Comparing Machine Learning Methods for Force Myography Based Estimation of Isokinetic Knee and Ankle Torques

Tim Wolk*, Charlotte Marquardt*, Miha Dežman and Tamim Asfour

***Abstract*— Wearable sensors enable accurate estimation of joint moments through easy-to-use myography-based methods, such as force myography (FMG), offering practical benefits and valuable insights into continuous muscle state estimation to enhance control strategies. This paper presents a comparative analysis of four commonly used machine learning methods, Gaussian process regression (GPR), support vector regression (SVR), feed-forward neural network (FFNN), and temporal convolutional network (TCN), for estimation of human knee and ankle joint torques based on joint angles, velocities, and FMG signals from eight muscles on the human leg. The performance of the methods was evaluated on isokinetic motions of ten participants and compared to the models enhanced by electromyography (EMG) signals. Among the evaluated models, neural networks consistently demonstrated the highest accuracy in both inter- and intra-participant validations. Incorporating FMG modality yielded comparable performance to EMG-based estimation for unknown participants. Additionally, FMG outperforms EMG-based estimation in novel task characteristics within a single participant. These findings demonstrate the potential of FMG as a viable alternative to EMG for human joint torque estimation and highlight its potential for personalized exoskeleton control.**

I. INTRODUCTION

Accurate estimation of human joint moments using wearable sensors provides valuable insights for exoskeleton control during real-world activities [1]. Various sensors and approaches can estimate joint torque, typically by usage of data-driven methods or neuromusculoskeletal (NMS) models. Joint angle encoders and inertial measurement units (IMUs) provide insights into movement kinematics and are often combined with additional sensors like instrumented insoles measuring ground reaction forces (GRFs) to estimate human joint torques [2]–[4]. However, there is an increasing interest in integrating muscle-level biomechanics to enable a more effective human-exoskeleton interaction. This integration may personalize control mechanisms and eliminate the need for manual adjustments of control parameters for each user [3]. Additionally, it may reduce the mental load of exoskeleton users during its operation [5].

Myography-based methods have emerged as a promising approach for extracting information about continuous muscle states for exoskeleton control [3], [6]. When muscles are activated, the muscle fibers are electrically stimulated, leading to muscle contraction and changes in the muscle's shape. Electromyography (EMG) is a widely recognized method for capturing the electrical signals of muscle activity [7]–[11]. However, obtaining high-quality EMG signals requires extensive filtering and signal post-processing. Several factors negatively impact signal quality, including electrode positioning on the muscle, skin contact quality with the electrode, and electrode displacement during muscle contractions [12]. In contrast, force myography (FMG) measures the mechanical phenomena associated with muscle contraction by detecting the normal forces resulting from changes in muscle volume shape, rather than electrical activity. This means that FMG does not require direct skin contact, precise sensor placement on the muscle, or complex post-processing [13]–[15].

Previous research has extensively explored using data-driven models to estimate joint torque from EMG signals [16]–[20]. Research shows that various data-driven methods can effectively estimate isometric joint torque [10]. Multiple models such as Gaussian process regression (GPR) [21], support vector regression (SVR) [22], [23], long short-term memory (LSTM) [21], artificial neural network (ANN) [24] and convolutional neural network (CNN) [25], [26], as well as temporal convolutional network (TCN) [1], [27] demonstrated successful estimation of human joint torques by combining kinematic information with EMG-based muscular information. Previous comparative studies have shown that CNNs outperform NMSs models [25] and LSTMs [26] in EMG-based lower limb joint torque estimation. However, EMG signals exhibit variability over days,

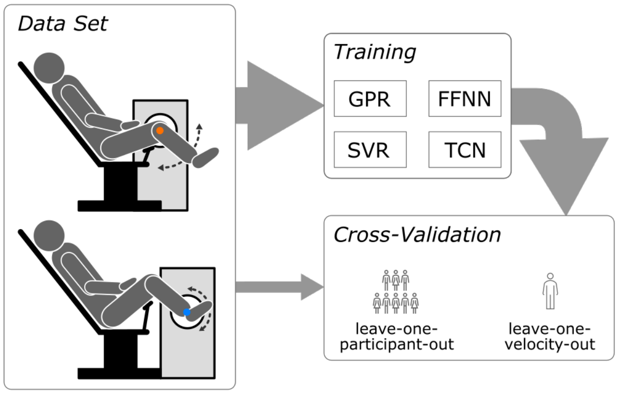


Fig. 1: Overview of the training and validation of the four torque estimation models.

*The first two authors contributed equally to this work.

This work has been supported by the Carl Zeiss Foundation through the JuBot project, the European Union's Horizon Europe Widening Program through the HERON project, and the German Federal Ministry of Research, Technology, and Space (BMFTR) under the Robotics Institute Germany (RIG).. The authors are with the High Performance Humanoid Technologies Lab, Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany.
Corresponding authors: {charlotte.marquardt,asfour}@kit.edu

impacting accuracy [10], [25], [28]. FMG-based attempts have primarily focused on motion classification tasks [28]–[30].

Although there have been only a few investigations of FMG-based torque estimation, GPR has been previously introduced for lower limb joint torque estimation and demonstrated that FMG outperforms EMG for single-participant torque estimation but not in inter-participant performance [31].

This paper presents a comparative analysis of four commonly used machine learning methods to estimate human knee and ankle joint torques using FMG signals, along with joint angle and velocity data. We compare the performance of these existing models using a data set including ten participants performing isokinetic knee and ankle joint motions. Additionally, we assess the same four models with EMG signals replacing the FMG signals for comparison. By examining these models, we aim to deepen our understanding of the potential of FMG-based joint torque estimation and its applicability in various approaches toward achieving continuous control using FMG. The insights gained from this analysis not only enhance existing knowledge about FMG and EMG but also set the stage for future research to optimize exoskeleton performance through tailored control mechanisms.

The paper is organized as follows. Section II describes the models and the training and validation methods. The quality of the torque estimation and its validation results are presented in Section III and discussed in Section IV. Section V concludes the paper.

II. METHODS

Creating accurate biomechanical models of joint movement and muscle activity is a complex task [3]. Data-driven regression models offer a promising alternative. We compare four estimation models to learn the relationship between joint angle, joint velocity, and muscle activity. This section outlines each model and the data and methods used for training and validation.

A. Gaussian Process Regression

GPR models are a probabilistic and parametric supervised learning method based on a Gaussian distribution [32]. Mathematically, this can be expressed by

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{1}$$

where an observed outcome $f(\mathbf{x})$ is estimated from an input $\mathbf{x}$ by a Gaussian process with the mean function m and the covariance function $k(\mathbf{x}, \mathbf{x}')$. Here, a radial basis function (RBF)

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp(-\frac{|\mathbf{x} - \mathbf{x}'|^2}{2l^2}), \tag{2}$$

with the hyperparameters length scale l and error variance σ^2, was chosen as a covariance function. The mean function is initially set to zero. These hyperparameters and the mean function are optimized, maximizing the log marginal likelihood and minimizing the validation loss. Similar to [31], a sparse gaussian process regression (SGPR) was used to train the model due to the computational complexity of $O(n^3)$, as well as the storage complexity of $O(n^2)$ in a traditional GPR. SGPR is a modification of GPR, which uses a subset consisting of k training datapoints as inducing points to reduce the computational complexity to $O(nk^2)$ and the storage complexity to $O(k^2)$. It was implemented using Python's *gpflow* library [33].

B. Support Vector Regression

As a supervised learning method, SVR estimates a continuous multivariate function and trains it using a loss function that equally penalizes both over- and underestimations [34]. It is formulated as an optimization problem attempting to minimize a convex ε-insensitive loss function, while minimizing the estimation error. The true output $f(\mathbf{x})$ can be approximated by

$$f(\mathbf{x}) \sim \sum_{i=1}^{N} (a_i^* - a_i) k(\mathbf{x}, \mathbf{x}') \tag{3}$$

with the input $\mathbf{x}$, the Lagrange multipliers a_i^*, a_i and the kernel $k(\mathbf{x}, \mathbf{x}')$. Similar to the GPR model, a RBF-kernel

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma |\mathbf{x} - \mathbf{x}'|^2) \tag{4}$$

with the free parameter γ was chosen. The threshold ε was set to 0.1. To optimize the parameters, the corresponding loss function is minimized based on the Karush-Kuhn-Tucker conditions. Due to the computational complexity of SVR, a subset of the total training data was used to train the model. The implementation was based on Python's *scikit-learn* library [35].

C. Feed Forward Neural Network

A straightforward approach to neural networks is the feed-forward neural network (FFNN) (also known as ANN), which consists of interconnected layers of artificial neurons that resemble biological neurons in both shape and function [36]. For an input vector $\mathbf{x}$ and a single hidden layer with seven neurons, $\mathbf{W}$ represents the matrix of the weights connecting each input with each neuron of the layer:

$$f(\mathbf{x}) \sim \theta(\mathbf{W}\mathbf{x} + \mathbf{b}). \tag{5}$$

The desired output $f(\mathbf{x})$ is derived with a bias $\mathbf{b}$ from each neuron. A rectified linear unit (ReLU) is chosen as the activation function. The bias and the weights are optimized via a backpropagation algorithm, minimizing the mean squared error. The architecture was inspired by [28], and implemented using Python's *tensorflow* library [37].

D. Temporal Convolutional Network

While CNNs are a common deep learning method, often used in vision tasks, TCNs extend the concept of convolutions into sequence modeling and causal, dilated convolutions to capture only past information, but also more long-range dependencies more efficiently [38]. The proposed TCN consists of three residual blocks, each containing two one-dimensional convolution layers with 50 filters, a kernel size of four, and dilation rates increasing across the residual blocks as powers of two, followed by a weight normalization and ReLU activation. The output layer consists of a convolutional layer with a kernel size of one and a single filter.

The structure of the TCN was simplified from [1] and implemented using Python's *tensorflow* library [37].

E. Data Set

The data used for the estimation consists of unilateral data of ten participants (m = 5 | f = 5, age 26.8 ± 3.2 years, height 175.2 ± 6.78 cm, weight 65.0 ± 6.8 kg) performing isokinetic sagittal knee and ankle joint motion (Fig. 2), acquired in a previous study [39]. The data includes multiple swings at four angular velocities (ankle: 30 °/s, 60 °/s, 90 °/s, 120 °/s | knee: 60 °/s, 90 °/s, 120 °/s, 150 °/s) carried out on an IsoMed 2000 device, which ensures a constant angular joint velocity by dynamically adjusting the resistance throughout the range of motion, resulting in variable joint torque. This provides a controlled experimental environment with minimal disturbances or external forces acting on the sensors during the motion while allowing for on-axis measurement of the joint angle and the true joint torque.

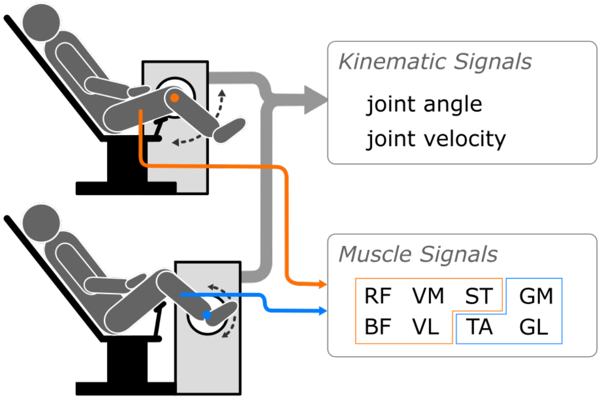


Fig. 2: Overview of the data acquisition process.

Muscle activity data from eight FMG units (design described in [15]) and EMG electrode pairs measuring activity of the *rectus femoris (RF)*, *biceps femoris (BF)*, *semitendinosus (ST)*, *vastus medialis (VM)*, *vastus lateralis (VL)*, *gastrocnemius medialis (GM)*, *gastrocnemius lateralis (GL)* and *tibialis anterior (TA)*.

Post-processing of the data included filtering and downsampling of the joint angle, joint torque, and EMG signals, which were originally sampled at 2000 Hz. The EMG signals were band-pass filtered between 20 Hz to 500 Hz, rectified and afterwards low-pass filtered at 6 Hz. Both filters applied were fourth-order bi-directional Butterworth filters to achieve zero phase distortion. Both joint angle and torque were filtered using a second-order bi-directional Butterworth filter with a cutoff frequency of 200 Hz to minimize the impact of noise on the training of the models. The angular joint velocity was derived from the joint angle. A second-order Butterworth filter with a cutoff frequency of 20 Hz was applied bi-directionally to the raw joint angle signal before calculating the gradient.

To calibrate the FMG signal, the baseline offset was removed using the mean values obtained from calibration measurements in a relaxed standing position. No filter was applied to the FMG signal. As the FMG sensor units allow a maximum sampling rate of 200 Hz, the joint angle, joint torque, and EMG signals were down-sampled and linearly interpolated to an equidistant number of data points to align and concatenate all data.

Each signal was z-score normalized per participant for a better comparison between the input configurations, and the estimated joint torque was later denormalized to its original scale.

F. Training and Validation

The models were trained on two distinct input configurations, one including FMG and the other including EMG.

$$\mathbf{x} = \begin{cases} (\theta_J, \omega_J, \mathbf{M}_{\text{EMG}})^T & \text{EMG} \\ (\theta_J, \omega_J, \mathbf{M}_{\text{FMG}})^T & \text{FMG} \end{cases} \tag{6}$$

where θ_J represents the joint angle, ω_J the joint angular velocity, and $\mathbf{M}$ the muscle signals.

For each joint, the muscle signal $\mathbf{M}$ included the primary muscles responsible for moving that joint [40] (Fig. 2):

$$\mathbf{M} = \begin{cases} (M_{\text{TA}}, M_{\text{GM}}, M_{\text{GL}}) & \text{ankle joint} \\ (M_{\text{BF}}, M_{\text{RF}}, M_{\text{ST}}, M_{\text{VM}}, M_{\text{VL}}) & \text{knee joint} \end{cases} \tag{7}$$

However, the effects of biarticular muscles, which influence movements of multiple joints simultaneously, have not been considered.

The estimation results were evaluated using an inter-participant leave-one-participant-out (LOPO) and an intra-participant leave-one-velocity-out (LOVO) cross-validation to assess the model performance and its generalization and personalization capabilities, respectively. The standard deviation of the model estimation was evaluated using the normalized root-mean-squared error (NRMSE):

$$\text{NRMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{\tau_i - \tilde{\tau}_i}{\tau_{max} - \tau_{min}})^2} \tag{8}$$

where n is the number of available data points, τ is the vector of true torque, $\tilde{\tau}$ is the vector of the estimated torque and τ_{max} and τ_{min} are the maximum and minimum true torque of each participant, respectively. A lower value of NRMSE implies a higher accuracy of each model. The variability of the model estimation was evaluated using the coefficient of determination (R^2), where a value closer to 1 implies a higher accuracy of each model.

To avoid overfitting, early stopping was used in the GPR, FFNN, and TCN, with a patience of 50 epochs. This method tracked the validation loss based on the last 20 % of the training data for each participant included in the training data, to avoid possible side effects of random sampling, and restored the best model if no improvement was observed over these 50 epochs. The SVR does not support callbacks in its implementation; thus, early stopping was not considered. To improve the training process, a batch size of 32 along with an Adaptive Moment Estimation (Adam) optimizer was employed with a learning rate of 0.001 for the FFNN and 0.0001 for the TCN. To accommodate the amount of training data in a SGPR, the initial inducing points were selected randomly as a subset of 0.25 % of the training data for LOPO and 2.5 % for LOVO. To reduce the computational complexity of the SVR for LOPO, the first 12 % of the data for each velocity of each relevant participant was taken as training data, while in LOVO the complete training data set was used.

III. RESULTS

Four previously presented torque estimation models were evaluated using LOPO and LOVO cross-validation to gain insight into their respective performance. The NRMSE for these validation methods are presented in Fig. 3 and Fig. 4 for both ankle and knee joint. The results of the LOPO validation are quantified in Table I as the mean and standard deviation of NRMSE over the ten testing participants.

In the LOPO validation, shown in Fig. 3 and Table I, the FFNN and TCN are among the best-performing models across both joints, both when using FMG and when using EMG signals. It can also be observed that SVR generally shows the worst performance in LOPO validation, while GPR falls in line with FFNN and TCN in most cases. In addition, the figure shows that the estimation errors based on FMG and EMG do not deviate significantly from each other in most cases. The FFNN and TCN tend to perform slightly better when using FMG, while showing less consistent error values compared to when using EMG.

TABLE I: Inter-participant validation (LOPO)

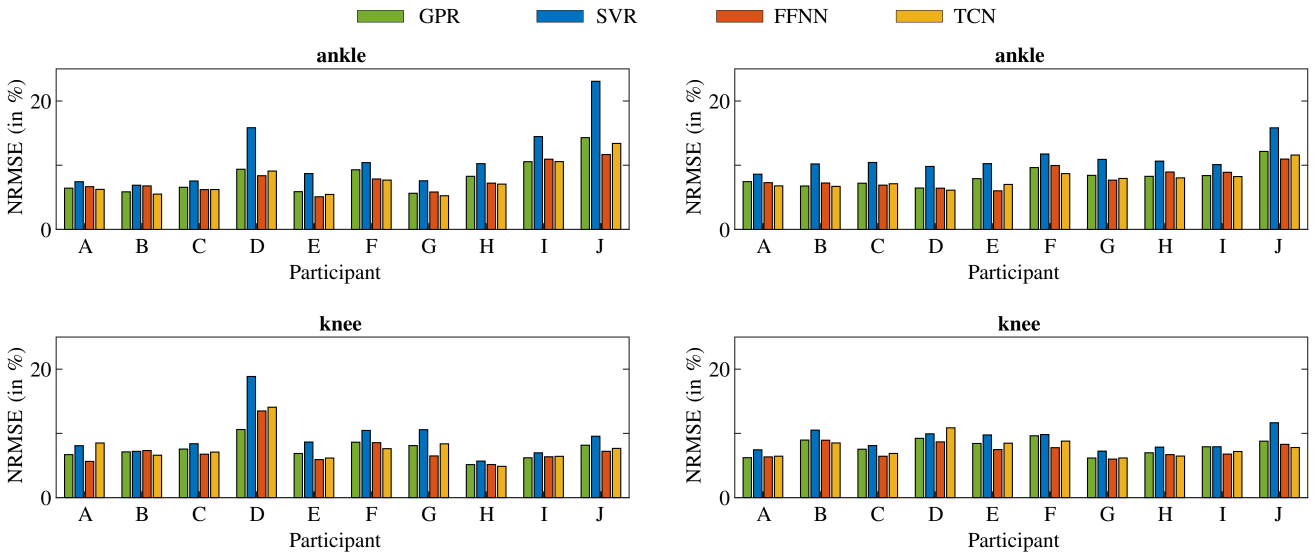
NRMSE (%)		GPR	SVR	FFNN	TCN
ankle	EMG	8.26 ± 1.6	10.85 ± 1.9	8.03 ± 1.6	7.82 ± 1.5
	FMG	8.22 ± 2.8	11.22 ± 5.2	7.66 ± 2.1	7.64 ± 2.6
knee	EMG	7.99 ± 1.2	9.02 ± 1.5	7.34 ± 1.0	7.76 ± 1.4
	FMG	7.51 ± 1.5	9.45 ± 3.6	7.29 ± 2.4	7.74 ± 2.5

$\mathbf{R^2}$		GPR	SVR	FFNN	TCN
ankle	EMG	0.90 ± .04	0.84 ± .05	0.91 ± .03	0.91 ± .03
	FMG	0.91 ± .05	0.81 ± .16	0.92 ± .05	0.92 ± .05
knee	EMG	0.91 ± .03	0.88 ± .03	0.92 ± .03	0.91 ± .04
	FMG	0.92 ± .03	0.86 ± .13	0.92 ± .06	0.91 ± .06

Values represent the mean and standard deviation.

The validation of LOVO, shown in Fig. 4 and Table II, shows similar results, with FFNN and TCN generally outperforming GPR and SVR in both joints and signal types. Here, it is also apparent that for the LOVO validation, the FMG based estimation outperforms the EMG based estimation in all cases, both in terms of the mean accuracy between speeds and in terms of the standard deviation of error values.

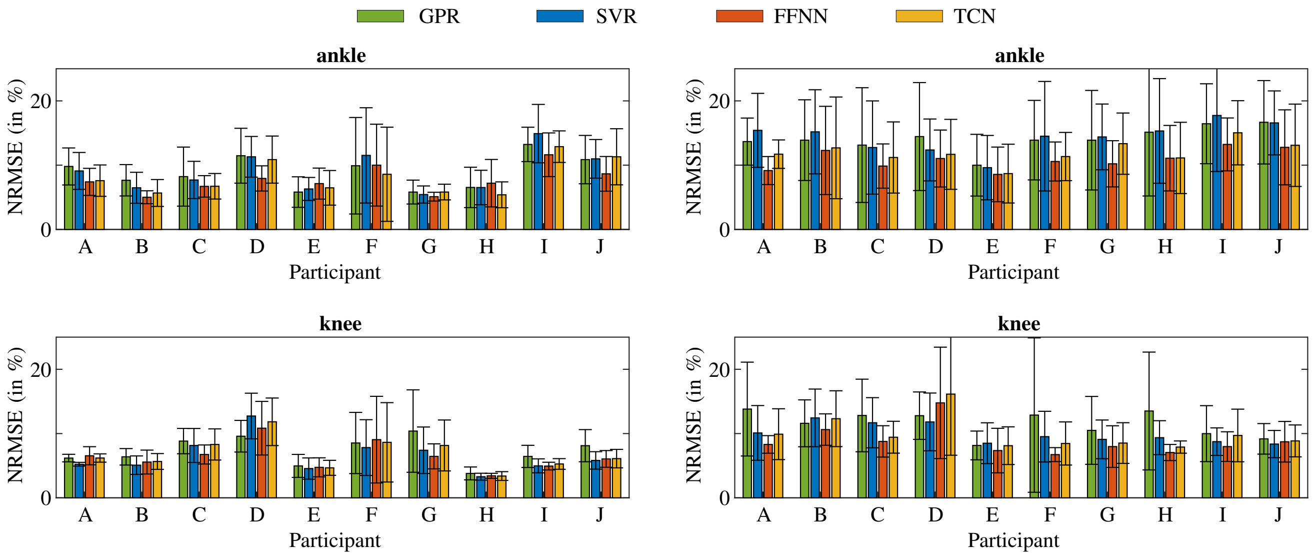
To address possible online implementation of the models in an exoskeleton, the inference time of each model was tested on a personal computer, running an Ubuntu 24 operating system, with an Intel® Core™ Ultra 7 155H processor and 16 GiB of DDR5 RAM operating at 5600 MT/s. Hereby, the neural networks (NNs) exhibited similar inference times for both the LOVO and LOPO methods. While the FFNN was the fastest, with an average inference time of about 0.25 ms, the TCN was slower at around 1.4 ms. The SVR showed more variability, with inference times of 0.36 ms for



(a) FMG

(b) EMG

Fig. 3: NRMSE of the LOPO cross-validation of all four models, GPR (green, left), SVR (blue, middle-left), FFNN (orange, middle-right), and TCN (yellow, right) of both the ankle (top) and knee (bottom) joint using (a) FMG and (b) EMG signals as input. The x-axis represents each test participant left out of the training data set.



(a) FMG

(b) EMG

Fig. 4: NRMSE of the LOVO cross-validation of all four models, GPR (green, left), SVR (blue, middle-left), FFNN (orange, middle-right), and TCN (yellow, right) of both the ankle (top) and knee (bottom) joint using (a) FMG and (b) EMG signals as input. The mean and standard deviation show the distribution over all four test velocities within each participant left out of the training data set.

LOVO and 0.63 ms for LOPO application. The GPR had the longest inference times, taking approximately 25 s for both validations.

TABLE II: Intra-participant validation (LOVO)

NRMSE (%)		GPR	SVR	FFNN	TCN
ankle	EMG	14.67 ± 9.4	14.76 ± 8.6	11.33 ± 6.8	12.50 ± 7.8
	FMG	9.23 ± 5.7	9.14 ± 5.1	7.72 ± 4.0	8.48 ± 5.5
knee	EMG	16.89 ± 9.8	14.67 ± 6.9	13.03 ± 7.2	14.58 ± 7.8
	FMG	10.73 ± 5.6	9.64 ± 6.0	9.39 ± 5.2	10.07 ± 6.0
R^2		GPR	SVR	FFNN	TCN
ankle	EMG	0.77 ± .21	0.77 ± .21	0.88 ± .09	0.85 ± .12
	FMG	0.91 ± .09	0.92 ± .10	0.94 ± .06	0.93 ± .08
knee	EMG	0.83 ± .18	0.88 ± .09	0.92 ± .07	0.89 ± .08
	FMG	0.93 ± .07	0.95 ± .06	0.95 ± .06	0.95 ± .06

Values represent the mean and standard deviation.

IV. DISCUSSION

Among the four methods evaluated, the LOPO validation indicated that both NNs are better suited to generalize to new users when utilizing FMG and EMG. In most cases, GPR and SVR performed worse. However, there appear to be minimal differences in performance among the GPR, FFNN, and TCN. Multiple training and testing sessions using LOPO revealed variability in the performance of the GPR across different runs. This inconsistency may stem from the sparsification process that relies on a very small random subset of the training data as inducing points. The SVR, which was also trained on a fixed, but larger, set of 12 % of the data (approximately two swings per velocity and participant), exhibited notably poorer performance. The variability in the NRMSE values shown in the FMG-based estimates suggests that electrical activation tends to be more consistent among individuals than the volumetric changes that occur during muscle contraction.

While [31] found that an FMG and EMG-enhanced GPR model performs better for unknown task characteristics within a participant compared to unknown participants, our results suggest the opposite, regardless of the model used. The FMG-based models show only slightly higher NRMSEs in intra-participant performance compared to inter-participant performance. In contrast, the EMG-based model performs notably worse within a subject, leading us to conclude that the intra-participant variability of EMG is greater than that of FMG. According to [39], FMG demonstrates more consistency in peak amplitude within a single participant and across multiple participants than EMG. However, the full width at half maximum (FWHM) of the signal in relation to joint angle varies more across various participants in FMG than in EMG. These findings suggest that the model focuses more on the signal's peak amplitude rather than the range of joint angles in which the muscle is active.

Previous studies on EMG-based joint torque estimation during isokinetic lower limb movements have reported NRMSE ranging from 7 % to 30 %, depending on the joint and modeling approach. Chandrapal et al. [19] demonstrated that a multi-layer perceptron (MLP) could estimate knee flexion and extension torques using EMG signals alone, with NRMSEs between 20 % to 30 %. Su et al. [23] employed a time delay artificial neural network (TDNN) to estimate ankle inversion and eversion torques using EMG and joint angle data, achieving an NRMSE of $7.9 \pm 0.03\%$, outperforming SVR ($9.1 \pm 0.04\%$) and ANN ($9.8 \pm 0.05\%$). Schulte et al. [25] used a convolutional neural network (CNN) to estimate knee torques during non-weighted flexion and extension, reporting an NRMSE of $9.2 \pm 4.4\%$ using EMG. These models were typically trained on 60 % to 80 % of the dataset randomly pooled across participants, potentially leading to better overall performance than what could be expected with unknown participants or varying joint velocities. All four models in this work achieved higher accuracy for previously unseen participants (Table I) than those aforementioned values in the literature, independent of the input muscle activity signal. However, intra-participant variability was greater in the EMG-based models compared to FMG-based models and prior studies, underlining the potential of FMG as an alternative to EMG.

An evaluation of the inference times revealed that FFNN, TCN, and SVR have suitable durations for use in online applications. Assuming there are no additional time requirements in the control system, an inference time of up to 5 ms is acceptable due to the sampling frequency of 200 Hz for the FMG units [15]. Under strict time constraints, the FFNN may be the most appropriate method. In contrast, the inference time of the GPR would impose severe limits on the control bandwidth, combined with its requirements for storage and computational power, making it impractical in its current form.

According to [27] EMG improves model performance more in non-cyclic than in cyclic real-life tasks. The limited set of isokinetic movements considered in this study, which included only flexion and extension, may thus have influenced the results. Consequently, the findings should be validated using an expanded dataset incorporating cyclic and non-cyclic real-life tasks.

Additionally, the performance of NNs may have been affected by the small amount of data utilized in this work, as NNs typically benefit from a larger volume of training data [41]. Therefore, it would be advantageous to enhance the dataset by including more motions, participants, and repetitions.

The findings in this work are based solely on normalized and filtered data, which is not directly applicable for online exoskeleton control. To effectively apply a data-driven model for online exoskeleton control, further investigation is needed to evaluate the model's performance using non-normalized data for its estimations and using filters better applicable for online applications.

While FMG shows great promise as an alternative to EMG in isokinetic knee and ankle joint motion, future research provides an opportunity to expand the dataset and incorporate real-life tasks. These efforts will undoubtedly optimize model performance and pave the way for innovative, practical exoskeleton control applications.

V. CONCLUSION

The presented comparative analysis demonstrates that FMG serves as an effective alternative to EMG for knee and ankle joint torque estimation, with neural network models achieving particularly high accuracy and fast inference times suitable for real-time applications.

Furthermore, the ability of FMG to perform comparably to or even better than EMG in inter- and intra-participant settings highlights its potential for enhancing exoskeleton control and personalizing musculoskeletal interactions. These insights will inform the design and control of exoskeletons and ultimately lead to improved human-exoskeleton interaction.

REFERENCES

[1] D. D. Molinaro, I. Kang, and A. J. Young, "Estimating human joint moments unifies exoskeleton control, reducing user effort," *Science Robotics*, vol. 9, no. 88, p. eadi8852, Mar. 2024.

[2] K. G. Rabe, M. H. Jahanandish, K. Hoyt, and N. P. Fey, "Use of Sonomyography for Continuous Estimation of Hip, Knee and Ankle Moments During Multiple Ambulation Tasks," in *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. New York, NY, USA: IEEE, Nov. 2020, pp. 1134–1139.

[3] Z. S. Mahdian, H. Wang, M. I. M. Refai, G. Durandau, M. Sartori, and M. K. MacLean, "Tapping Into Skeletal Muscle Biomechanics for Design and Control of Lower Limb Exoskeletons: A Narrative Review," *Journal of Applied Biomechanics*, vol. 39, no. 5, pp. 318 – 333, 2023.

[4] Q. Zhao, R. Deepak, B. A. Gebre, K. J. Nolan, K. Pochiraju, and D. Zanotto, "Gaussian Process Regression Models for On-Line Ankle Moment Estimation in Exoskeleton-Assisted Walking," in *2024 10th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. Heidelberg, Germany: IEEE, Sep. 2024, pp. 1171–1176.

[5] O. Sherif, M. M. Bassuoni, and O. Mehrez, "A survey on the state of the art of force myography technique (FMG): analysis and assessment," *Medical & Biological Engineering & Computing*, vol. 62, no. 5, pp. 1313–1332, May 2024.

[6] C. Fleischer and G. Hommel, "A Human–Exoskeleton Interface Utilizing Electromyography," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 872–882, Aug. 2008.

[7] J. Taborri, E. Palermo, S. Rossi, and P. Cappa, "Gait Partitioning Methods: A Systematic Review," *Sensors*, vol. 16, no. 1, 2016.

[8] J. Taborri, J. Keogh, A. Kos, A. Santuz, A. Umek, C. Urbanczyk, E. Van Der Kruk, and S. Rossi, "Sport Biomechanics Applications Using Inertial, Force, and EMG Sensors: A Literature Overview," *Applied Bionics and Biomechanics*, vol. 2020, pp. 1–18, Jun. 2020.

[9] S. Jiang, P. Kang, X. Song, B. Lo, and P. B. Shull, "Emerging Wearable Interfaces and Algorithms for Hand Gesture Recognition: A Survey," *IEEE Reviews in Biomedical Engineering*, pp. 1–1, 2021.

[10] A. Ziai and C. Menon, "Comparison of regression models for estimation of isometric wrist joint torques using surface electromyography," *Journal of NeuroEngineering and Rehabilitation*, vol. 8, no. 1, p. 56, 2011.

[11] D. Xiong, D. Zhang, X. Zhao, and Y. Zhao, "Deep Learning for EMG-based Human-Machine Interaction: A Review," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 3, pp. 512–533, Mar. 2021.

[12] D. G. E. Robertson, G. E. Caldwell, J. Hamill, G. Kamen, and S. N. Whittlesey, *Research Methods in Biomechanics*, 2nd ed. Human Kinetics, 2014.

[13] C. Castellini and V. Ravindra, "A wearable low-cost device based upon Force-Sensing Resistors to detect single-finger forces," in *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*. Sao Paulo, Brazil: IEEE, Aug. 2014, pp. 199–203.

[14] M. Connan, E. Ruiz Ramírez, B. Vodermayer, and C. Castellini, "Assessment of a Wearable Force- and Electromyography Device and Comparison of the Related Signals for Myocontrol," *Frontiers in Neurorobotics*, vol. 10, 2016.

[15] C. Marquardt, P. Weiner, M. Dežman, and T. Asfour, "Embedded barometric pressure sensor unit for force myography in exoskeletons," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, Ginowan, Okinawa, Japan, November 2022, pp. 67–73.

[16] Y. Koike and M. Kawato, "Estimation of dynamic joint torques and trajectory formation from surface electromyography signals using a neural network model," *Biological Cybernetics*, vol. 73, no. 4, pp. 291–300, Sep. 1995.

[17] J.-J. Luh, G.-C. Chang, C.-K. Cheng, J.-S. Lai, and T.-S. Kuo, "Isokinetic elbow joint torques estimation from surface EMG and joint kinematic data: using an artificial neural network model," *Journal of Electromyography and Kinesiology*, vol. 9, no. 3, pp. 173–183, Apr. 1999.

[18] K. Ullah and Jung-Hoon Kim, "A mathematical model for mapping EMG signal to joint torque for the human elbow joint using nonlinear regression," in *2009 4th International Conference on Autonomous Robots and Agents*. Wellington: IEEE, Feb. 2009, pp. 103–108.

[19] M. Chandrapal, X. Chen, W. Wang, B. Stanke, and N. Le Pape, "Investigating improvements to neural network based EMG to joint torque estimation," *Paladyn, Journal of Behavioral Robotics*, vol. 2, no. 4, pp. 185–192, Dec. 2011.

[20] D. Kim, K. Koh, G. Oppizzi, R. Baghi, L.-C. Lo, C. Zhang, and L.-Q. Zhang, "Simultaneous Estimations of Joint Angle and Torque in Interactions with Environments using EMG," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. Paris, France: IEEE, May 2020, pp. 3818–3824.

[21] M. Wang, Z. Chen, H. Zhan, J. Zhang, X. Wu, D. Jiang, and Q. Guo, "Lower Limb Joint Torque Prediction Using Long Short-Term Memory Network and Gaussian Process Regression," *Sensors*, vol. 23, no. 23, p. 9576, Dec. 2023.

[22] Q. L. Li, Y. Song, and Z. G. Hou, "Estimation of Lower Limb Periodic Motions from sEMG Using Least Squares Support Vector Regression," *Neural Processing Letters*, vol. 41, no. 3, pp. 371–388, Jun. 2015.

[23] C. Su, S. Chen, H. Jiang, and Y. Chen, "Ankle Joint Torque Prediction Based on Surface Electromyographic and Angular Velocity Signals," *IEEE Access*, vol. 8, pp. 217 681–217 687, 2020.

[24] H. Dinovitzer, M. Shushtari, and A. Arami, "Feedforward Control of Lower Limb Exoskeletons: Which Torque Profile Should We Use?" *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 382–389, Jan. 2024.

[25] R. V. Schulte, M. Zondag, J. H. Buurke, and E. C. Prinsen, "Multi-Day EMG-Based Knee Joint Torque Estimation Using Hybrid Neuromusculoskeletal Modelling and Convolutional Neural Networks," *Frontiers in Robotics and AI*, vol. 9, p. 869476, Apr. 2022.

[26] L. Moreira, J. Figueiredo, J. P. Vilas-Boas, and C. P. Santos, "Kinematics, Speed, and Anthropometry-Based Ankle Joint Torque Estimation: A Deep Learning Regression Approach," *Machines*, vol. 9, no. 8, p. 154, Aug. 2021.

[27] K. L. Scherpereel, D. D. Molinaro, M. K. Shepherd, O. T. Inan, and A. J. Young, "Improving Biological Joint Moment Estimation During Real-World Tasks With EMG and Instrumented Insoles," *IEEE Transactions on Biomedical Engineering*, vol. 71, no. 9, pp. 2718–2727, Sep. 2024.

[28] M. R. U. Islam, A. Waris, E. N. Kamavuako, and S. Bai, "A comparative study of motion detection with FMG and sEMG methods for assistive applications," *Journal of Rehabilitation and Assistive Technologies Engineering*, vol. 7, 2020, publisher: SAGE Publications Ltd STM.

[29] Z. G. Xiao and C. Menon, "A Review of Force Myography Research and Development," *Sensors*, vol. 19, no. 20, 2019.

[30] M. Anvaripour, M. Khoshnam, C. Menon, and M. Saif, "FMG- and RNN-Based Estimation of Motor Intention of Upper-Limb Motion in Human-Robot Collaboration," *Frontiers in Robotics and AI*, vol. 7, p. 573096, Dec. 2020.

[31] C. Marquardt, A. Schulz, M. Dežman, G. Kurz, T. Stein, and T. Asfour, "Force myography based torque estimation in human knee and ankle joints," in *IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, USA, May 2025, accepted.

[32] C. E. Rasmussen and Christopher K. I. Williams, *Gaussian process for machine learning*. London, England: The MIT Press, 2006, oCLC: 999818656.

[33] A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman, "GPflow: A Gaussian process library using TensorFlow," 2016, version Number: 1.

[34] M. Awad and R. Khanna, "Support Vector Regression," in *Efficient Learning Machines*. Berkeley, CA: Apress, 2015, pp. 67–80.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, 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," Jun. 2018, arXiv:1201.0490 [cs].

[36] M. Awad and R. Khanna, "Deep Neural Networks," in *Efficient Learning Machines*. Berkeley, CA: Apress, 2015, pp. 127–147.

[37] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," Mar. 2016, arXiv:1603.04467 [cs].

[38] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," 2018, version Number: 2.

[39] C. Marquardt, G. Kurz, M. Dežman, T. Stein, and T. Asfour, "Comparative analysis of force myography and electromyography signals in isokinetic ankle and knee joint motion," in *IEEE/RAS/EMBS International Conference on Rehabilitation Robotics (ICORR)*, Chicago, USA, May 2025, accepted.

[40] D. A. Neumann and E. E. Rowan, *Kinesiology of the musculoskeletal system: foundations for physical rehabilitation*, 1st ed. St. Louis: Mosby, 2002, oCLC: 992214437.

[41] C. Aggarwal, "An Introduction to Neural Networks," in *Neural Networks and Deep Learning*. Cham: Springer International Publishing, 2023, pp. 1–27.