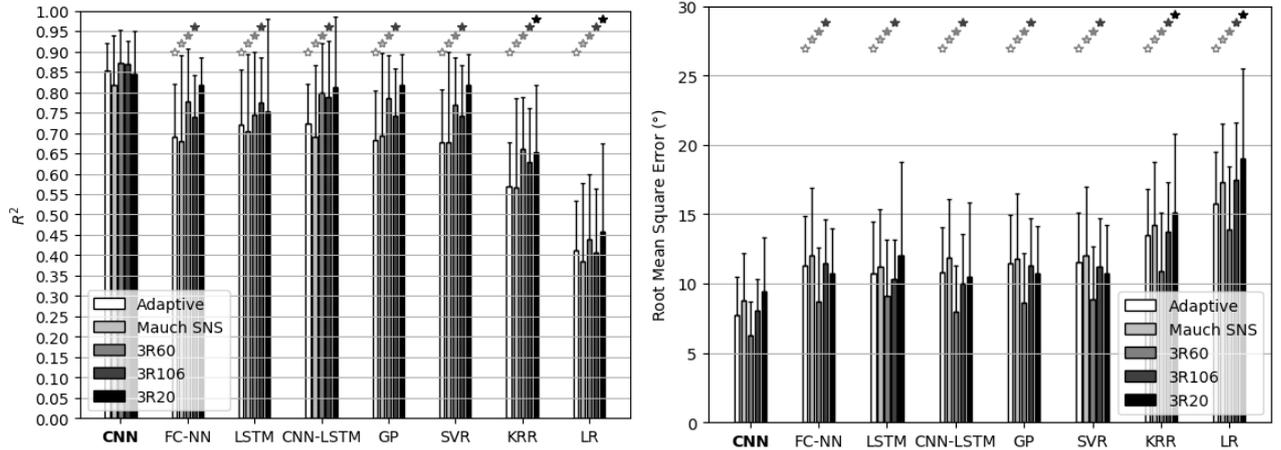
Figure 7: R^2 (a,b) and $RMSE$ (c,d) values (mean and standard deviation) of the activities in the MLK dataset. Results are shown both for sitting-specific models (left bars of each model) and the sitting-generic models (right bars of each model).

3.3 TIPS dataset

Figures 8a and 8b show the performance of the chosen models on the TIPS dataset. From both figures, it becomes clear that the CNN outperforms all other models, for every tested prosthesis. Except for the Ottobock 3R20 prosthesis, significant performance differences are found between the CNN and all other models. Significant performance differences with the CNN for the 3R20 prosthesis are only found with the KRR and LR models. This is seen for both the R^2 metric and the $RMSE$ metric.

The R^2 value found for the best-performing model is 0.853 ± 0.068 for the Adaptive prosthesis. For the Mauch SNS prosthesis, the R^2 value is 0.818 ± 0.122 . For the Ottobock 3R60, the R^2 is 0.871 ± 0.082 . The R^2 value found for the Ottobock 3R106 is 0.868 ± 0.058 . Finally, the R^2 value found for the Ottobock 3R20 is 0.844 ± 0.105 .

Large standard deviations can be observed for the $RMSE$ values shown in figure 8b as a result of variability in knee joint angle ranges made by the different subjects in the TIPS dataset. The lowest $RMSE$ values for the different prostheses are found using the CNN. For the Adaptive prosthesis, the $RMSE$ value is $7.76 \pm 2.77^\circ$. For the Mauch SNS prosthesis, the $RMSE$ value is $8.77 \pm 3.40^\circ$. For the Ottobock 3R60, the $RMSE$ is $6.31 \pm 2.37^\circ$. For the Ottobock 3R106, the $RMSE$ is $8.06 \pm 2.31^\circ$. Finally, the $RMSE$ of the Ottobock 3R20 is $9.46 \pm 3.87^\circ$.



(a) R^2 values for the knee joint estimations in the TIPS dataset.

(b) $RMSE$ values for the knee joint estimations in the TIPS dataset.

Figure 8: R^2 (a) and $RMSE$ (b) values (mean and standard deviation) of the different prostheses tested in the TIPS dataset. The best-performing model for each prosthesis is marked bold. Statistically significant different performance between the best-performing model and the other models is shown by an asterisk, for each of the tested prostheses.

4 Discussion

Eight different algorithms were tested on their ability to estimate lower limb joint kinematics from electromyography. Four objectives were set up to determine which algorithm(s) performed best. The objectives of this study were: 1) Determine performance of the algorithms in non-weight-bearing situations and sit-to-stand transitions. 2) Determine performance of the algorithms for different weight-bearing ADLs. 3) Assess the influence additional historic IMU data had on the performance. 4) Determine the performance for transfemoral amputees.

The different studied activities and situations showed that different algorithms can obtain similar performance. Generally, the CNN, FC-NN, GP and SVR showed the best performance for learning the non-linear relationship between EMG and joint angles. The linear method showed lower performance compared to the aforementioned algorithms, which showed that learning a linear relationship was not sufficient to accurately learn the relation between EMG and joint angles. Learning a linear relationship after transformation to a higher dimensional space, which is done by the KRR algorithm, was also outperformed by the aforementioned algorithms. This showed that the applied transformation did not result in a completely linearised solution space. Neural networks that included LSTM layers showed lower performance than the other types of neural networks, which showed that this type of layer did not have additional benefits in the presented analysis. Possibly, due to the increased complexity of the LSTM-based models, the relation between EMG and kinematics could not be learned as accurately with the available data. Comparison of the different activities showed that for most of the activities, the best performance was obtained using a CNN. The results suggested that the CNNs were able to ex-

tract most information from the EMG signals to learn the relationship between joint kinematics and EMG.

The remainder of this section is divided into six subsections. First, a reflection is presented for each individual objective. Next, other findings, not directly related to the objectives, are presented. Finally, recommendations for future research are outlined.

4.1 Non-weight-bearing tasks & sit-to-stand transitions

The first objective of this thesis was to find the best approach for the estimation of joint kinematics in non-weight-bearing situations and sit-to-stand transitions. This analysis was done for both sitting-specific models and sitting-generic models. First the sitting-specific models are discussed per activity, secondly the sitting-generic results are discussed.

4.1.1 Non-weight-bearing tasks

From the R^2 values for the non-weight-bearing tasks, it could be concluded that the knee joint angles were predicted accurately. Good performance was also obtained for the ankle joint estimations, although the mean performance is lower than for the knee joint. Using a CNN, R^2 values of 0.839 ± 0.038 and 0.956 ± 0.013 were found for the ankle and knee respectively. Especially for the knee joint, these results were promising to be applied for direct voluntary control of the prosthesis in the sitting state. Caution has to be taken in the generalisability of the results, as only three able-bodied subjects could be measured.

An additional benefit to the reported performance was that the non-weight-bearing tasks consisted of three different ankle/knee movement tasks (knee flexion/extension and ankle plantar-/dorsiflexion in different knee positions). Thus, a single model was able

to accurately predict the angles required for different tasks. Therefore, no further transitions between control states are required to control the prosthesis for the execution of a desired movement in the sitting state. Creation of such a model for estimation of non-weight-bearing tasks, requires training data on all activities. Therefore, an extensive calibration routine has to be conducted when the algorithm is used on the target population, to ensure that desired ankle and knee movements can be made.

Comparison of the non-weight-bearing results with literature is difficult, as different data collection procedures were set up. Studies that estimate joint angles in a non-weight-bearing situation focussed on estimating one type of leg movement (extension-flexion of the knee), where different movement speed and loads were studied [11, 13]. The lowest *RMSE* values found for the non-weight-bearing tasks (ankle: $8.62 \pm 1.48^\circ$, knee: $7.21 \pm 1.18^\circ$) were comparable with the *RMSE* values presented in the work by Zhang et al. for leg extension with small load [11]. Zhang et al. reported *RMSE* values of $7.32 \pm 1.82^\circ$ and $11.32 \pm 2.15^\circ$ for the ankle and knee joint respectively. Zhang et al. [11] used the SRE of the EMG as input to the studied models. The work by Yang et al. [13] showed that the *RMSE* decreased when using multiple EMG features to predict knee joint angles. The used EMG features were the Root Mean Square, Wavelet Coefficients and Permutation Entropy of the EMG signals. At comparable leg movement speeds (without load) to our study, the authors reported *RMSE* values of $2.79 \pm 0.37^\circ$. In the presented study it was chosen to use a single feature (SRE of the EMG) per measured EMG channel for comparison of the different models. The focus of the study was the comparison of various machine learning algorithms instead of comparison of various EMG features, while retaining as much similarity in input data as possible. Possibly, extracting different EMG features could result in lower *RMSE* values for the non-weight-bearing tasks in the MLK dataset. As performance could not only improve for non-weight-bearing tasks, further discussion is provided in section 4.5 (General findings).

4.1.2 Sit-to-stand transitions

Two datasets were used to determine the performance for sit-to-stand transitions. Similar performance of the ankle angle estimations was observed for both datasets. In both datasets, performance for the ankle joint was poor, as mean R^2 performance was below 0.5. A large degree of variation in ankle joint angles was observed in the repetitions of a sit-to-stand transition in both datasets. An example of this large degree of variation was shown in figure 6a, as a large standard deviation was observed for the true ankle angles. The used algorithms learned a more generalised pattern for the ankle angle during sit-to-stand transitions, as reflected by the lower standard deviation for the estimated angles in figure 6a. Hence, as the different algorithms were not able to learn the degree of variation between the

ankle angles, poor overall performance was obtained for the ankle joint. Therefore, it could be concluded that the estimation of ankle angles based on EMG was not feasible to apply in sit-to-stand transitions.

A simulation study by Yoshioka et al. [41] showed that muscle forces generated by knee and hip extensors were required to perform a sit-to-stand transition. Muscle force generation of the ankle plantar- and dorsiflexors was not necessarily required to complete the transitions. An earlier study by Yoshioka et al. [42] showed that a successful sit-to-stand transition does not require a joint moment at the ankle joint. Hence, active assistance in sit-to-stand transitions by the prosthesis could be possible in the absence of an accurate estimate of the desired ankle kinematics when the knee can be supported sufficiently.

Performance over the two datasets differed for the knee joint. In the MLK dataset, an R^2 value of 0.952 ± 0.025 was found (using a CNN), compared with 0.765 ± 0.115 in the ENABL3S dataset (using a FC-NN). Furthermore, large performance differences were seen in terms of *RMSE* for both datasets (MLK: $5.97 \pm 1.24^\circ$, ENABL3S: $15.99 \pm 4.15^\circ$). Comparing both datasets, less variation in repetitions in a sit-to-stand transition was observed in the MLK dataset. In the MLK dataset, the sit-to-stand transitions were collected by performing thirty consecutive repetitions. In the ENABL3S dataset, the sit-to-stand transitions were the starting point and end point of a circuit. In this circuit, subjects were allowed to transition voluntarily between locomotion tasks [22]. Due to this difference in data collection procedure, more variation in the true knee joint angles was seen for the sit-to-stand transitions data in the ENABL3S dataset compared with the MLK dataset.

Another difference between the two datasets were the measured muscles. The ENABL3S dataset contained EMG of the soleus muscle, which assists in plantarflexion of the ankle. Although an additional muscle that directly acts on the ankle joint was present in the ENABL3S dataset, no considerable differences in performance were observed for the ankle joint. In the MLK dataset, two other muscles were measured compared to the ENABL3S dataset: the adductor magnus and gluteus maximus. Possibly, their relation to stabilising the hip joint and extending the knee and hip joint respectively, resulted in better estimation of the knee joint. Changing the included EMG channels, so that datasets had the same EMG channels, was not further investigated in this study. The reason for this was that additional hyperparameter optimisation (for reduced input data) is necessary to make a fair comparison.

As the number of subjects included in the MLK dataset was lower compared to the ENABL3S dataset, generalisability for the former dataset was limited. Hence, the overall applicability for estimating knee joint angles during sit-to-stand transitions is doubtful. Extension of the data collection for the MLK dataset could show whether the obtained average performances generalise to a larger population than three subjects.

If the extension of the dataset shows that the results on the MLK dataset are generalisable, training models on fully standardised stand-sit-stand repetitions could be a feasible approach to predict knee joint angles. Further extension of the applicability of this approach could be realised by training models for varying types (and heights) of seats. Using the predicted knee joint angles, control signals can be presented to the prosthesis so that additional energy can be generated by the prosthesis to assist in the sit-to-stand transition. Providing active assistance by the prosthesis during sit-to-stand transitions can help in improving the quality of life and mobility of a transfemoral amputee [43].

4.1.3 Sitting-generic models

A preliminary analysis revealed that the sitting-specific models were not able to generalise towards a different activity. Using a model that was trained on a sitting-generic level showed promise for the application to create complete voluntary control in a sitting situation. In all conditions, except the ankle angle in sit-to-stand transitions, comparable performance to sitting-specific models was obtained when a sitting-generic model was used to predict joint angles. Only results for the ankle angle in the sit-to-stand transitions showed considerably lower performance. Performance for the ankle angle in the sitting-specific situation already showed low performance, due to the large degree of variation of the true ankle angle. Furthermore, the observed ankle angle ranges differ in both types of activities. In the non-weight-bearing tasks, the ankle joint moved mainly towards a plantarflexed position (angle range from approximately -60° to 15°), whereas in the sit-to-stand transitions the ankle joint was in dorsiflexed position (angle range from approximately 5° to 25°). Hence, the generalisability to both situations became poor for the sit-to-stand transitions, as a large degree of variation was present in the training set. The variation was observed less for the knee joint, as for both the non-weight-bearing tasks and sit-to-stand transitions, flexion and extension of the knee were performed with comparable angle ranges (non-weight-bearing tasks: 0° to 90° , sit-to-stand transitions 0° to 80°).

4.2 ADLs

The second objective of this thesis was to determine the performance of estimating joint kinematics in different weight-bearing ADLs, to assess if a voluntary control approach could be feasible for amputees in these activities.

Realisation of safe voluntary control in a weight-bearing situation requires that prediction errors remain small, as errors can result in an imbalance or even falls for the prosthesis user [44]. Estimation of knee joint angles showed promising results for the three studied ADLs. Mean R^2 values were 0.830, 0.902 and 0.884 for level ground walking, stair ascent and stair descent respectively. Whether these perfor-

mances were sufficient for providing safe enough control over the prosthetic knee, could not be concluded from the results. Only for stair descent estimation of ankle joint angles showed comparable performance in terms of R^2 (0.863). Hence, only for this activity application of voluntary control of the ankle joint could be feasible. It is expected that the prediction errors of the ankle angle result in an unsafe situation for the other tested ADLs.

Comparison of the ankle and knee angles in the different ADLs showed that the ankle had more sudden changes (dorsiflexion to plantarflexion and vice versa) in the joint angles for the studied activities. The knee joint angles had more gradual changes from flexion to extension (and extension to flexion). Therefore, for the estimation of the knee joint angles, relatively less variation needed to be learned by the models compared with the ankle joint. This difference resulted in better performance in the estimation of knee joint angles compared to ankle joint angles. Furthermore, more muscles that acted directly on the knee joint were available compared with muscles that acted on the ankle joint.

Joint angle estimation based on EMG for level ground walking was studied earlier, but comparable studies were not done for the other studied ADLs. Chen et al. [12] used a simple neural network combined with deep belief neural networks to estimate lower limb joint angles during level ground walking. Chen et al. found an $RMSE$ of $2.45 \pm 0.57^\circ$ for the ankle joint, which was better than the results reported in this study. Furthermore, they presented an $RMSE$ of $3.96 \pm 0.69^\circ$ for the knee joint, which was considerably lower than the $RMSE$ reported in this study (ankle: $3.35 \pm 0.86^\circ$, knee: $7.60 \pm 2.54^\circ$). Usage of deep belief networks to select different lower-dimensional features from the SRE of the EMG signals could be the reason for the better performance reported by Chen et al. [12]. However, the authors presented no results on what performance was obtained with the original SRE features.

Task-specific models were created, as the high-level controller of the MyLeg prosthesis recognises different locomotion tasks. The state-machine can select the appropriate low-level control of the prosthesis after recognition of the locomotion task. To assess if voluntary control in the locomotion states could be preferred over the set low-level control by the states, an assessment was done on what performance could be achieved for each individual activity. Incorrect selection of the locomotion state can result in safety issues, both when relying on pre-defined low-level control or voluntary control of the prosthesis for the recognised activity. Compared to a task-specific model, a generic weight-bearing voluntary control algorithm could reduce the safety issues associated with selection of the wrong state. However, as it is unsure whether the developed models for specific activities are safe enough to use, it is expected that a generalisation to a generic weight-bearing algorithm suffers from the safety issues related to erroneous predictions.

4.3 Influence IMUs

The third objective was to assess the influence of the addition of historic information from IMUs on the model performances. Addition of historic information from IMUs, on the shank and thigh, to the EMG data significantly improved performance in almost all of the studied activities. This result implied that additional information on movement of the leg has a positive influence on the model’s predictive abilities. These findings corresponded with work presented by Stival et al. [45], who estimated joint angles for two activities in the upper extremity. Although no significance was reported in their study, a performance increase was reported when combining EMG and accelerometer data.

The use of the historic IMU feature without EMG signals significantly decreased the performance in most of the studied activities. Especially for the sit-to-stand transitions a relatively large performance decrease was observed, as the R^2 decreased approximately 0.2 for both the ankle and knee joint. Compared to the other studied activities, the sit-to-stand transition was a relatively static movement, where the subject did not need to perform large movements of the thigh and shank. During level ground walking more thigh and shank movement was observed, which resulted in a not significant difference in the performance.

Reconstructing joint angles based on gyroscope and accelerometer data is commonly done using IMUs. In such reconstruction schemes, an accurate estimate of the joint angles could be obtained from combining sensor readings from the IMUs, a subject-specific scaled biomechanical model and a Kalman filter [46]. A systematic review by Poitras et al. [47] showed that validity of estimating ankle and knee joint angles based on IMUs was high for the sagittal plane. The reconstruction procedure differed from the applied machine learning approach to predict joint angles, as estimations of the machine learning approach were based solely on the learned relationship between input features and the target output.

Machine learning, and specifically the use of neural networks, was previously used to estimate knee joint angles based on a minimal IMU set-up. Using relative orientations (in quaternions, with respect to the pelvis) of IMUs on both shanks and the pelvis, knee joint angles could be estimated accurately in running [48]. Hence, using different information from the IMUs could give an improvement to the IMU-based estimates. Although estimation of joint angles based solely on IMUs was not the goal of this study, the results showed that the selected IMU feature could be improved upon, to further increase the estimation performance when using EMG in combination with historic IMU data. Using the estimation procedure of Wouda et al. [48] in a historic context, to predict desired joint angles ahead in time, should be studied to determine if performance can be improved. The ENABL3S dataset did not easily facilitate this approach, as no orientation data from the IMUs was presented in the published data [22].

4.4 Transfemoral amputees

The final objective of this study was to determine the performance of the algorithms when applied to transfemoral amputees. This analysis gave a first indication of the feasibility of the proposed approach for the intended users. The results on the TIPS dataset indicated that the CNN performed best in learning the relationship between the EMG of the stump muscles with the knee angles obtained using the different prostheses. The CNN significantly outperformed all other models on both performance metrics for all tested prostheses except for the 3R20 prosthesis.

Comparison of knee joint angle estimation during level ground walking for able-bodied subjects and amputees could be done by comparing the results of the ENABL3S and TIPS datasets. Three important notes had to be made with this comparison. Firstly, the knee angle for able-bodied subjects had more variation during the stance phase. Secondly, the ENABL3S dataset contained more repetitions for each subject. Hence, in the ENABL3S dataset more variability needed to be accounted for by the models. On the other hand, more data was available to learn the more complex joint angles. Thirdly, EMG from the TA and GM were available for the able-bodied subjects. Although these considerations had to be taken into account for the comparison of both results, the estimation performance for the amputees was slightly better than the performance for able-bodied individuals (highest R^2 of 0.871 ± 0.082 and 0.830 ± 0.106 for the TIPS dataset and ENABL3S dataset respectively). These results suggested that using EMG of the residual muscles in the stump of an amputee, in combination with a CNN, was a feasible approach to estimate knee joint angles during level ground walking for an amputee.

Comparison of the various prostheses showed small differences in performance. Comparison of all tested models, the more complex prostheses (Adaptive and Mauch SNS) showed slightly lower performance than the less complex prostheses. The difference in performance can be explained by the behaviour of the less complex prostheses in the stance phase. The 3R20, 3R106 and 3R60 prostheses showed almost no variability in joint angles during the stance phase of level ground walking. Therefore, the joint angles required for level ground walking were less complex to learn for the models. Even though only limited data points were available to train the models, the complexity of the joint angles throughout the activity made it possible for the models to learn the relationship based on the EMG of the muscles in the stump. No comparable work was found that estimated prosthetic knee joint angles based on EMG signals of the stump of a transfemoral amputee.

A different approach was presented by Farmer et al. [49] for estimation of ankle joint angles in transtibial amputees. Their study used a nonlinear autoregressive neural network with exogenous input (NARX) to estimate the ankle angle in level ground walking. EMG signals were used as exogenous input of the model.

The NARX approach resulted in average *RMSE* values ranging from (1.2° to 5.4°). Due to the cyclic nature of the ankle angle during gait, the model relied mainly on the autoregressive feedback of the predicted joint angles. However, their work also showed that including EMG as exogenous input, reduced the reported estimation errors. As the main objective of this study is to realise direct voluntary control in the sitting state, where cyclicity of movements is not necessarily present, it is doubtful whether a NARX can be successfully applied to non-weight-bearing situations.

4.5 General findings

Next to the objectives, several other findings require additional discussion.

Testing of robustness to additional noise showed that the performance declined similarly for most of the studied models. At low noise levels (2% and 4% MVC), the models showed some robustness to noise for the ankle joint, as no significant differences were found for the R^2 performance. At higher noise levels, all models, except for the KRR, LR and LSTM models showed no robustness in performance for the ankle joint. All noise conditions showed a significant decrease in performance of the knee joint for all models. The best-performing models (CNN, FC-NN, GP and SVR), considering all activities, showed very similar behaviour over the different noise conditions. Only the CNN showed slightly lower robustness for the ankle joint, as a significant different *RMSE* value is found starting at the 2% MVC level ($p = 0.047$). These differences were not found for the R^2 metric, which showed that the additional noise had a comparable effect on the best-performing models. Evaluation of absolute performance degradation showed that most of the studied models had a relatively large difference between the baseline level and the 8% MVC level, except for the LR and, to a lesser extent, the KRR model. However, comparison with the different studied activities showed that the baseline performance for the LR and KRR models (0% MVC) was not sufficient to benefit from the robustness to noise of these models.

In offline testing of upper limb prosthesis control, the robustness of the LR model to additional noise on the EMG signal was not observed [35]. On the other hand, robustness was observed when training the model on a first day and testing the model on the second day [36]. It should be noted that the outcome studied in [35, 36], had a different nature. In both studies [35, 36], the performance was determined by assessing if subjects retained the ability to complete prosthesis control tasks under different conditions than what the algorithms were trained on. The work presented in this thesis studied whether performance of correct estimation of joint angles could be retained in conditions unseen during training.

At higher noise levels, the *RMSE* metric showed large errors compared to the lower noise levels (e.g. for the FC-NN: at baseline a mean *RMSE* of 15.99° compared to the 8% MVC level with a mean *RMSE*

of 33.29°). The large increase could be explained by the effect the noise contamination has on the EMG signals. At higher noise levels, the amplitude of the SRE increased. If there was little muscle activity (when sitting for instance), the effect of the additional noise was relatively large. The predictions showed that the joint angles estimated in a sitting position were close to the joint angles observed while the subject was standing, because of the greater muscle activity in the noisy test data. Evidently, large estimation errors were seen which was shown by the *RMSE* values of these models.

This study used a Bayesian method to find optimal hyperparameters for each studied activity, which was not described in found literature for joint angle estimation studies. Earlier work either used a grid search to find optimal hyperparameters [16, 19] or only reported what hyperparameter values were chosen to use [12, 13]. The Bayesian optimisation technique increased the possibility to search through larger regions of the hyperparameter space, whilst limiting the additional computational cost associated with searching through a more extensive hyperparameter space. Especially for the computationally complex models (the neural networks) this strategy gave an efficient method to find a model’s optimal structure and settings.

The optimal hyperparameters were found on a subject- and activity-generic level, whilst testing was performed on a subject-specific model. This strategy was chosen to prevent overfitting, without requiring many hyperparameter optimisation rounds. As a limited amount of data was available for the various activities, the model performance in terms of generalisability is tested by developing five different instances of the same model per subject (5-cv). Finding subject-specific optimal hyperparameters would require a nested cross-fold validation, where the five instances of the 5-cv all required an optimisation routine on the training set to prevent overfitting on the test set. Optimal hyperparameters for this study were found using a different performance outcome than the tested performance outcome (subject-generic performance versus subject-specific performance), however the use of all data in the subject-generic optimisation routine may have introduced some form of overfitting as well. Contrarily, finding subject-generic hyperparameters could have also resulted in finding model parameters that did not perform well on subject-specific models, as less data was present for the subject-specific models compared with the subject-generic models.

Comparison of the non-weight-bearing tasks and the different ADLs showed that estimation of lower limb joint angles was best applicable to non-weight-bearing tasks, especially for the ankle joint. In non-weight-bearing tasks, unconstrained movements of the leg could be made that require activation of specific muscles. In weight-bearing activities, such as standing, muscle activations are, aside from changing joint orientations, required to prevent falling [50]. Hence, different forces need to be generated in both types of activities. As EMG gives a measure for the gener-

ated force, there is a more direct relationship between change in joint position and EMG in a non-weight-bearing task compared with a weight-bearing task.

Furthermore, in weight-bearing activities changes to both hip and knee joint angles occurred. The changes in both joint angles were made possible partially due to the contraction of the RF and BF, which are biarticular muscles (muscles that span two joints) [51]. Hence, EMG measured for these muscles were not necessarily directly related to a change in the knee joint angle, as muscle activity could also be required to act on the hip joint. Movement of the hip joint in the non-weight-bearing tasks was not necessary. Therefore, the measured muscle activity of the biarticular muscles had a more direct relation to changes in the knee joint angle in the non-weight-bearing tasks.

Comparison with state-of-the-art of the studied activities learned that different EMG features could be used. In this study, a single feature was used for the kernel-based models, the FC-NN and the linear method, so that similarity of the input data was obtained. The limited information provided by this feature possibly resulted in poorer model performance for the presented methods, as lower *RMSE* values were reported in the studies by Chen et al. [12] and Yang et al. [13]. Many different EMG features were developed, in time-domain, frequency-domain and time-frequency domain [52]. Different combinations of features were reported for estimation of joint angles [13, 14, 17], although four time-domain features proposed by Hudgins et al. [53] were used most frequently. Furthermore, features can be selected with techniques such as deep belief networks as shown by Chen et al. [12]. Finding an optimal feature set is not trivial, as many different combinations of features have to be evaluated. Therefore, a large research interest is seen in selecting the right set of EMG features [54]. This study showed that an algorithm capable of learning features from input data (CNN), outperformed the other studied models in most tested scenarios.

4.6 Recommendations

Based on the results of this study, several recommendations for future work can be outlined.

Comparison of this study with earlier work showed that in earlier work, lower *RMSE* values are reported, which could be due to the constructed EMG features. As many different EMG features exist, numerous combinations can be chosen. Future work can focus on extracting an optimal feature set to further explore and compare performance for the SVR, GP, KRR, LR and FC-NN models.

All studied datasets tested the different conditions per subject on a single day. As the EMG of a subject differs over time [36], robustness of a model over time could only be tested by adding artificial noise. Testing subjects on multiple days could give a better indication of the generalisability of a trained model. Such an analysis could give an indication in whether recalibration of a model is a requirement to maintain good

performance over time. Furthermore, such an analysis could further clarify whether different algorithms show different degrees of robustness in performance. Based on the presented results, no clear conclusion could be drawn on which of the good-performing algorithms shows best robustness to more noisy conditions.

In the current study, joint angles were estimated solely in the sagittal plane. In a non-weight-bearing situation, it can be interesting to assess the feasibility of estimation of ankle joint angles in other planes, so that for instance inversion/eversion of the feet can also be controlled.

Controlling the joint configuration of a prosthesis in a non-weight-bearing situation based on EMG signals was the goal of the presented approach. The ability to predict desired joint angles for an amputee in this situation could not be tested. Analysis of the performance in a non-weight-bearing situation showed promising results for able-bodied subjects. Furthermore, the analysis of level ground walking for amputees also showed promising performance when joint angles were estimated based on EMG of muscles in the stump. Therefore, it is rational to do a further analysis that tests the presented approach on amputees who are asked to perform non-weight-bearing activities. To collect data for this analysis, an amputee can be asked to perform several knee flexion/extension exercises with the unaffected leg and try to mimic the movements with the stump simultaneously (bilateral movements). In this data collection procedure, the assumption is made that the change in knee angle of the intact leg is similar to the knee angle change that would be made due to the activity in the stump. An alternative strategy to bilateral movements is the use of a visual stimulus as ground truth where the assumption is made that the subject mimics the visual stimulus. Visual stimuli were also applied in upper extremity prosthetic control tasks, for example in the publicly available Non Invasive Adaptive Prosthetics (Ninapro) database [55]. Data collection for the analysis of non-weight-bearing tasks in amputees should focus on the knee joint first. If this analysis shows promising results, testing could be extended towards amputees who have had TMR surgery. After TMR surgery, EMG from the (relocated) TA and GM muscles can be recorded. Additional research can be done to assess whether estimation of ankle joint angles shows promise, so that the amputee can regain complete control over the lower limb joints in a non-weight-bearing situation.

Joint angles on itself are not sufficient for the online control of a prosthesis. Estimated joint angles could be used as a target signal to generate a control command at the prosthesis level. Hargrove et al. [56] used a pattern recognition approach to classify knee flexion and extension, as well as ankle plantar- and dorsiflexion. From the recognised activity, a target angle was outputted by the pattern recognition system that was presented to an impedance controller which mapped the desired angle to a joint torque. Using the joint torque, the prosthesis could go to the required position. Comparison of the direct estimation of joint angles with a

pattern recognition based methodology should be carried out. In doing so, it could be assessed if amputees demonstrate a preference for either of the techniques.

Musculoskeletal modelling is another approach that shows great promise in providing control signals (joint moments) that could be presented to a prosthesis based on EMG, for both the upper and lower extremity [57, 58]. In the upper extremity it was possible to control a prosthesis by using EMG signals from the amputee’s residual limb in combination with joint angles of the prosthesis [57]. The combination of these inputs was mapped onto joint moments, using a physiologically correct musculoskeletal model. The prosthesis was then able to convert the joint moments into control commands, that subsequently changed the joint configuration of the prosthesis. As the joint moments solution space was physiologically constrained by the musculoskeletal model, good system robustness to unseen joint configurations or muscle activations was obtained [57]. Future work can focus on systematically comparing the different approaches on controlling a lower extremity prosthesis and identify which technique is most preferable for an amputee.

5 Conclusion

The main research question of this study has been to find the best method to estimate lower limb joint kinematics based on EMG for online control of a transfemoral prosthesis. To answer this research question, eight previously used machine learning methods (either in the upper and lower extremity) have been compared on their ability to predict lower limb joint angles from EMG for an individual subject. Hyperparameters for all models have been optimised on an activity-generic level so that optimal models could be compared. Across the three different datasets, the CNN model performs best most often for the different studied activities. However, the CNN model shows slightly worse robustness in performance in more noisy conditions. Predicting joint angles based on EMG is more accurate for non-weight-bearing activities compared to weight-bearing activities (level ground walking, stairs climbing). This is true for both the ankle and knee joint, although performance differences for the ankle joint are larger than for the knee joint. The addition of historic information from IMUs to the EMG shows an improvement in performance, even though simple features from the IMU sensors have been used. Future work can focus on using more extensive information from both the EMG and IMU signals to improve performance. Level ground walking for transfemoral amputees showed slightly better performance than level ground walking for able-bodied subjects, which warrants for further research into joint angle estimation based on EMG for transfemoral amputees in a non-weight-bearing situation.

References

- [1] Behrouz Fard, Pieter U. Dijkstra, Roy E. Stewart, and Jan H. B. Geertzen. Incidence rates of dysvascular lower extremity amputation changes in northern netherlands: A comparison of three cohorts of 1991-1992, 2003-2004 and 2012-2013. *PLOS ONE*, 13(9):e0204623, September 2018.
- [2] Richa Sinha and Wim J. A. Van Den Heuvel. A systematic literature review of quality of life in lower limb amputees. *Disability and Rehabilitation*, 33(11):883–899, January 2011.
- [3] Matthijs H. Spaan, Aline H. Vrieling, Pim van de Berg, Pieter U. Dijkstra, and Helco G. van Keeken. Predicting mobility outcome in lower limb amputees with motor ability tests used in early rehabilitation. *Prosthetics and Orthotics International*, 41(2):171–177, October 2016.
- [4] Michael Windrich, Martin Grimmer, Oliver Christ, Stephan Rinderknecht, and Philipp Beckerle. Active lower limb prosthetics: a systematic review of design issues and solutions. *BioMedical Engineering OnLine*, 15(S3), December 2016.
- [5] Christopher S. Crowe, Kate A. Impastato, Alex C. Donaghy, Caryn Earl, Janna L. Friedly, and Kari A. Keys. Prosthetic and orthotic options for lower extremity amputation and reconstruction. *Plastic and Aesthetic Research*, 2019, February 2019.
- [6] R. Carloni. MyLeg: smart and intuitive osseointegrated transfemoral prosthesis embodying advanced dynamic behaviours. myleg.eu. 2018, Accessed on: 13-03-2020.
- [7] Levi J. Hargrove, Laura A. Miller, Kristi Turner, and Todd A. Kuiken. Myoelectric pattern recognition outperforms direct control for transhumeral amputees with targeted muscle reinnervation: A randomized clinical trial. *Scientific Reports*, 7(1), October 2017.
- [8] Blair Hu, Elliott Rouse, and Levi Hargrove. Fusion of bilateral lower-limb neuromechanical signals improves prediction of locomotor activities. *Frontiers in Robotics and AI*, 5, June 2018.
- [9] Ning Jiang, S. Dosen, K-R Muller, and D. Farina. Myoelectric control of artificial limbs—is there a need to change focus? [in the spotlight]. *IEEE Signal Processing Magazine*, 29(5):152–150, September 2012.
- [10] E.C. Wentink, V.G.H. Schut, E.C. Prinsen, J.S. Rietman, and P.H. Veltink. Detection of the onset of gait initiation using kinematic sensors and EMG in transfemoral amputees. *Gait & Posture*, 39(1):391–396, January 2014.
- [11] Feng Zhang, Pengfeng Li, Zeng-Guang Hou, Zhen Lu, Yixiong Chen, Qingling Li, and Min Tan. sEMG-based continuous estimation of joint angles of human legs by using BP neural network. *Neurocomputing*, 78(1):139–148, February 2012.
- [12] Jiangcheng Chen, Xiaodong Zhang, Yu Cheng, and Ning Xi. Surface EMG based continuous estimation of human lower limb joint angles by using deep belief networks. *Biomedical Signal Processing and Control*, 40:335–342, February 2018.
- [13] Chen Yang, Xugang Xi, Sijia Chen, Seyed M. Miran, Xian Hua, and Zhizeng Luo. SEMG-based multifeatures and predictive model for knee-joint-angle estimation. *AIP Advances*, 9(9):095042, September 2019.

- [14] Aadeel Akhtar, Navid Aghasadeghi, Levi Hargrove, and Timothy Bretl. Estimation of distal arm joint angles from EMG and shoulder orientation for transhumeral prostheses. *Journal of Electromyography and Kinesiology*, 35:86–94, August 2017.
- [15] Ning Jiang, Johnny LG Vest-Nielsen, Silvia Muceli, and Dario Farina. EMG-based simultaneous and proportional estimation of wrist/hand kinematics in unilateral trans-radial amputees. *Journal of NeuroEngineering and Rehabilitation*, 9(1):42, 2012.
- [16] Jimson G Ngeo, Tomoya Tamei, and Tomohiro Shibata. Continuous and simultaneous estimation of finger kinematics using inputs from an EMG-to-muscle activation model. *Journal of NeuroEngineering and Rehabilitation*, 11(1):122, 2014.
- [17] 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 Transactions on Neural Systems and Rehabilitation Engineering*, 22(6):1198–1209, November 2014.
- [18] M. Xiloyannis, C. Gavriel, A. A. C. Thomik, and A. A. Faisal. Gaussian process autoregression for simultaneous proportional multi-modal prosthetic control with natural hand kinematics. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(10):1785–1801, Oct 2017.
- [19] J. M. Hahne, F. Biebmann, N. Jiang, H. Rehbaum, D. Farina, F. C. Meinecke, K.-R. Muller, and L. C. Parra. Linear and nonlinear regression techniques for simultaneous and proportional myoelectric control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(2):269–279, March 2014.
- [20] Tianzhe Bao, Ali Zaidi, Shengquan Xie, and Zhiqiang Zhang. Surface-EMG based wrist kinematics estimation using convolutional neural network. In *2019 IEEE 16th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. IEEE, May 2019.
- [21] Tianzhe Bao, Syed Ali Raza Zaidi, Shengquan Xie, Pengfei Yang, and Zhiqiang Zhang. A cnn-lstm hybrid framework for wrist kinematics estimation using surface electromyography, 2019. arXiv: 1912.00799 [eess.SP].
- [22] Blair Hu, Elliott Rouse, and Levi Hargrove. Benchmark datasets for bilateral lower-limb neuromechanical signals from wearable sensors during unassisted locomotion in able-bodied individuals. *Frontiers in Robotics and AI*, 5:14, 2018.
- [23] H.J. Hermens, B. Freriks, C. Disselhorst-Klug, and G. Rau. Development of recommendations for semg sensors and sensor placement procedures. *Journal of electromyography and kinesiology*, 10(5):361–374, 2000.
- [24] R.V. Schulte, E.C. Prinsen, L. Schaake, E.S. van Staveren, H.J. Hermens, and J.H. Buurke. Synchronization of wearable motion capture and EMG measurement systems. unpublished, submitted for review to *Sensors*.
- [25] J. Han, J. Pei, and M. Kamber. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2011.
- [26] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [27] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.
- [28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.
- [29] Kartik Chitturi and Peter Onyisi. How easily can neural networks learn relativity? *Journal of Physics: Conference Series*, 1085:042020, September 2018.
- [30] François Chollet et al. Keras. <https://keras.io>, 2015.
- [31] Jose Luis Rojo-Alvarez, Manel Martinez-Ramon, Jordi Munoz-Mari, and Gustau Camps-Valls. *Digital Signal Processing with Kernel Methods*. Wiley-IEEE Press, 1st edition, 2018.
- [32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [33] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS11, pages 2546–2554, Red Hook, NY, USA, 2011. Curran Associates Inc.
- [34] J. Bergstra, D. Yamins, and D. D. Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML13, pages I115–I123. JMLR.org, 2013.
- [35] Janne M. Hahne, Marko Markovic, and Dario Farina. User adaptation in myoelectric man-machine interfaces. *Scientific Reports*, 7(1), June 2017.
- [36] 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. *Science Robotics*, 3(19):eaat3630, June 2018.
- [37] Yuni Teh, Richard B. Woodward, and Levi J. Hargrove. Comparing the effects of signal noise on pattern recognition and linear regression-based myoelectric controllers. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, July 2018.
- [38] Cristina Roldán-Jiménez, Paul Bennett, and Antonio I. Cuesta-Vargas. Muscular activity and fatigue in lower-limb and trunk muscles during different sit-to-stand tests. *PLOS ONE*, 10(10):e0141675, October 2015.
- [39] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.
- [40] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay

- Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
- [41] Shinsuke Yoshioka, Akinori Nagano, Dean C. Hay, and Senshi Fukushima. The minimum required muscle force for a sit-to-stand task. *Journal of Biomechanics*, 45(4):699–705, February 2012.
- [42] Shinsuke Yoshioka, Akinori Nagano, Ryutaro Himeno, and Senshi Fukushima. Computation of the kinematics and the minimum peak joint moments of sit-to-stand movements. *BioMedical Engineering OnLine*, 6(1):26, 2007.
- [43] H.A. Varol, F. Sup, and M. Goldfarb. Powered sit-to-stand and assistive stand-to-sit framework for a powered transfemoral prosthesis. In *2009 IEEE International Conference on Rehabilitation Robotics*. IEEE, June 2009.
- [44] Fan Zhang, Ming Liu, and He Huang. Detection of critical errors of locomotion mode recognition for volitional control of powered transfemoral prostheses. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, August 2015.
- [45] Francesca Stival, Stefano Michieletto, Andrea De Agno, and Enrico Pagello. Toward a better robotic hand prosthesis control: Using EMG and IMU features for a subject independent multi joint regression model. In *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*. IEEE, August 2018.
- [46] Daniel Roetenberg, Henk Luinge, and Per Slycke. Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors. *Xsens Motion Technol. BV Tech. Rep.*, 3, 01 2009.
- [47] Isabelle Poitras, Frédérique Dupuis, Mathieu Biemann, Alexandre Campeau-Lecours, Catherine Mercier, Laurent Bouyer, and Jean-Sébastien Roy. Validity and reliability of wearable sensors for joint angle estimation: A systematic review. *Sensors*, 19(7):1555, March 2019.
- [48] Frank J. Wouda, Matteo Giuberti, Giovanni Bellusci, Erik Maartens, Jasper Reenalda, Bert-Jan F. van Beijnum, and Peter H. Veltink. Estimation of vertical ground reaction forces and sagittal knee kinematics during running using three inertial sensors. *Frontiers in Physiology*, 9:218, March 2018.
- [49] Samuel Farmer, Barbara Silver-Thorn, Philip Voglewede, and Scott A Beardsley. Within-socket myoelectric prediction of continuous ankle kinematics for control of a powered transtibial prosthesis. *Journal of Neural Engineering*, 11(5):056027, September 2014.
- [50] Kei Masani, Dimitry G. Sayenko, and Albert H. Vette. What triggers the continuous muscle activity during upright standing? *Gait & Posture*, 37(1):72–77, January 2013.
- [51] Christian Schumacher, Andrew Berry, Daniel Lemus, Christian Rode, André Seyfarth, and Heike Vallery. Biarticular muscles are most responsive to upper-body pitch perturbations in human standing. *Scientific Reports*, 9(1), October 2019.
- [52] Nurhazimah Nazmi, Mohd Abdul Rahman, Shin-Ichiroh Yamamoto, Siti Ahmad, Hairi Zamzuri, and Saiful Mazlan. A review of classification techniques of EMG signals during isotonic and isometric contractions. *Sensors*, 16(8):1304, August 2016.
- [53] B. Hudgins, P. Parker, and R.N. Scott. A new strategy for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering*, 40(1):82–94, 1993.
- [54] Angkoon Phinyomark and Erik Scheme. EMG pattern recognition in the era of big data and deep learning. *Big Data and Cognitive Computing*, 2(3):21, August 2018.
- [55] 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. *Scientific Data*, 1(1), December 2014.
- [56] Levi J. Hargrove, Ann M. Simon, Robert Lipschutz, Suzanne B. Finucane, and Todd A. Kuiken. Non-weight-bearing neural control of a powered transfemoral prosthesis. *Journal of NeuroEngineering and Rehabilitation*, 10(1):62, 2013.
- [57] Massimo Sartori, Guillaume Durandau, Strahinja Došen, and Dario Farina. Robust simultaneous myoelectric control of multiple degrees of freedom in wrist-hand prostheses by real-time neuromusculoskeletal modeling. *Journal of Neural Engineering*, 15(6):066026, October 2018.
- [58] Massimo Sartori, Monica Reggiani, Dario Farina, and David G. Lloyd. EMG-driven forward-dynamic estimation of muscle force and joint moment about multiple degrees of freedom in the human lower extremity. *PLoS ONE*, 7(12):e52618, December 2012.

Appendix A Hyperparameter search

This appendix presents an overview for the different hyperparameter searches per dataset. Section A.1 presents the hyperparameter search results for the EMG-only models in the ENABL3S dataset, section A.2 presents the hyperparameter search results for the (EMG+)IMU models in the ENABL3S dataset, section A.3 presents the hyperparameter search results for the MLK dataset and section A.4 presents the hyperparameter search results for the TIPS dataset.

A.1 ENABL3S EMG-only

This section shows the hyperparameter search results for the ENABL3S dataset. Four different optimisations were done for the four different activities: level ground walking (lgw), stair ascent (sa), stair descent (sd) and sit-to-stand transitions (sts).

SVR

Tables 3 and 4 show the optimised hyperparameters for the SVR model, including the search space and the used parameter per activity. Hyperparameters are settings in scikit-learn’s SVR model.

Table 3: Hyperparameter search for the SVR model for the ankle joint

Hyperparameter	Search space	lgw	sa	sd	sts
'kernel'	'rbf', 'polynomial', 'sigmoid', 'linear'	'rbf'	'rbf'	'rbf'	'rbf'
'C'	logarithmic space from 10^{-1} to 10^1 in 50 steps	2.223	1.842	1.048	0.409
' ϵ '	uniform space from 0.01 to 0.5 in steps of 0.01	0.08	0.10	0.03	0.02
'coef0'	uniform space from -5 to 5 in steps of 0.1	1.5	-2.5	1.7	1.3
'degree'	2, 3, 4	3	3	2	4

Table 4: Hyperparameter search for the SVR model for the knee joint

Hyperparameter	Search space	lgw	sa	sd	sts
'kernel'	'rbf', 'polynomial', 'sigmoid', 'linear'	'rbf'	'rbf'	'rbf'	'rbf'
'C'	logarithmic space from 10^{-1} to 10^1 in 50 steps	9.103	2.223	0.869	10
' ϵ '	uniform space from 0.01 to 0.5 in steps of 0.01	0.07	0.04	0.05	0.06
'coef0'	uniform space from -5 to 5 in steps of 0.1	4.7	0.9	-3.3	-5.0
'degree'	2, 3, 4	2	3	3	4

FC-NN

Table 5 shows the optimised hyperparameters for the FC-NN model, including the search space and the used parameter per activity.

Table 5: Hyperparameter search for the FC-NN model

Hyperparameter	Search space	lgw	sa	sd	sts
Number of hidden layers	1, 2, 3, 4, 5, 6	3	3	5	5
Neurons in hidden layer	uniform space from 10 to 500 in steps of 1	314	429	263	263
		272	427	397	397
		300	357	180	180
				183	186
				336	336
Dropout ratio	uniform space from 0.05 to 0.5	0.064	0.052	0.159	0.076
Optimiser	'RMSprop', 'ADAM', 'SGD'	'ADAM'	'ADAM'	'ADAM'	'ADAM'

CNN

Table 6 shows the optimised hyperparameters for the CNN model, including the search space and the used parameter per activity.

GP

Tables 7 and 8 show the optimised hyperparameters for the GP models (for ankle and knee joint respectively), including the search space and the used parameter per activity. Hyperparameters are settings in scikit-learn’s

Table 6: Hyperparameter search for the CNN model

Hyperparameter	Search space	lgw	sa	sd	sts
Number of convolutional layers	1, 2, 3, 4, 5	1	2	3	5
Neurons in convolutional layer	16, 32, 64, 128	32	128 64	128 128 64	64 64 128 32 128
Kernel size	3, 5, 7	5	3	3	3
Pool size	2, 3, 4	2	3	4	3
Number of hidden FC-layers	0, 1, 2, 3, 4	2	3	0	4
Neurons in FC-layer	uniform space from 10 to 500 in steps of 1	246 351	479 47 221	-	479 47 221 50
Dropout ratio	uniform space from 0.05 to 0.5	0.070	0.108	0.347	0.161
Optimiser	'RMSprop', 'ADAM', 'SGD'	'ADAM'	'ADAM'	'ADAM'	'RMSprop'

GaussianProcessRegressor model. The kernels are imported from scikit-learn's gaussian_process library. Various kernels can be chosen and combined. The parameters in the kernel are learned by the model, so only the choice for the kernel needs to be optimised.

Table 7: Hyperparameter search for the GP model for the ankle joint

Hyperparameter	Search space	lgw	sa	sd	sts
'kernel'	1) Matern() 2) ConstantKernel() * RBF() 3) ConstantKernel() * RBF() + WhiteKernel() 4) RBF() + WhiteKernel() 5) RBF()	3)	4)	4)	4)

Table 8: Hyperparameter search for the GP model for the knee joint

Hyperparameter	Search space	lgw	sa	sd	sts
'kernel'	1) Matern() 2) ConstantKernel() * RBF() 3) ConstantKernel() * RBF() + WhiteKernel() 4) RBF() + WhiteKernel() 5) RBF()	4)	4)	4)	4)

LSTM

Table 9 shows the optimised hyperparameters for the LSTM model, including the search space and the used parameter per activity.

Table 9: Hyperparameter search for the LSTM model

Hyperparameter	Search space	lgw	sa	sd	sts
Number of LSTM-neurons	uniform space from 20 to 60 in steps of 1	38	55	37	50
Number of hidden FC-layers	0, 1, 2, 3, 4	4	2	3	3
Neurons in FC-layer	uniform space from 10 to 500 in steps of 1	444 183 393 187	127 165	355 421 332	198 231 60
Dropout ratio	uniform space from 0.05 to 0.5	0.071	0.126	0.098	0.334
Optimiser	'RMSprop', 'ADAM', 'SGD'	'ADAM'	'RMSprop'	'ADAM'	'ADAM'

CNN-LSTM

Table 10 shows the optimised hyperparameters for the CNN-LSTM model, including the search space and the used parameter per activity.

Table 10: Hyperparameter search for the CNN-LSTM model

Hyperparameter	Search space	lgw	sa	sd	sts
Number of convolutional layers	1, 2, 3, 4, 5	3	1	2	2
Neurons in convolutional layer	16, 32, 64, 128	128 128 128	128	128 32	128 16
Kernel size	3, 5, 7	5	7	3	7
Pool size	2, 3, 4	3	4	4	4
Number of LSTM-neurons	uniform space from 20 to 60 in steps of 1	40	47	59	53
Number of hidden FC-layers	0, 1, 2, 3	1	3	2	3
Neurons in FC-layer	uniform space from 10 to 500 in steps of 1	448	350 330 417	72 352	74 319 320
Dropout ratio	uniform space from 0.05 to 0.5	0.305	0.123	0.067	0.099
Optimiser	'RMSprop', 'ADAM', 'SGD'	'ADAM'	'ADAM'	'ADAM'	'ADAM'

KRR

Tables 11 and 12 show the optimised hyperparameters for the KRR model (for the ankle and knee joint respectively), including the search space and the used parameter per activity. Hyperparameters are settings in scikit-learn's KernelRidge model.

Table 11: Hyperparameter search for the KRR model for the ankle joint

Hyperparameter	Search space	lgw	sa	sd	sts
'kernel'	'rbf', 'poly', 'linear', 'sigmoid'	'poly'	'poly'	'poly'	'rbf'
'alpha'	uniform space from 0.02 to 5 in steps of 0.01	3.91	0.48	0.19	0.05
'coef0'	uniform space from -5 to 5 in steps of 0.1	2.4	-4.7	-3.1	3.0
'degree'	2, 3, 4	4	4	4	3

Table 12: Hyperparameter search for the KRR model for the knee joint

Hyperparameter	Search space	lgw	sa	sd	sts
'kernel'	'rbf', 'poly', 'linear', 'sigmoid'	'poly'	'poly'	'poly'	'poly'
'alpha'	uniform space from 0.02 to 5 in steps of 0.01	3.91	0.37	0.24	0.97
'coef0'	uniform space from -5 to 5 in steps of 0.1	4.6	4.2	-1.6	3.5
'degree'	2, 3, 4	4	4	4	4

A.2 ENABL3S IMU

The statistical analysis of the EMG-only results showed that the CNN is also tested with IMU and EMG+IMU input. These models were also optimised, which are shown in the tables below. For sd and sts, the FC-NN, GP and SVR were also tested. All optimisation results are shown in this section.

SVR

Tables 13 and 14 show the optimised hyperparameters for the SVR in the EMG+IMU situation. Tables 15 and 16 show the optimised hyperparameters for the SVR in the IMU-only situation. The search space and the used parameter per activity are listed.

Table 13: Hyperparameter search for the SVR model for the ankle joint in the EMG+IMU situation

Hyperparameter	Search space	lgw	sa	sd	sts
'kernel'	'rbf', 'polynomial', 'sigmoid', 'linear'	-	-	'rbf'	'rbf'
'C'	logarithmic space from 10^{-1} to 10^1 in 50 steps	-	-	1.048	0.720
'epsilon'	uniform space from 0.01 to 0.5 in steps of 0.01	-	-	0.03	0.03
'coef0'	uniform space from -5 to 5 in steps of 0.1	-	-	-1.4	-0.7
'degree'	2, 3, 4	-	-	3	2

FC-NN

Tables 17 and 18 show the optimised hyperparameters for the FC-NN models (for the EMG+IMU and IMU-only situation respectively), including the search space and the used parameter per activity.

Table 14: Hyperparameter search for the SVR model for the knee joint in the EMG+IMU situation

Hyperparameter	Search space	lgw	sa	sd	sts
'kernel'	'rbf', 'polynomial', 'sigmoid', 'linear'	-	-	'rbf'	'rbf'
'C'	logarithmic space from 10^{-1} to 10^1 in 50 steps	-	-	2.024	7.543
' ϵ '	uniform space from 0.01 to 0.5 in steps of 0.01	-	-	0.01	0.03
'coef0'	uniform space from -5 to 5 in steps of 0.1	-	-	3.1	-3.0
'degree'	2, 3, 4	-	-	2	3

Table 15: Hyperparameter search for the SVR model for the ankle joint in the IMU-only situation

Hyperparameter	Search space	lgw	sa	sd	sts
'kernel'	'rbf', 'polynomial', 'sigmoid', 'linear'	-	-	'rbf'	'rbf'
'C'	logarithmic space from 10^{-1} to 10^1 in 50 steps	-	-	1.048	6.866
' ϵ '	uniform space from 0.01 to 0.5 in steps of 0.01	-	-	0.03	0.16
'coef0'	uniform space from -5 to 5 in steps of 0.1	-	-	2.6	-3.6
'degree'	2, 3, 4	-	-	4	3

Table 16: Hyperparameter search for the SVR model for the knee joint in the IMU-only situation

Hyperparameter	Search space	lgw	sa	sd	sts
'kernel'	'rbf', 'polynomial', 'sigmoid', 'linear'	-	-	'rbf'	'rbf'
'C'	logarithmic space from 10^{-1} to 10^1 in 50 steps	-	-	4.715	7.543
' ϵ '	uniform space from 0.01 to 0.5 in steps of 0.01	-	-	0.04	0.21
'coef0'	uniform space from -5 to 5 in steps of 0.1	-	-	1.9	2.8
'degree'	2, 3, 4	-	-	4	3

Table 17: Hyperparameter search for the FC-NN model, for the EMG+IMU situation

Hyperparameter	Search space	lgw	sa	sd	sts
Number of hidden layers	1, 2, 3, 4, 5, 6	-	-	6	4
Neurons in hidden layer	uniform space from 10 to 500 in steps of 1	-	-	246 363 490 215 150 80	468 460 495 117
Dropout ratio	uniform space from 0.05 to 0.5	-	-	0.050	0.090
Optimiser	'RMSprop', 'ADAM', 'SGD'	-	-	'ADAM'	'RMSprop'

Table 18: Hyperparameter search for the FC-NN model, for the IMU-only situation

Hyperparameter	Search space	lgw	sa	sd	sts
Number of hidden layers	1, 2, 3, 4, 5, 6	-	-	4	4
Neurons in hidden layer	uniform space from 10 to 500 in steps of 1	-	-	156 233 348 81	463 372 38 157
Dropout ratio	uniform space from 0.05 to 0.5	-	-	0.175	0.325
Optimiser	'RMSprop', 'ADAM', 'SGD'	-	-	'ADAM'	'RMSprop'

CNN

Tables 19 and 20 show the optimised hyperparameters for the CNN models (for the EMG+IMU and IMU-only situation respectively), including the search space and the used parameter per activity.

GP

Tables 21 and 22 show the optimised hyperparameters for the GP in the EMG+IMU situation. Tables 23 and 24 show the optimised hyperparameters for the GP in the IMU-only situation. Hyperparameters are settings in scikit-learn's GaussianProcessRegressor model. The kernels are imported from scikit-learn's gaussian_process library. Various kernels can be chosen and combined. The parameters in the kernel are learned by the model, so only the choice for the kernel needs to be optimised. .

Table 19: Hyperparameter search for the CNN model, for the EMG+IMU situation

Hyperparameter	Search space	lgw	sa	sd	sts
Number of convolutional layers	1, 2, 3, 4, 5	3	3	3	1
Neurons in convolutional layer	16, 32, 64, 128	128 128 16	128 128 64	128 128 16	32
Kernel size	3, 5, 7	3	3	3	5
Pool size	2, 3, 4	4	4	4	2
Number of hidden FC-layers	0, 1, 2, 3, 4	0	0	2	2
Neurons in FC-layer	uniform space from 10 to 500 in steps of 1	-	-	293 293	127 264
Dropout ratio	uniform space from 0.05 to 0.5	0.121	0.398	0.101	0.155
Optimiser	'RMSprop', 'ADAM', 'SGD'	'ADAM'	'ADAM'	'ADAM'	'RMSprop'

Table 20: Hyperparameter search for the CNN model, for the IMU-only situation

Hyperparameter	Search space	lgw	sa	sd	sts
Number of convolutional layers	1, 2, 3, 4, 5	1	2	1	1
Neurons in convolutional layer	16, 32, 64, 128	128	16 32	32	128
Kernel size	3, 5, 7	3	3	7	7
Pool size	2, 3, 4	3	4	4	2
Number of hidden FC-layers	0, 1, 2, 3, 4	3	2	4	4
Neurons in FC-layer	uniform space from 10 to 500 in steps of 1	487 351 152	246 263	469 183 179 308	487 364 261 386
Dropout ratio	uniform space from 0.05 to 0.5	0.057	0.135	0.218	0.112
Optimiser	'RMSprop', 'ADAM', 'SGD'	'ADAM'	'ADAM'	'ADAM'	'RMSprop'

Table 21: Hyperparameter search for the GP model for the ankle joint in the EMG+IMU situation

Hyperparameter	Search space	lgw	sa	sd	sts
'kernel'	1) Matern() 2) ConstantKernel() * RBF() 3) ConstantKernel() * RBF() + WhiteKernel() 4) RBF() + WhiteKernel() 5) RBF()	-	-	3)	4)

Table 22: Hyperparameter search for the GP model for the knee joint in the EMG+IMU situation

Hyperparameter	Search space	lgw	sa	sd	sts
'kernel'	1) Matern() 2) ConstantKernel() * RBF() 3) ConstantKernel() * RBF() + WhiteKernel() 4) RBF() + WhiteKernel() 5) RBF()	-	-	1)	1)

Table 23: Hyperparameter search for the GP model for the ankle joint in the IMU-only situation

Hyperparameter	Search space	lgw	sa	sd	sts
'kernel'	1) Matern() 2) ConstantKernel() * RBF() 3) ConstantKernel() * RBF() + WhiteKernel() 4) RBF() + WhiteKernel() 5) RBF()	-	-	3)	3)

Table 24: Hyperparameter search for the GP model for the knee joint in the IMU-only situation

Hyperparameter	Search space	lgw	sa	sd	sts
'kernel'	1) Matern() 2) ConstantKernel() * RBF() 3) ConstantKernel() * RBF() + WhiteKernel() 4) RBF() + WhiteKernel() 5) RBF()	-	-	3)	3)

A.3 MLK

This section shows the hyperparameter search results for the MLK dataset. Three different optimisations were done for the following activities: sit-to-stand transitions (sts), combined non-weight-bearing activities (nwb)

and a combination of the two activities (nwb+sts).

SVR

Tables 25 and 26 show the optimised hyperparameters for the SVR model, including the search space and the used parameter per activity. Hyperparameters are settings in scikit-learn’s SVR model.

Table 25: Hyperparameter search for the SVR model for the ankle joint in the MLK dataset

Hyperparameter	Search space	sts	nwb	nwb+sts
'kernel'	'rbf', 'polynomial', 'sigmoid', 'linear'	'rbf'	'rbf'	'rbf'
'C'	logarithmic space from 10^{-1} to 10^1 in 50 steps	0.160	7.543	7.543
' ϵ '	uniform space from 0.01 to 0.5 in steps of 0.01	0.01	0.10	0.10
'coef0'	uniform space from -5 to 5 in steps of 0.1	-2.6	-0.5	-4.9
'degree'	2, 3, 4	3	3	4

Table 26: Hyperparameter search for the SVR model for the knee joint in the MLK dataset

Hyperparameter	Search space	sts	nwb	nwb+sts
'kernel'	'rbf', 'polynomial', 'sigmoid', 'linear'	'rbf'	'rbf'	'rbf'
'C'	logarithmic space from 10^{-1} to 10^1 in 50 steps	6.251	1.842	7.543
' ϵ '	uniform space from 0.01 to 0.5 in steps of 0.01	0.03	0.01	0.07
'coef0'	uniform space from -5 to 5 in steps of 0.1	2.4	-3.3	1.9
'degree'	2, 3, 4	3	2	2

FC-NN

Table 27 shows the optimised hyperparameters for the FC-NN model, including the search space and the used parameter per activity.

Table 27: Hyperparameter search for the FC-NN model in the MLK dataset

Hyperparameter	Search space	sts	nwb	nwb+sts
Number of hidden layers	1, 2, 3, 4, 5, 6	5	4	4
Neurons in hidden layer	uniform space from 10 to 500 in steps of 1	239 239 127 264 179	314 272 300 302	342 359 383 468
Dropout ratio	uniform space from 0.05 to 0.5	0.108	0.053	0.0148
Optimiser	'RMSprop', 'ADAM', 'SGD'	'ADAM'	'ADAM'	'ADAM'

CNN

Table 28 shows the optimised hyperparameters for the CNN model, including the search space and the used parameter per activity.

Table 28: Hyperparameter search for the CNN model in the MLK dataset

Hyperparameter	Search space	sts	nwb	nwb+sts
Number of convolutional layers	1, 2, 3, 4, 5	3	3	2
Neurons in convolutional layer	16, 32, 64, 128	64 64 128	128 64 128	128 16
Kernel size	3, 5, 7	5	3	7
Pool size	2, 3, 4	2	4	4
Number of hidden FC-layers	0, 1, 2, 3, 4	2	2	3
Neurons in FC-layer	uniform space from 10 to 500 in steps of 1	156 476	499 491	246 351 152
Dropout ratio	uniform space from 0.05 to 0.5	0.117	0.391	0.120
Optimiser	'RMSprop', 'ADAM', 'SGD'	'ADAM'	'ADAM'	'ADAM'

GP

Tables 29 and 30 show the optimised hyperparameters for the GP model, including the search space and the used parameter per activity. Hyperparameters are settings in scikit-learn’s GaussianProcessRegressor model. The kernels are imported from scikit-learn’s gaussian_process library. Various kernels can be chosen and combined. The parameters in the kernel are learned by the model, so only the choice for the kernel needs to be optimised.

Table 29: Hyperparameter search for the GP model for the ankle joint in the MLK dataset

Hyperparameter	Search space	sts	nwb	nwb+sts
'kernel'	1) Matern() 2) ConstantKernel() * RBF() 3) ConstantKernel() * RBF() + WhiteKernel() 4) RBF() + WhiteKernel() 5) RBF()	3)	4)	4)

Table 30: Hyperparameter search for the GP model for the knee joint in the MLK dataset

Hyperparameter	Search space	sts	nwb	nwb+sts
'kernel'	1) Matern() 2) ConstantKernel() * RBF() 3) ConstantKernel() * RBF() + WhiteKernel() 4) RBF() + WhiteKernel() 5) RBF()	4)	4)	4)

LSTM

Table 31 shows the optimised hyperparameters for the LSTM model, including the search space and the used parameter per activity.

Table 31: Hyperparameter search for the LSTM model in the MLK dataset

Hyperparameter	Search space	sts	nwb	nwb+sts
Number of LSTM-neurons	uniform space from 20 to 60 in steps of 1	29	42	59
Number of hidden FC-layers	0, 1, 2, 3, 4	4	0	2
Neurons in FC-layer	uniform space from 10 to 500 in steps of 1	426 421 473 377	-	454 215
Dropout ratio	uniform space from 0.05 to 0.5	0.109	0.191	0.075
Optimiser	'RMSprop', 'ADAM', 'SGD'	'ADAM'	'ADAM'	'ADAM'

CNN-LSTM

Table 32 shows the optimised hyperparameters for the CNN-LSTM model, including the search space and the used parameter per activity.

Table 32: Hyperparameter search for the CNN-LSTM model in the MLK dataset

Hyperparameter	Search space	sts	nwb	nwb+sts
Number of convolutional layers	1, 2, 3, 4, 5	3	3	3
Neurons in convolutional layer	16, 32, 64, 128	128 64 128	32 64 16	64 128 16
Kernel size	3, 5, 7	3	5	7
Pool size	2, 3, 4	3	4	4
Number of LSTM-neurons	uniform space from 20 to 60 in steps of 1	31	24	40
Number of hidden FC-layers	0, 1, 2, 3	0	1	0
Neurons in FC-layer	uniform space from 10 to 500 in steps of 1	-	381	-
Dropout ratio	uniform space from 0.05 to 0.5	0.172	0.051	0.317
Optimiser	'RMSprop', 'ADAM', 'SGD'	'ADAM'	'RMSprop'	'ADAM'

KRR

Tables 33 and 34 show the optimised hyperparameters for the KRR model, including the search space and the used parameter per activity. Hyperparameters are settings in scikit-learn’s KernelRidge model.

Table 33: Hyperparameter search for the KRR model for the ankle joint in the MLK dataset

Hyperparameter	Search space	sts	nwb	nwb+sts
'kernel'	'rbf', 'poly', 'linear', 'sigmoid'	'poly'	'rbf'	'poly'
'alpha'	uniform space from 0.02 to 5 in steps of 0.01	0.42	0.02	0.26
'coef0'	uniform space from -5 to 5 in steps of 0.1	3.8	-1.0	4.6
'degree'	2, 3, 4	4	2	4

Table 34: Hyperparameter search for the KRR model for the knee joint in the MLK dataset

Hyperparameter	Search space	sts	nwb	nwb+sts
'kernel'	'rbf', 'poly', 'linear', 'sigmoid'	'poly'	'poly'	'poly'
'alpha'	uniform space from 0.02 to 5 in steps of 0.01	0.04	0.12	0.64
'coef0'	uniform space from -5 to 5 in steps of 0.1	1.6	3.4	4.6
'degree'	2, 3, 4	3	4	4

A.4 TIPS

This section shows the hyperparameter search results for the TIPS dataset. Five different optimisations were done for all different prosthetic knees. The 3R20 corresponds to P1, the 3R106 corresponds to P2, the 3R60 corresponds to P3, the Mauch SNS corresponds to P4 and the Adaptive Knee corresponds to P5.

SVR

Table 35 a shows the optimised hyperparameters for the SVR model, including the search space and the used parameter per activity. Hyperparameters are settings in scikit-learn’s SVR model.

Table 35: Hyperparameter search for the SVR model in the TIPS dataset

Hyperparameter	Search space	P1	P2	P3	P4	P5
'kernel'	'rbf', 'polynomial', 'sigmoid', 'linear'	'rbf'	'rbf'	'rbf'	'rbf'	'rbf'
'C'	logarithmic space from 10^{-1} to 10^1 in 50 steps	7.543	7.543	2.683	3.237	2.223
'epsilon'	uniform space from 0.01 to 0.5 in steps of 0.01	0.02	0.03	0.07	0.07	0.04
'coef0'	uniform space from -5 to 5 in steps of 0.1	3.4	-3.2	0.3	-2.1	0.3
'degree'	2, 3, 4	4	4	4	4	3

FC-NN

Table 36 shows the optimised hyperparameters for the FC-NN model, including the search space and the used parameter per activity.

Table 36: Hyperparameter search for the FC-NN model in the TIPS dataset

Hyperparameter	Search space	P1	P2	P3	P4	P5
Number of hidden layers	1, 2, 3, 4, 5, 6	4	6	4	3	5
Neurons in hidden layer	uniform space from 10 to 500 in steps of 1	427 84 72 412	473 64 360 121 457 109	268 386 79 327	429 427 357	135 444 237 402 343
Dropout ratio	uniform space from 0.05 to 0.5	0.152	0.055	0.084	0.058	0.059
Optimiser	'RMSprop', 'ADAM', 'SGD'	'ADAM'	'ADAM'	'ADAM'	'ADAM'	'ADAM'

CNN

Table 37 shows the optimised hyperparameters for the CNN model, including the search space and the used parameter per activity.

Table 37: Hyperparameter search for the CNN model in the TIPS dataset

Hyperparameter	Search space	P1	P2	P3	P4	P5
Number of convolutional layers	1, 2, 3, 4, 5	3	3	2	3	2
Neurons in convolutional layer	16, 32, 64, 128	64 64 128	128 32 32	128 32	32 32 32	64 64
Kernel size	3, 5, 7	5	3	7	3	3
Pool size	2, 3, 4	2	2	4	4	2
Number of hidden FC-layers	0, 1, 2, 3, 4	1	2	3	2	4
Neurons in FC-layer	uniform space from 10 to 500 in steps of 1	152	395 186	355 186 446	452 364	74 352 310 454
Dropout ratio	uniform space from 0.05 to 0.5	0.203	0.172	0.105	0.355	0.075
Optimiser	'RMSprop', 'ADAM', 'SGD'	'RMSprop'	'ADAM'	'ADAM'	'ADAM'	'ADAM'

GP

Table 38 shows the optimised hyperparameters for the GP model, including the search space and the used parameter per activity. Hyperparameters are settings in scikit-learn's GaussianProcessRegressor model. The kernels are imported from scikit-learn's gaussian_process library. Various kernels can be chosen and combined. The parameters in the kernel are learned by the model, so only the choice for the kernel needs to be optimised.

Table 38: Hyperparameter search for the GP model in the TIPS dataset

Hyperparameter	Search space	P1	P2	P3	P4	P5
'kernel'	1) Matern() 2) ConstantKernel() * RBF() 3) ConstantKernel() * RBF() + WhiteKernel() 4) RBF() + WhiteKernel() 5) RBF()	4)	4)	4)	4)	4)

LSTM

Table 39 shows the optimised hyperparameters for the LSTM model, including the search space and the used parameter per activity.

Table 39: Hyperparameter search for the LSTM model in the TIPS dataset

Hyperparameter	Search space	P1	P2	P3	P4	P5
Number of LSTM-neurons	uniform space from 20 to 60 in steps of 1	52	59	34	58	43
Number of hidden FC-layers	0, 1, 2, 3, 4	3	4	3	2	4
Neurons in FC-layer	uniform space from 10 to 500 in steps of 1	161 103 473	249 231 60 180	340 77 196	482 264	351 112 190 206
Dropout ratio	uniform space from 0.05 to 0.5	0.227	0.055	0.155	0.178	0.051
Optimiser	'RMSprop', 'ADAM', 'SGD'	'RMSprop'	'ADAM'	'RMSprop'	'ADAM'	'ADAM'

CNN-LSTM

Table 40 shows the optimised hyperparameters for the CNN-LSTM model, including the search space and the used parameter per activity.

KRR

Table 41 shows the optimised hyperparameters for the KRR model, including the search space and the used parameter per activity. Hyperparameters are settings in scikit-learn's KernelRidge model.

Table 40: Hyperparameter search for the CNN-LSTM model in the TIPS dataset

Hyperparameter	Search space	P1	P2	P3	P4	P5
Number of convolutional layers	1, 2, 3, 4, 5	2	3	3	3	3
Neurons in convolutional layer	16, 32, 64, 128	16 64	64 64 64	64 32 64	64 128 64	64 64 64
Kernel size	3, 5, 7	3	3	3	7	3
Pool size	2, 3, 4	3	4	4	3	4
Number of LSTM-neurons	uniform space from 20 to 60 in steps of 1	50	54	54	27	54
Number of hidden FC-layers	0, 1, 2, 3	1	0	2	1	0
Neurons in FC-layer	uniform space from 10 to 500 in steps of 1	131	-	452 122	482	-
Dropout ratio	uniform space from 0.05 to 0.5	0.349	0.137	0.079	0.438	0.265
Optimiser	'RMSprop', 'ADAM', 'SGD'	'RMSprop'	'ADAM'	'ADAM'	'ADAM'	'ADAM'

Table 41: Hyperparameter search for the KRR model in the TIPS dataset

Hyperparameter	Search space	P1	P2	P3	P4	P5
'kernel'	'rbf', 'poly', 'linear', 'sigmoid'	'poly'	'poly'	'poly'	'poly'	'poly'
'alpha'	uniform space from 0.02 to 5 in steps of 0.01	0.50	0.02	0.28	0.03	0.54
'coef0'	uniform space from -5 to 5 in steps of 0.1	4.4	2.3	3.2	2.2	3.3
'degree'	2, 3, 4	4	4	4	3	4