This paper appears in Mechatronics (A Journal of IFAC), 2023.
https://doi.org/10.1016/j.mechatronics.2023.103060

Uncertainty-aware Safe Adaptable Motion Planning of Lower-limb Exoskeletons Using Random Forest Regression

Mojtaba Akbari[a,*], Javad K. Mehr[a,b], Lei Ma[a], Mahdi Tavakoli[a]

[a]*Department of Electrical and Computer Engineering, University of Alberta, AB, Canada T6G 1H9*
[b]*Department of Medicine, University of Alberta, Edmonton, AB, Canada, T6G 2R3*

Abstract

Human safety and data security are two of the main concerns that have limited the utilization of deep learning-based techniques in medical robotic applications. Such concerns are amplified by uncertainty in the deep learning run-time predictions. In this paper, we propose a novel framework for incorporating uncertainty analysis that is fast enough (updates in $20Hz$) to be used in the control loop of a medical robot and that considers both the training and testing phases of the deep learning algorithm. As a case study focusing on the use of a lower-limb exoskeleton to assist the walking of people with disability, we learn the passive human-exoskeleton system's dynamics using Random Forest Regression (RFR) and quantify the uncertainty level of its prediction. Whereas prior art fed the estimated human-robot interaction torque values to the adaptable Central Pattern Generators (CPGs) to refine the gait trajectories, our contribution is to leverage the knowledge of the predictions' uncertainty levels to ensure safety in human-robot interaction. Our proposed framework for uncertainty-aware control of medical robots finds the similarities of labels and predictions in the training set using Kullback-Leibler (KL) divergence, while in the test phase, it detects out-of-distribution (OOD) data using Mahalanobis distance between test feature and training distribution. As compared to state-of-the-art methods, the proposed method is real-time and addresses the issue of uncertainty in the decisions of the robot controller. We have tested the proposed method on ExoH3 (Tehnaid S.L.) lower-limb exoskeleton. The experiments were conducted to evaluate the performance of the uncertainty analysis technique. The results demonstrate that our proposed uncertainty analysis technique can detect OOD features resulting in unsafe motion planning. We also showcase the importance and effectiveness of using uncertainty analysis in the lower-limb exoskeleton case study.

Keywords: Rehabilitation Wearable Robots, Artificial Intelligence, Deep Learning, Medical Robots, Uncertainty Analysis

List of Variables and Definitions

τ_{input}	Total Torque Measured by Torque Sensors

*Corresponding author
Email address: akbari@ualberta.ca (Mojtaba Akbari)

τ_{exo}	HRI Torque Applied to Exoskeleton's Joints
τ_{motor}	Exoskeleton Motor Torque
τ_h	Human Torque Vector
τ_{HRI}	Physical Human-robot Interaction Torque
$\tau_{h,pass}$	Passive Torque Dynamic of Human-Exoskeleton
$M_q(q)$	Inertia Matrix
$C_q(q)$	Coriolis, Centrifugal and Damping Term
$G_q(q)$	Gravitational Torque
$q, \dot{q}, \ddot{q}$	Position, Velocity and Acceleration of Joints
$E_i(t)$	Energy Transferred to Joint i
N_i	Number of Adjacent Joints to Joint i
$a(t)$	Amplitude of the Movement
$f(t)$	Frequency of the Movement
f_{test}	Test Feature
$C_{RF}(f_{test})$	Adaptable Uncertainty Gain
D	Adaptable Uncertainty Threshold
c_{i_j}, d_{i_j}	Fourier Series Coefficients
O_{train}	Random Forest Predictions for Training Data
L_{train}	Training Labels
F_{train}	Training Features
D_{KL}	KL Divergence
D_M	Mahalanobis Distance
D_{M_t}	Mahalanobis Distance after Thresholding
$P_l(\overrightarrow{x})$	Distribution of Training Labels
$P_p(\overrightarrow{x})$	Distribution of Training Predictions
$P_t(\overrightarrow{x})$	Distribution of Training Set

$\overrightarrow{x}$	Vector of One Feature
x_i	Element of Feature Vector
y_i	Sample from Training Predictions
N_f	Total Number of Samples in Feature Set
μ_l, σ_l	Training Labels Mean and Standard Deviation
μ_p, σ_p	Predictions Mean and Standard Deviation
μ_f, Σ_f	Training Data Mean and Standard Deviation

1. Introduction and Background

Deep learning has been successfully applied to many medical applications over the past decades, empowered by a large amount of available data and the data-driven system development paradigm. However, current deep learning-based methods still suffer from quality issues that can give rise to major concerns, especially in the context of medical applications [1], many scenarios of which have high safety requirements. In general, medical robots are often safety-critical systems and thorough safety analysis is required, especially when data-driven Artificial Intelligence (AI) is integrated as part of the brain of the robot control system as a decision-maker [2]. In particular, a typical data-driven AI model (e.g., deep neural network) learns the decision logic to handle future unseen data that follow a similar training data distribution. In other words, the safety concern is highlighted by the fact that deep learning does not provide a statistical guarantee of reliable performance for a wide range of input scenarios and has limited capability in handling data that fall outside the learning distribution [3]. For example, in the context of autonomous driving systems that use a visual perception system to understand the environment, although the visual system may have faced many situations during training, it still cannot be trained on all possible scenarios. Therefore, it is essential to perform analysis to understand when an input data point is outside the decision boundary of the deep neural network (DNN) to avoid actions based on uncertain predictions that could have catastrophic safety repercussions. The same situation may happen for medical robots with even greater safety-critical concerns as AI-enabled medical robots rely on imagers and sensors that feed a DNN to help decision-making. This motivates us to propose the uncertainty assessment technique in the control loop of the medical robot, to continuously monitor the actions and decisions of DNN in the AI-powered medical robots and enhance safety in human-robot interaction.

Uncertainty analysis of DNN prediction against a particular input has been investigated in the literature for different types of DNNs [4]. In [5], Abdar *et al.*, do a comprehensive review of different techniques, applications and challenges in uncertainty quantification of deep learning while [6, 7] discuss the challenges of uncertainty estimation in the context of medical applications. Based on the literature, there are three main categories of uncertainty analysis techniques in the prediction of DNN so far. The first category learns model uncertainty along with its prediction in the DNN's feedforward, which means that a well-trained DNN should have a prediction label and

its corresponding uncertainty of the prediction result vector [8]. The second category of methods considers the uncertainty brought in the DNN development process, such as from different DNN architectures, which can be simulated methods like adding some additional layers to estimate the uncertainty of predictions. This is usually implemented with dropout layers. Gal and Izmailov *et al.* [9, 10, 11] propose a method for uncertainty analysis of DNNs by mathematical modelling their dropout layers based on intrinsic dropout features. Dropout layers randomly select a ratio of input features for training to prevent overfitting in DNNs. Using the gradient of batches is a common method for training DNNs and updating their weight. Wen *et al.* [12] introduce a new type of layer inspired by dropout called Flipout for decorrelating the gradient within mini-patch during training and also estimating uncertainty by measuring perturbation in the weights. The idea developed by Liu *et al.* [13] shows that adding a weight normalization step during training and replacing the output layer with a Gaussian Process increases the uncertainty awareness of the DNN.

In addition to the two main categories of uncertainty analysis mentioned above, out-of-distribution (OOD) is a widely used safety analysis method and can be considered an uncertainty analysis technique. The method proposed in [13, 14] uses the distance between the test sample and training data for OOD detection and uncertainty level estimation in DNN. Several other state-of-the-art OOD detection techniques are commonly used in the literature. The first one is the simple baseline method [15] where in and out of distribution samples are classified with different probability distributions with softmax probability. The softmax layer is used at the end of the network structure to assign a probability to each output label. The second method is called Out-of-DIstribution detector for Neural Networks (ODIN) [16]. This method shows that small perturbations substantially affect in-distribution samples more than OOD samples. The third method is Mahalanobis [17] that computes a score by measuring Mahalanobis distance [18] between the test sample and the closest Gaussian distribution. The fourth method called Outlier Exposure [19] is the enhancement of the simple baseline method with the integration of the OOD data into the training of the DNN model. The last method is Likelihood-Ratio [20] that utilizes a separately trained neural network for learning OOD score. A review and a comparison of the OOD detection methods mentioned above and uncertainty analysis in deep learning can be found in [1, 21, 22]. The methods mentioned above are not fast enough to be used in real-time applications such as within the control loop of a robot. Therefore, this paper revisits the algorithms mentioned above to make them real-time for integration into the robot control loop.

Lower-limb exoskeleton is a case study that we use to evaluate the performance of the proposed uncertainty analysis algorithm in the control loop of the robot. Central Pattern Generators (CPGs) is a strategy for motion planning of lower-limb exoskeleton and bio-inspired robots, whose inherent features like time-continuous rhythmic motions are similar to natural bipedal locomotion [23]. Up to the present, CPG for trajectory planning of exoskeleton has been investigated in [24, 25, 26, 27, 28]. In addition, DNNs have been used in different parts of exoskeleton control such as torque estimation, trajectory shaping, etc., by taking advantage of their model-free nature and fast processing rate. The techniques proposed in [23, 27] use a DNN for passive torque estimation based on kinematic inputs and leverage the estimation inside the CPG for gait trajectory planning. To this end, they collected exoskeleton joint torques for driving the user to different frequencies of walking in the absence of any interaction between the human and the robot. The

collected data was then fed into the DNN to estimate the passive torque for any given position and velocity of the joints. However, the previous methods that have been proposed in DNN-based exoskeleton control have missed the critical consideration of the reliability and safety of the DNN predictions for irrelevant, unseen, or unprepared scenarios, especially when the input lies outside their learned decision boundary [1]. The proposed method considers the training data and test input to analyze the trajectory shaping algorithm's reliance on the deep learning algorithm's prediction. As a result of this method, the safety issue of using an exoskeleton for walking while a CPG is used for trajectory shaping will be resolved.

In this paper, we solve the problem of using uncertainty of deep learning decision-makers in the control loop of the robot while maintaining the real-time necessity of the algorithm, which has not been investigated as mentioned above. In this regard, random forest regression (RFR) is adopted to predict the passive dynamics of the human-exoskeleton system (known as a human user interacting with the exoskeleton) using the position, velocity and acceleration of six joints as inputs. This generates an estimation of the human-exoskeleton interaction torque, which is used to inform and update the motion planning (put simply, the human applying positive/negative torques on the exoskeleton joints results in a faster/slower planned motion). Our contribution is to further manipulate CPG dynamics by performing uncertainty estimation to reduce the potential safety risks. Specifically, we consider the uncertainty of RFR predictions in the adaptable CPG dynamics for gait motion planning to control the exoskeleton and enhance safety in human-robot interaction appropriately. Our proposed uncertainty analysis technique uses Kullback-Leibler (KL) divergence between training labels and predictions to measure the random forest uncertainty. In addition, we use Mahalanobis distance between the current input and training distributions to check the distance between the test feature and the training set to refine the uncertainty value. The Mahalanobis distance acts as the OOD detection part of our proposed technique. Note that the proposed uncertainty analysis technique is a framework that considers the distribution of data and is independent of the specific learning technique used inside the framework. Overall, the contributions of this paper are summarized as follows:

- We propose a novel uncertainty analysis technique for deep learning-based medical systems considering the training and testing data distributions. We leverage KL divergence as the prediction uncertainty measure and refine its value using Mahalanobis distance of the test sample from the training distribution. We demonstrate the effectiveness of the proposed framework in the control of exoskeletons. We design to move the computational complexity to the training phase to adapt the uncertainty analysis to be applicable in real-time control of exoskeletons for the first time compared with the latest existing methods like [29] which has significant computational complexities.

- We manipulate the CPG dynamics using the calculated uncertainty in predictions to have a safe human-robot interaction during task execution. For the first time, our proposed uncertainty technique detects potentially unsafe actions of the exoskeleton resulting from less certain predictions by deep learning. Furthermore, it adjusts CPGs trajectory's amplitude and frequency while de-emphasizing the contribution of such uncertain predictions.

- Another novelty of our proposed method comes from leveraging RFR for predicting the in-

teraction torque between the exoskeleton and the passive component of human-exoskeleton dynamics based on velocity, position and acceleration of the exoskeleton joints as input. The advantages of using RFR compared to the state-of-the-art methods [30, 31, 32] are its model-free nature and independence from the type of the exoskeleton, which are very important for our application due to the complex user-exoskeleton dynamic nature and different types of exoskeleton being used in clinics. Furthermore, RFR has low computational complexity, which is of great importance for our application scenario, considering the real-time control requirement compared with DNNs.

The rest of this paper is organized as follows. First, in Section 2, we introduce the proposed uncertainty analysis technique and details of modified CPG equations with the consideration of prediction uncertainty. Then, in Section 3, we demonstrate the experimental result of the proposed method, followed by the concluding remarks in Section 4.

2. Methodology

This section introduces the background and mathematical formulations of the proposed uncertainty-aware exoskeleton control technique. This strategy adjusts the gains in the CPG based on the uncertainty of random forest predictions to give the exoskeleton the ability to have safe interactions with human users. The overview of the proposed technique is summarized in Figure 1. Here, τ_{motor} is the total torque measured by exoskeleton torque sensors, τ_{exo} is the torque applied to the exoskeleton's joint by joint-level position controller, and τ_h is the human torque vector.

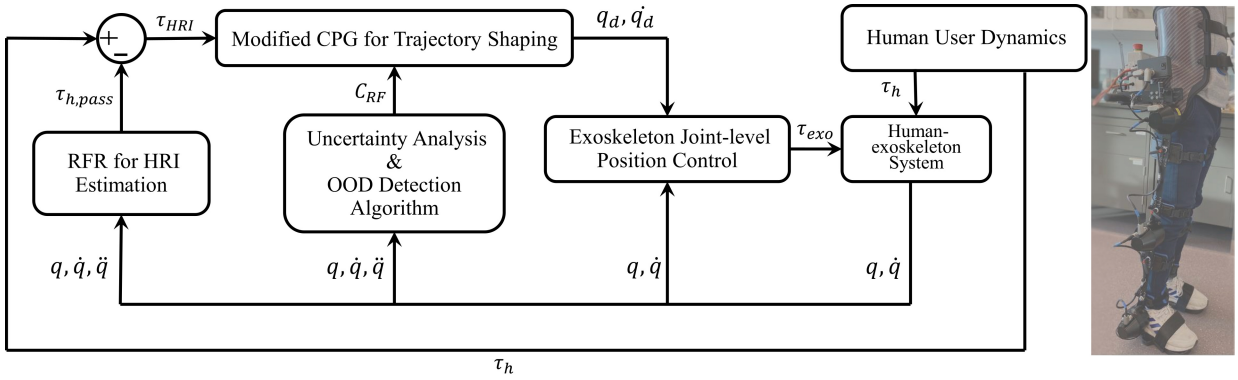


Figure 1: Overview of proposed uncertainty-aware exoskeleton control method.

2.1. Random Forest Regressor Method for HRI Estimation

The non-linear dynamics of multi-DOF lower-limb exoskeleton interacting with the human user is

$$M_q(q)\ddot{q} + C_q(q)\dot{q} + G_q(q) = \tau_{HRI} + \tau_{motor} + \tau_{h,pass}. \quad (1)$$

Here, q is the exoskeleton's joint positions vector, $M_q(q)$ is the inertia matrix, $C_q(q)$ is the Coriolis, centrifugal and damping term, and $G_q(q)$ is the gravitational torques. The torque values on the right side of (1) are exoskeleton motor torque (τ_{motor}), human-robot interaction (HRI) torque

(τ_{HRI}), and passive dynamics of human exoskeleton system ($\tau_{h,pass}$). The passive dynamics of the human exoskeleton system is defined as the exoskeleton joint torques required to drive a user in the absence of any interaction between the human and the exoskeleton. Given these passive torque values, the active human-robot interaction torque can be calculated as the difference between measured joint torque applied by human (τ_h) and estimated passive torque. Our proposed method uses ensemble decision trees in the form of a random forest regression (RFR) for estimating the passive torque ($\tau_{h,pass}$) that is used for estimating HRI torque (τ_{HRI}) using

$$\tau_{HRI} = \tau_h - \tau_{h,pass}. \tag{2}$$

τ_{HRI} calculated in (2) is used to calculate the energy transferred between the human user and exoskeleton. The frequency and amplitude of the desired trajectory will be affected by the energy in walking with an exoskeleton. The mathematical details are given in Section 2.2.

Random forests are multiple decision trees with voting schemes at the end of these trees for making predictions. Decision trees have proven to have a good performance for regression problems and are easy to be trained on commodity hardware [33, 34]. The complexity and the performance of the regression algorithm are essential considerations in our application under the context of a medical robot. The performance is also critical as the decision coming from RFR affects the physical human-robot interaction (pHRI). Figure 2 shows the structure of the model that has been developed for estimating the passive torque of the human ($\tau_{h,pass}$).

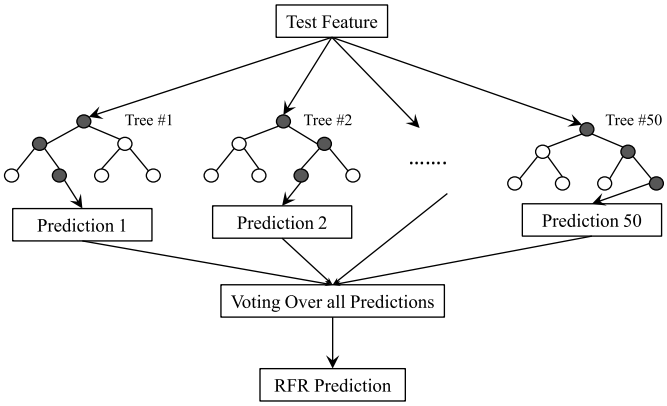


Figure 2: Random forest model for human passive torque estimation $\tau_{h,pass}$

In this paper, we asked a human user to walk with an exoskeleton over the ground in several training data collection trials. We used CPGs with different walking speeds for trajectory shaping during these trials. Each exoskeleton joint's position, velocity, acceleration and corresponding motor torques have been saved. The user was asked to comply with the exoskeleton-imposed motion and not apply any force to the exoskeleton so that the torque of each joint equals the passive torque. We changed the velocity of the exoskeleton walking for training the random forest. The position, velocity and acceleration of the exoskeleton's joints $(q, \dot{q}, \ddot{q})$ are the input to the random forest model and the estimated passive torques of each joint ($\tau_{h,pass}$) is the output. Applying torques to the joint of the exoskeleton will result in faster motions. Hence, τ_{HRI}, which has a direct relation to the energy transferred between the user and exoskeleton, is used in the modified adaptable CPGs algorithm for shaping the trajectory of the exoskeleton along with its uncertainty.

2.2. Modified Adaptable CPG for Trajectory Shaping

Modified adaptable CPG dynamics are used for designing gait trajectories based on HRI torque. The HRI torque (τ_{HRI}) is used for calculating the energy transferred between the user and exoskeleton. This is inspired by the method in [27, 23]. We calculate the energy of each joint i by using the HRI torque and the velocity of each joint as

$$E_i(t) = \int_0^t \tau_{HRI_i}(t)\dot{q}_i(t). \tag{3}$$

Here, $\tau_{HRI_i}(t)$ is calculated using (2) for each joint and $\dot{q}_i(t)$ is the velocity of each joint coming from the exoskeleton sensors $(i = 1, ..., n)$. The modified adaptable CPG dynamics for the desired joint trajectory generation based on the amplitude and the frequency update are proposed as

$$\begin{aligned}
\dot{\phi}_i(t) &= f(t) + \sum_{j=1}^{N_i} v_{ij} \sin(\phi_i(t) - \phi_j(t) - \psi_{ij}) \\
\ddot{f}(t) &= \alpha_f(\frac{\alpha_f}{4}(F + C_{RF}(f_{test}) \sum_{k=1}^{n} \lambda_k E_k - f(t)) - \dot{f}(t)) \\
\ddot{a}(t) &= \alpha_a(\frac{\alpha_a}{4}(A + C_{RF}(f_{test}) \sum_{k=1}^{n} \eta_k E_k - a(t)) - \dot{a}(t)).
\end{aligned} \tag{4}$$

Here, N_i is the number of adjacent joints to the joint i. $a(t)$ is the amplitude of the movement, and $f(t)$ is the frequency of the movement. α_f and α_a are constant parameters. λ_k and η_k are constant gains for updating the frequency and amplitude of the gait cycles, based on the injected pHRI energy E_k defined in (3). A change in frequency will result in changes in walking speed, while a change in amplitude will result in changes in the walking step length of the user with the exoskeleton. In the above, we have modified the adaptable CPG dynamic proposed in [27, 23] by adding $C_{RF}(f_{test})$ to the energy term in the frequency and amplitude dynamic formula. Here $C_{RF}(f_{test})$ is the uncertainty in the estimation of passive torque using RFR and will be explained in Section 2.3. Adding $C_{RF}(f_{test})$ to (4) enables the algorithm to scale the effect of the pHRI energy in the CPG gait trajectory update. For instance, if the input feature is far from the distribution of the training data, the distance value will be high. Then this technique is able to reduce the effect of the energy term in (4) by using $C_{RF}(f_{test})$ found in (8).

We now can formulate the desired trajectory of joint i using (4) and Fourier series as

$$q_{d_i}(t) = a(t)(c_{i_0} + \sum_{j=1}^{N_i} (c_{i_j} \cos j\phi_i(t) + d_{i_j} \sin j\phi_i(t))). \tag{5}$$

Here, c_{i_j} and d_{i_j} are the Fourier series coefficients for each joint's trajectory. The trajectory calculated in (5) considers the uncertainty of the prediction in real-time using (8) and (4). The proposed adaptable CPG can be used in any other motion planning approach for exoskeletons in a similar manner. We did it for CPGs as one example of motion planning methods for the lower-limb exoskeleton.

2.3. Uncertainty Analysis & OOD Detection

Our proposed uncertainty analysis technique considers both the training and run-time prediction phase for determining the uncertainty of the prediction. The similarity between training labels and model predictions during training is a measure that can be used to assess the model's reliability for an input that falls inside the training distribution. We use Kullback-Leibler (KL) divergence as a similarity measure in this regard. Our proposed technique finds the Mahalanobis distance between the input feature and training data distribution during the test phase to check how far the test feature is from the training data for updating uncertainty values. This mechanism refines the initial uncertainty measure found using KL divergence for the test sample in real time. Using Mahalanobis distance for OOD detection is inspired by the method proposed in [17]. The overview of the proposed uncertainty analysis and OOD detection technique is shown in Figure 3. O_{train}, L_{train} and F_{train} are predictions of the training set, labels of the training set and input training features, respectively. These data are used for calculating KL divergence.

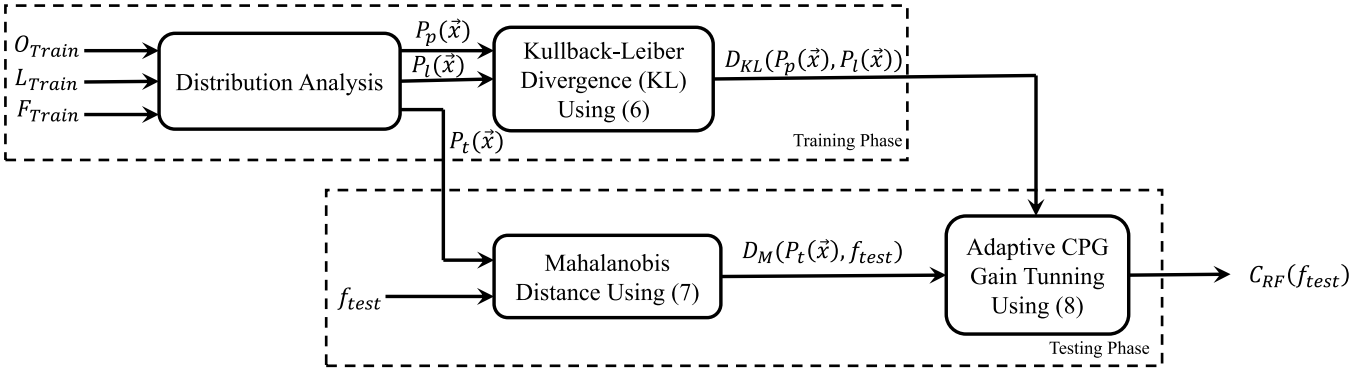


Figure 3: Overview of proposed uncertainty analysis and OOD detection technique.

2.3.1. KL Divergence as Similarity Measure

The Kullback-Leibler (KL) divergence, $D_{KL}(P_l(\overrightarrow{x}), P_p(\overrightarrow{x}))$ is a statistical measure of how a distribution $P_p(\overrightarrow{x})$ is similar to a reference distribution $P_l(\overrightarrow{x})$. The KL divergence between two discrete distributions $P_l(\overrightarrow{x})$ and $P_p(\overrightarrow{x})$ is calculated as

$$D_{KL}(P_l(\overrightarrow{x}), P_p(\overrightarrow{x})) = \sum_{\overrightarrow{x} \in \chi} P_l(\overrightarrow{x}) \log \frac{P_l(\overrightarrow{x})}{P_p(\overrightarrow{x})}. \tag{6}$$

Here, $P_l(\overrightarrow{x})$ and $P_p(\overrightarrow{x})$ are the distributions of the labels and predictions on the training set, respectively. χ is the probability space where distributions are defined. The mathematical details of the distribution analysis for KL divergence is demonstrated in Section 5 (Appendix).

2.3.2. Mahalanobis Distance for OOD Detection

Mahalanobis distance is a statistical measure of the distance between a point and a distribution. Assume a vector $\overrightarrow{v} = [v_i]_{i=1}^{n}$ and a distribution P with mean value of $\overrightarrow{\mu}$ and covariance matrix (Σ). The Mahalanobis distance between vector $\overrightarrow{v}$ and distribution P is

$$D_M(P, \overrightarrow{v}) = \sqrt{(\overrightarrow{v} - \overrightarrow{\mu})^T \Sigma^{-1} (\overrightarrow{x} - \overrightarrow{\mu})}. \tag{7}$$

The covariance matrix Σ is a positive definite matrix; hence its inverse exists and is also positive definite, and the root square is shown in (7) always has a real value.

In this paper, we use (7) to check how far input feature (f_{test}) is to the distribution of the training set $(P_t(\overrightarrow{x}))$ for OOD detection. We calculate the histograms of training data, mean value $\overrightarrow{\mu_f}$ and covariance matrix Σ_f. Then we use (7) to find the distance value $(D_M(P_t(\overrightarrow{x}), f_{test}))$. The distance found using (7) refines the initial uncertainty estimation obtained using (6) in real-time for the test sample. The resulting Mahalanobis distance is used in our proposed adaptable CPGs gain tuning technique described in Section 2.3.3.

2.3.3. Adaptable CPGs Gain Tuning Technique

Our proposed adaptable gain tuning algorithm should have the ability to account for the effect of uncertainty in the CPGs trajectory planning based on the distance between input feature and training distribution. We propose

$$C_{RF}(f_{test}) = \begin{cases} D_{KL}(P_l(\overrightarrow{x}), P_p(\overrightarrow{x})) \times D, & \text{if } D < 1 \\ 0, & \text{if } D \geq 1 \end{cases} \tag{8}$$

as the uncertainty input to be fed into the CPGs for controlling human-robot interaction during task execution. Here, C is a constant value (in the range of 60 to 80, which has been found empirically). $D \triangleq |1 - \frac{D_M(P_t(\overrightarrow{x}), f_{test})}{C}|$, and $D_{KL}(P_l(\overrightarrow{x}), P_p(\overrightarrow{x}))$ and $D_M(P_t(\overrightarrow{x}), f_{test})$ are from (6) and (7). The summarized version of the proposed uncertainty technique for adaptable CPG gain tuning is shown in Algorithm 1. This is important to note that the proposed method only changes the coefficients of some terms in 2nd-order ordinary differential equations (ODEs) shown in (5) at a 20Hz rate. The ODEs will be numerically solved at a higher rate, so in practice, we see only a smooth and continuous gradual transition. The mathematical explanation of the effect of uncertainty in the CPG algorithm is demonstrated in Section 2.2.

Algorithm 1 Proposed Adaptable CPG Gain Tuning

Require: Training feature distribution $(P_t(\overrightarrow{x}))$, Training label distribution $(P_l(\overrightarrow{x}))$, Random forest prediction for training set distribution $(P_p(\overrightarrow{x}))$, Gain constant (C), Training data distribution $\overrightarrow{\mu_{P_t}}$, KL divergence between predictions and labels $D_{KL}(P_l(\overrightarrow{x}), P_p(\overrightarrow{x}))$

1: **for** each f_{test} **do**
2: Find $D_M(P_t(\overrightarrow{x}), f_{test})$ using (7)
3: Find $C_{RF}(f_{test})$ using (8)
4: Update CPG equations and calculate amplitude and frequency of the trajectory using (4) and find $C_{RF}(f_{test})$ using (8)
5: **end for**

3. Results and Discussion

As a showcase of the proposed framework, an exoskeleton was utilized in this paper. The exoskeleton is operated by a human user who applies force to its joints, resulting in additional

torques on the torque sensors. If the interaction torque does not align with the distribution of data on which the model was trained, an unsafe situation may arise. This would result in the model making inaccurate decisions and potentially endangering the user who interacts with the exoskeleton.

We have tested our proposed method on the ExoH3 (Technaid S.L.) exoskeleton, Madrid, Spain. A non-disabled human user with a height of 173 cm and weight of 67 kg wore the exoskeleton while also using crutches as shown in Figure 4. The trajectory of the walking has been saved and Fourier series analysis was conducted on the acquired trajectory to obtain a minimum number of series and the best coefficients. Eight terms of the Fourier series were sufficient and attained coefficients of the hip, knee and ankle motions are listed in Table 1. Additionally, parameters and initial values of the CPG dynamics for hip, knee and ankle joints of both legs are listed in Table 2.

Figure 4: Experimental setup with the human user.

This experiment was planned for two scenarios with and without uncertainty analysis to evaluate the proposed technique's performance in the exoskeleton's control loop. Real-time Desktop MATLAB/Simulink was employed for receiving the sensory data and sending the control commands to the exoskeleton. The sampling frequency was 20 Hz. For this, the exoskeleton was connected to a laptop with a CAN interface (Vector VN1610) running on a Core i7 CPU with 16 GB RAM. We implemented the RFR model on a different PC using Python programming language and scikit-learn machine learning library [35]. We used the UDP communication protocol for sending and receiving data between the PC and the laptop.

We trained the RFR model for learning the passive human-exoskeleton dynamics using our training dataset. The same user was asked to wear the exoskeleton and walk on the ground without interacting with the exoskeleton. The authors are aware that it is hard to make human-robot interaction zero during the experiment. The HRI torque has two parts, intentional and unintentional torque. The intentional torque is used in the adaptable CPG for trajectory shaping. The average value of the torque was subtracted from the experimental torque to make sure that unintentional torque is not included in the experiment. The kinematic data (position, velocity and

	Hip initial motion	Knee initial motion	Ankle initial motion
Coefficients of Fourier series	$a_0 = 5.057$, $a_1 = 6.222$, $a_2 = 0.521$, $a_3 = 0.052$, $a_4 = 0.456$, $a_5 = 0.032$, $a_6 = 0.186$, $a_7 = -0.06$, $a_8 = 0.014$, $b_1 = 19.24$, $b_2 = 5.028$, $b_3 = 1.272$, $b_4 = -0.01$, $b_5 = -0.49$, $b_6 = 0.131$, $b_7 = -0.01$, $b_8 = 0.073$,	$a_0 = 12.28$, $a_1 = 19.88$, $a_2 = 11.76$, $a_3 = 7.327$, $a_4 = 4.888$, $a_5 = 2.369$, $a_6 = 0.957$, $a_7 = -0.38$, $a_8 = 0.180$, $b_1 = -0.95$, $b_2 = -0.37$, $b_3 = 0.707$, $b_4 = 1.083$, $b_5 = 1.231$, $b_6 = 0.962$, $b_7 = 0.405$, $b_8 = 0.115$,	$a_0 = 6.842$, $a_1 = -6.46$, $a_2 = -2.77$, $a_3 = -0.35$, $a_4 = 0.369$, $a_5 = 0.590$, $a_6 = 0.552$, $a_7 = -0.28$, $a_8 = 0.165$, $b_1 = -4.03$, $b_2 = 4.996$, $b_3 = 5.156$, $b_4 = 1.475$, $b_5 = 0.876$, $b_6 = 0.388$, $b_7 = 0.242$, $b_8 = -0.004$,

Table 1: Coefficients of the Fourier series (5) for the hip, knee and ankle initial motions based on the analysis of normal gait trajectories

	Hip, knee and ankle CPGs' parameters
Dynamic parameter values	$\nu_{h-h} = 0.5$, $\nu_{h-k} = 0.5$, $\nu_{k-h} = 0.5$, $\nu_{a-k} = 0.5$, $\nu_{k-a} = 0.5$, $\nu_{a-a} = 0.5$, $\alpha_f = 15\pi$, $\alpha_a = 15\pi$, $\psi = 0.7$, $\lambda = 0.3$, $\eta = 0.3$, $F = 0.35\pi$, $A = 1$
Initial values	$\phi_{right}(0) = 2\ rad$, $\phi_{left}(0) = 2 + \pi\ rad$, $f(0) = 0.1\pi\ rad/s$, $a(0) = 0.1\pi\ rad$

Table 2: Parameter and initial values of CPG dynamics (4) for the hip, knee and ankle joints

acceleration) and the torque values of each exoskeleton joint were recorded during this walking. 80% of the data was used to train the model, and the rest was used for testing. After training the model using 5-fold cross-validation, we reached the average mean absolute error (MAE) value of $0.014(N.m)$ on our test set.

In the first scenario, the user walked with the exoskeleton while interacting with it, but the uncertainty analysis algorithm was turned off. In this scenario, we revised the CPG algorithm to create the desired trajectory for the exoskeleton joints by considering the energy transferred between the user and the exoskeleton. In the second scenario, while the CPG was revised according to the user-exoskeleton energy transfer, we also turned the uncertainty analysis technique on. We evaluated its performance in detecting unsafe decisions and changing CPG's gains when the exoskeleton is making a potentially unsafe decision and taking potentially unsafe action in two user trials. In this way, we evaluated and compared the performance of the uncertainty analysis technique by monitoring the amplitude and frequency of the trajectory created by the CPG with and without the uncertainty gain. We present the experimental results for the first scenario in the time interval between 20 (sec) and 80 (sec) as it is a time interval during which the user interacts with the exoskeleton. The time interval for the second scenario is between 50 (sec) and 250 (sec) as in this time interval, the user interacts with the exoskeleton and applies additional torques to the exoskeleton's joint.

3.1. Scenario one: CPG without Uncertainty Estimation Technique

We used eight terms of typical human gait's hip, knee and ankle motions Fourier coefficients for calculating CPGs' gait trajectory. The walking data were acquired during our experiment to make a training dataset. During the experiment, first, the CPG dynamics calculates the amplitude and frequency of the trajectory using (4) with $C_{RF}(f_{test}) = 1$. Then, the desired trajectory is calculated in (5) using Fourier series coefficients.

A human user wore the exoskeleton as shown in Figure 4. We evaluated the position control algorithm's performance by comparing the exoskeleton points desired and current trajectories. This result is shown for the left and right knees in Figure 5. Figure 5 shows that the exoskeleton is able to follow the desired trajectory created by the CPG dynamics as the user interacts with it.

Next, the user applies active torques on different joints of the exoskeleton to analyze the CPGs' performance in changing the frequency and amplitude of the motions based on the energy transferred between the user and the exoskeleton. Figure 6a shows the total energy transferred between the user and the exoskeleton during the experiment. Figure 6b and Figure 6c show the amplitude and frequency of the desired trajectory created by CPGs following this energy transformation.

The experimental results in Figure 6 show that CPGs is able to change the desired trajectory of the robot as the user applies active torque to the robot's joints. This is implied in the spikes of Figure 6b and Figure 6c that happen when the user adds active torque to the system, shown in Figure 6a.

3.2. Scenario Two: Uncertainty Estimation Technique Performance Analysis

The performance of the proposed uncertainty analysis technique experimented with decisions from exoskeletons that may be unsafe for human users. This may result from irregular changes

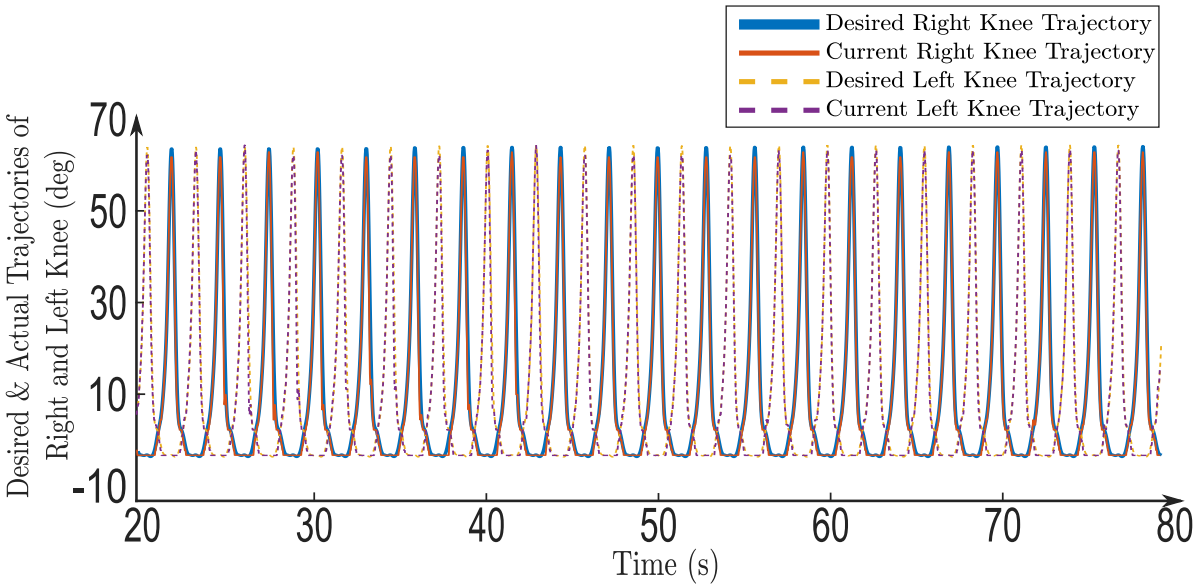
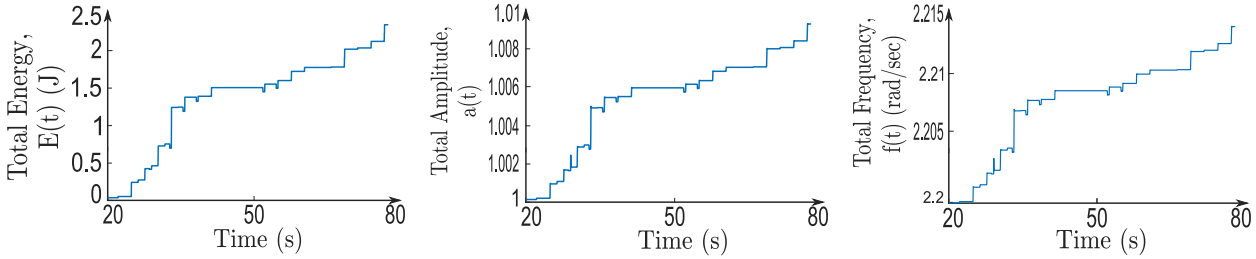


Figure 5: Desired and current trajectories of the right and left knee joints of the exoskeleton.



(a) Energy transferred between user and exoskeleton during an experiment.

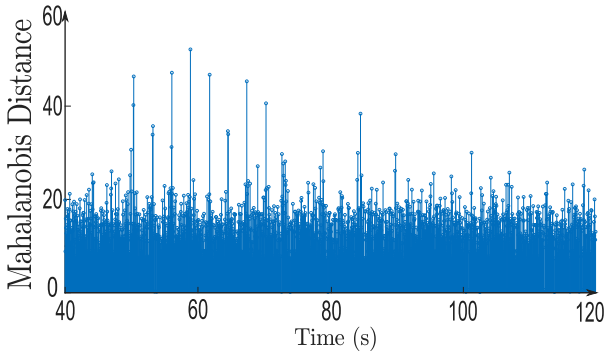
(b) Amplitude of the desired trajectory calculated by CPGs

(c) Frequency of the desired trajectory calculated by CPGs

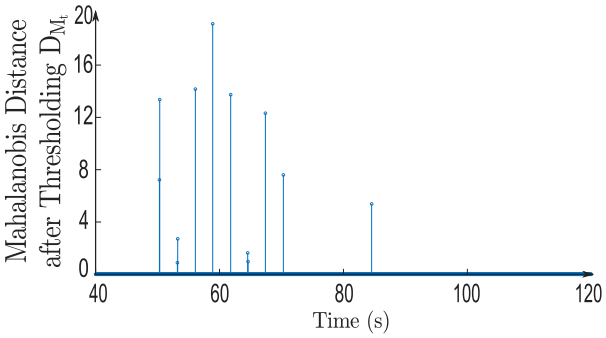
Figure 6: Performance of CPGs in designing trajectory based on energy transferred between user and exoskeleton

in the velocity of the exoskeleton during usage. As mentioned earlier, the training data was collected for walking on the ground with no interaction between the user and the exoskeleton. If the situation changes, the user may apply irregular torque to the joints of the exoskeleton, and CPG may interpret it as a speed-up request from the user. This can be an unsafe decision, and the uncertainty technique should detect it as OOD and stop the exoskeleton from speeding up. We tried to simulate this situation for the exoskeleton during our experiment in two separate user trials to evaluate the performance of the proposed uncertainty analysis technique. During our first user trial, we analyzed the performance of Mahalanobis distance and the thresholding algorithm in detecting unsafe actions. The proposed uncertainty analysis technique monitored the distance between the coming features and training set to inform the exoskeleton when an unsafe situation happens. The Mahalanobis distance between a test sample and the training set is a criterion for the proposed OOD detection algorithm. The distance between test features and training set before (Figure 7a) and after thresholding (Figure 7b) are shown in Figure 7.

The uncertainty detection technique should detect unsafe decisions of the exoskeleton and ap-



(a) Original values of Mahalanobis distance between test features and training set



(b) Mahalanobis distance between test features and training set after thresholding.

Figure 7: Mahalanobis distance between test features and training set (7a) before and (7b) after thresholding for user trial #1.

ply appropriate gains to CPG dynamics to modify its trajectory. The performance of the proposed uncertainty detection technique is shown in Figure 7b. During our first user trials, the distance between input features and the training set jumps when the user applies excessive torques to the exoskeleton, which results in an unsafe decision from the RFR. This situation is simulated in the experiment by asking the user to resist the changes in the trajectory coming from CPG. These jumps were exactly when we asked the user to start applying additional torques to the joint of the exoskeleton. We used two threshold values to keep jumps in Figure 7a and discard the rest. th_{up} and th_{down} are two threshold values that isolate unsafe actions of the user using

$$D_{M_t}(f_{test}) = \begin{cases} S, & \text{if } S < th_{down} \\ 0, & \text{if } th_{up} > S > th_{down} \\ S, & \text{if } S > th_{up}. \end{cases} \tag{9}$$

Here, $S \triangleq D_M(f_{test}) - D_{offset}$ and we chose $th_{down} = -20$ and $th_{up} = 25$ to make sure that the thresholding method only selects unsafe actions and discards smooth interactions between the user and exoskeleton. We found these thresholds experimentally by walking with exoskeleton. The value of offset is $D_{offset} = 15$. The results shown in Figure 7b present a corresponding peak that can be considered as the level of uncertainty in our test feature.

Our proposed adaptable CPG gain tuning technique should limit the trajectory's amplitude and frequency growth whenever the user applies excessive torques to exoskeleton joints, which is caused an unsafe decision from exoskeleton. Furthermore, the gain value should be tuned based on the level of the uncertainty coming from the Mahalanobis distance of the test feature and training set using (8) and (7). We investigated the performance of our proposed adaptable CPG gain tuning during our second user trial. For this purpose, we first need to check whether our algorithm is real-time or not. Figure 8 shows the real-time features of our proposed method as an adaptable CPG gain tuning algorithm is triggered whenever the distance exceeds the tolerance interval. This is shown with lines from the top figure to the bottom figure in Figure 8. Figure 8 shows that our proposed method is able to detect unsafe decision from the kinematic data of the robot in real time, which make it applicable to real-time applications like robotic trajectory shaping.

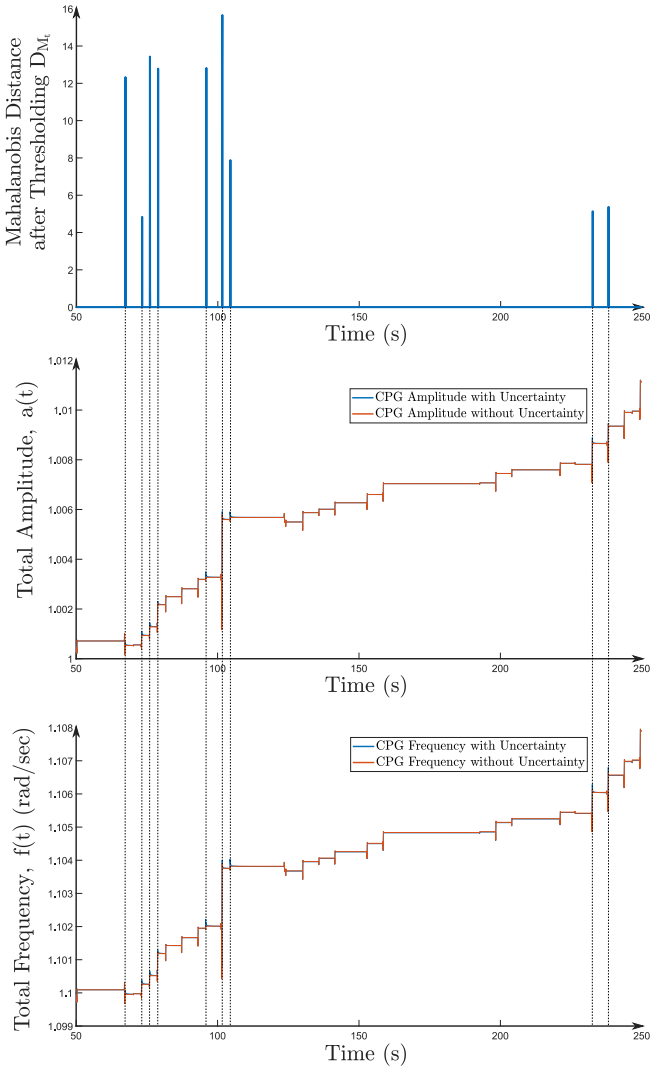
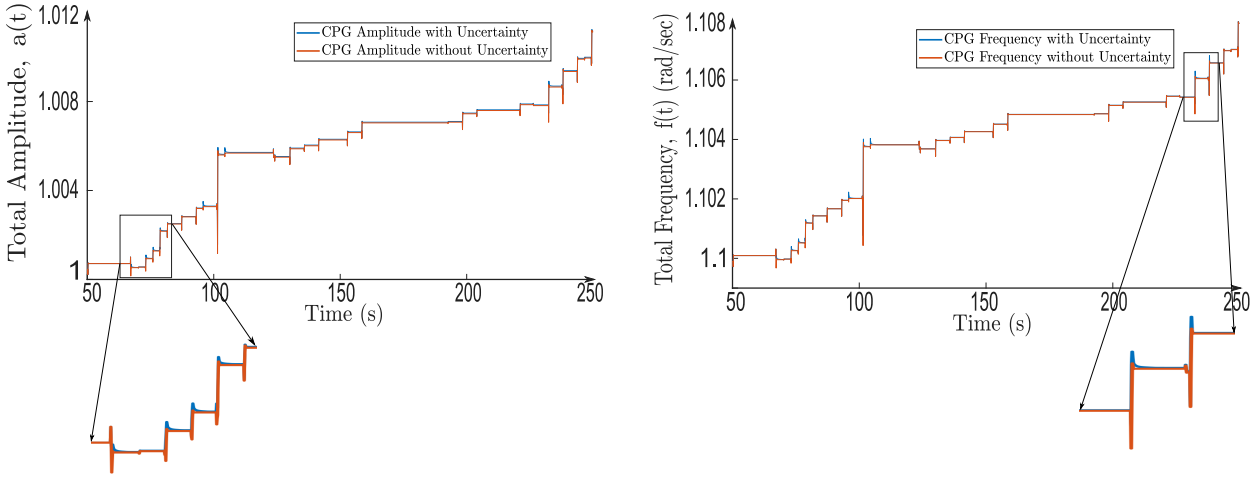


Figure 8: Analyzing real-time performance of proposed method in user trial # 2.

Secondly, we monitored the amplitude and frequency of the trajectory generated by CPG during user trial # 2. The results are reported in Figure 9. The frequency and amplitude shown in Figure 9b and Figure 9a indicate that our proposed uncertainty algorithm is able to detect irregular actions and control the trajectory generated by CPG. The behaviour of the proposed method can be seen in Figure 9 as the blue trajectory, which is an output of the proposed algorithm, cancels the effect of unsafe actions in the trajectory generated by CPG (the blue line does not follow the orange line in the unsafe situations). The proposed method detects irregular jumps, decreases the trajectory's amplitude and frequency, and smoothly converges to the corresponding value (the blue line was shown with the thicker font for demonstration purposes. In the actual result, the two lines are exactly fit to each other). This decrease and smooth convergence in the frequency and amplitude of the trajectory will vanish the effect of unsafe action on the user. Furthermore, a decrease in frequency will control the speed of the exoskeleton. A decrease in the amplitude will control

the gait of the trajectory generated by CPG to enhance the safety of human-robot interaction.



(a) Comparison of amplitude of the desired trajectory with and without uncertainty analysis.

(b) Comparison of frequency of the desired trajectory with and without uncertainty analysis.

Figure 9: Performance of the proposed uncertainty analysis algorithm in adjusting trajectory's amplitude and frequency during user trial # 2.

4. Conclusion and Future Works

Deep learning has been applied in many medical applications. However, the safety and security concerns of using it in the control loop of medical robots have not been thoroughly investigated, which is a safety-critical application of deep learning. In this paper, we proposed a method that can evaluate the uncertainty of the deep learning algorithm in real-time and use this uncertainty measure in the control loop of the robot to inform the system whenever the situation is unsafe for the user. Our proposed method finds the training features and label distributions during the training phase. When the training phase ends, the proposed method finds the distribution of the predictions when the training features feed into the model. The Kullback-Leibler (KL) divergence between predictions and labels is the initial uncertainty of the model. The uncertainty of the prediction for the input feature is updated based on the Mahalanobis distance of the test feature from the training distribution. The calculated uncertainty will update the effect of energy transferred between the user and the robot in the CPGs dynamics. This paper used a random forest regression to estimate the human robot's passive torque.

The proposed method has been tested in the control loop of the ExoH3 (Technaid S.L.) exoskeleton with six degrees of freedom. The experiments were conducted in two scenarios. In the first scenario, we asked the user to walk with the exoskeleton and apply active torque to its joints to revise the CPG dynamics. In the second scenario, we evaluated the performance of the proposed uncertainty analysis technique in two user trials. The proposed technique was able to detect unsafe decisions of the exoskeleton and tuned CPGs gains considering the level of uncertainty in the coming data during both user trials.

Although the proposed method has the ability to reduce the safety concerns in using medical robots for clinical applications, it has some limitations that need to be addressed. In the next steps of this project, we will address the concerns and limitations of the proposed method. The first concern is the energy loss in changing trajectories generated by CPGs. This issue has a direct impact on the ability of our proposed method to satisfy the preferences of the user. The second concern is fixing the user-dependant nature of the training data which can be fixed by state-of-the-art online learning algorithms like reinforcement learning. The last concern is the stability analysis of the gait generated by the CPGs which can be addressed in future. The proposed method can be used as a framework for uncertainty analysis and will be used in different medical robotic applications in the future.

5. Appendix

5.1. Distribution Analysis

Let us define x_i as a sample from $\overrightarrow{x}$, where $\overrightarrow{x}$ is the vector of one feature and x_i is an element of this vector $(x_i \in \overrightarrow{x})$. Then n_{x_i} is the number of times that the input data is in the interval of $[x_i - \varepsilon, x_i + \varepsilon]$. Here ε is the value that controls the length of histogram intervals in our probability function estimation. Then we have

$$\begin{aligned} &\sum_{i=1}^{M} n_{x_i} = N_f \\ &P_{x_i} = \frac{n_{x_i}}{N_f}. \end{aligned} \tag{10}$$

Here, M is the number of distinguished samples and N_f is the total number of samples in the dataset. We need to calculate the distribution of the dataset to find KL divergence using (6), while (10) is only useful for finding the probability of one sample. To solve this problem, we estimate the distribution of (L_{train}) and (O_{train}) using Gaussian fitting as the distribution of the acquired data follows Gaussian distribution shown in Figure 10. Then we have

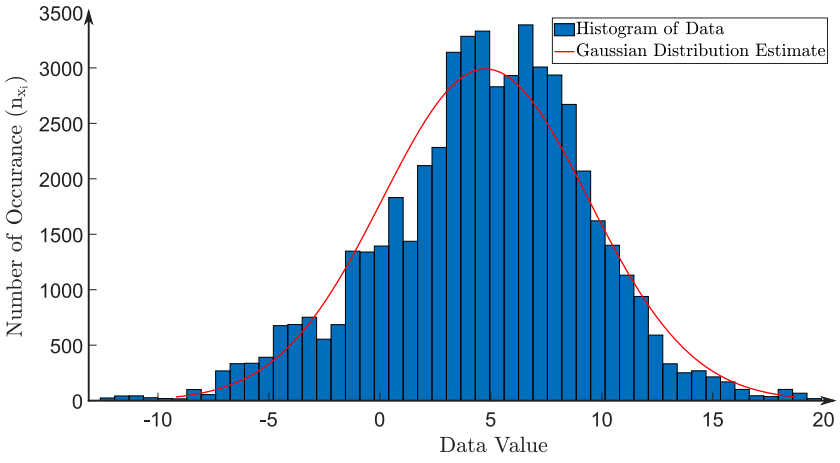


Figure 10: Training dataset with Gaussian distribution fitted to the data.

$$L_{train} \sim \mathcal{N}(\mu_l, \sigma_l), \tag{11}$$

where mean value (μ_l) and standard deviation (σ_l) can be calculated using $\mu_l = \sum_{i=1}^{N} x_i$ and $\sigma_l = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_l)^2}$. We now formalize the distribution of the training labels $P_l(\overrightarrow{x})$ as

$$P_l(\overrightarrow{x}) = \frac{1}{\sqrt{2\pi\sigma_l^2}} \exp \frac{(\overrightarrow{x} - \overrightarrow{\mu_l})^2}{2\sigma_l^2}. \tag{12}$$

We follow the same procedure for finding the distributions of O_{trains} $(P_p(\overrightarrow{x}))$ as

$$\begin{aligned} O_{train} &\sim \mathcal{N}(\mu_p, \sigma_p) \\ \mu_p &= \sum_{i=1}^{N} y_i \\ \sigma_p &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \mu_p)^2}. \end{aligned} \tag{13}$$

We now formalize $P_p(\overrightarrow{x})$ as

$$P_p(\overrightarrow{y}) = \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp \frac{(\overrightarrow{y} - \overrightarrow{\mu_p})^2}{2\sigma_p^2}. \tag{14}$$

Here, y_i is the sample from O_{train}. We calculate the required distributions using (12) and (14), then we use (6) for calculating KL divergence between distributions as our initial uncertainty measure.

References

[1] D. Berend, X. Xie, L. Ma, L. Zhou, Y. Liu, C. Xu, J. Zhao, Cats are not fish: deep learning testing calls for out-of-distribution awareness, in: Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering, Association for Computing Machinery, New York, NY, USA, 2020, pp. 1041–1052.

[2] S. O'Sullivan, N. Nevejans, C. Allen, A. Blyth, S. Leonard, U. Pagallo, K. Holzinger, A. Holzinger, M. I. Sajid, H. Ashrafian, Legal, regulatory, and ethical frameworks for development of standards in artificial intelligence (ai) and autonomous robotic surgery, The international journal of medical robotics and computer assisted surgery 15 (1) (2019) e1968.

[3] X. Zhang, X. Xie, L. Ma, X. Du, Q. Hu, Y. Liu, J. Zhao, M. Sun, Towards characterizing adversarial defects of deep learning software from the lens of uncertainty, in: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, ICSE '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 739–751.

[4] Z. Nado, N. Band, M. Collier, J. Djolonga, M. W. Dusenberry, S. Farquhar, A. Filos, M. Havasi, R. Jenatton, G. Jerfel, et al., Uncertainty baselines: Benchmarks for uncertainty & robustness in deep learning, arXiv preprint arXiv:2106.04015 (2021).

[5] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, et al., A review of uncertainty quantification in deep learning: Techniques, applications and challenges, Information Fusion 76 (2021) 243–297.

[6] M. Ng, F. Guo, L. Biswas, S. E. Petersen, S. K. Piechnik, S. Neubauer, G. Wright, Estimating uncertainty in neural networks for cardiac mri segmentation: A benchmark study, arXiv preprint arXiv:2012.15772 (2020).
[7] M. Ng, Estimating uncertainty in neural networks for cardiac mri segmentation, Ph.D. thesis, University of Toronto (Canada) (2020).
[8] L. Bertoni, S. Kreiss, A. Alahi, Monoloco: Monocular 3d pedestrian localization and uncertainty estimation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6861–6871.
[9] Y. Gal, J. Hron, A. Kendall, Concrete dropout, arXiv preprint arXiv:1705.07832 (2017).
[10] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: international conference on machine learning, PMLR, 2016, pp. 1050–1059.
[11] P. Izmailov, S. Vikram, M. D. Hoffman, A. G. G. Wilson, What are bayesian neural network posteriors really like?, in: International conference on machine learning, PMLR, 2021, pp. 4629–4640.
[12] Y. Wen, P. Vicol, J. Ba, D. Tran, R. Grosse, Flipout: Efficient Pseudo-Independent weight perturbations on Mini-Batches (Mar. 2018). arXiv:1803.04386.
[13] J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, B. Lakshminarayanan, Simple and principled uncertainty estimation with deterministic deep learning via distance awareness, Advances in Neural Information Processing Systems 33 (2020) 7498–7512.
[14] B. Lakshminarayanan, D. Tran, J. Liu, S. Padhy, T. Bedrax-Weiss, Z. Lin, Simple and principled uncertainty estimation with deterministic deep learning via distance awareness (2020).
[15] D. Hendrycks, K. Gimpel, A baseline for detecting misclassified and out-of-distribution examples in neural networks, arXiv preprint arXiv:1610.02136 (2016).
[16] S. Liang, Y. Li, R. Srikant, Enhancing the reliability of out-of-distribution image detection in neural networks, arXiv preprint arXiv:1706.02690 (2017).
[17] K. Lee, K. Lee, H. Lee, J. Shin, A simple unified framework for detecting out-of-distribution samples and adversarial attacks, Advances in neural information processing systems 31 (2018).
[18] P. C. Mahalanobis, On the generalized distance in statistics, National Institute of Science of India, 1936.
[19] D. Hendrycks, M. Mazeika, T. Dietterich, Deep anomaly detection with outlier exposure, arXiv preprint arXiv:1812.04606 (2018).
[20] J. Serrà, D. Álvarez, V. Gómez, O. Slizovskaia, J. F. Núñez, J. Luque, Input complexity and out-of-distribution detection with likelihood-based generative models, arXiv preprint arXiv:1909.11480 (2019).
[21] J. Mena, O. Pujol, J. Vitrià, A survey on uncertainty estimation in deep learning classification systems from a bayesian perspective, ACM Computing Surveys (CSUR) 54 (9) (2021) 1–35.
[22] G. Pang, C. Shen, L. Cao, A. V. D. Hengel, Deep learning for anomaly detection: A review, ACM Computing Surveys (CSUR) 54 (2) (2021) 1–38.
[23] M. Sharifi, J. K. Mehr, V. K. Mushahwar, M. Tavakoli, Adaptive cpg-based gait planning with learning-based torque estimation and control for exoskeletons, IEEE Robotics and Automation Letters 6 (4) (2021) 8261–8268.
[24] S. O. Schrade, Y. Nager, A. R. Wu, R. Gassert, A. Ijspeert, Bio-inspired control of joint torque and knee stiffness in a robotic lower limb exoskeleton using a central pattern generator, in: 2017 International Conference on Rehabilitation Robotics (ICORR), IEEE, 2017, pp. 1387–1394.
[25] K. Gui, H. Liu, D. Zhang, A generalized framework to achieve coordinated admittance control for multi-joint lower limb robotic exoskeleton, in: 2017 International Conference on Rehabilitation Robotics (ICORR), IEEE, 2017, pp. 228–233.
[26] D. Zhang, Y. Ren, K. Gui, J. Jia, W. Xu, Cooperative control for a hybrid rehabilitation system combining functional electrical stimulation and robotic exoskeleton, Frontiers in neuroscience 11 (2017) 725.
[27] J. K. Mehr, M. Sharifi, V. K. Mushahwar, M. Tavakoli, Intelligent locomotion planning with enhanced postural stability for lower-limb exoskeletons, IEEE Robotics and Automation Letters 6 (4) (2021) 7588–7595.
[28] M. Sharifi, J. K. Mehr, V. K. Mushahwar, M. Tavakoli, Autonomous locomotion trajectory shaping and nonlinear control for lower limb exoskeletons, IEEE/ASME Transactions on Mechatronics 27 (2) (2022) 645–655.
[29] M. H. Shaker, E. Hüllermeier, Aleatoric and epistemic uncertainty with random forests, in: International Symposium on Intelligent Data Analysis, Springer, 2020, pp. 444–456.
[30] R. Riener, T. Edrich, Identification of passive elastic joint moments in the lower extremities, Journal of biomechanics 32 (5) (1999) 539–544.

[31] L. Liu, B. J. Misgeld, A. Pomprapa, S. Leonhardt, A testable robust stability framework for the variable impedance control of 1-dof exoskeleton with variable stiffness actuator, IEEE Transactions on Control Systems Technology 29 (6) (2021) 2728–2737.
[32] B. Hwang, D. Jeon, A method to accurately estimate the muscular torques of human wearing exoskeletons by torque sensors, Sensors 15 (4) (2015) 8337–8357.
[33] J. R. Quinlan, Induction of decision trees, Machine learning 1 (1) (1986) 81–106.
[34] S. R. Safavian, D. Landgrebe, A survey of decision tree classifier methodology, IEEE transactions on systems, man, and cybernetics 21 (3) (1991) 660–674.
[35] 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, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.