A Wearable, Patient-Adaptive Freezing of Gait Detection System for Biofeedback Cueing in Parkinson's Disease

Val Mikos, *Student Member, IEEE*, Chun-Huat Heng, *Senior Member, IEEE*, Arthur Tay, Shih-Cheng Yen, Nicole Shuang Yu Chia, Karen Mui Ling Koh, Dawn May Leng Tan, and Wing Lok Au

***Abstract*—Freezing of Gait (FoG) is a common motor-related impairment among Parkinson's disease patients, which substantially reduces their quality of life and puts them at risk of falls. These patients benefit from wearable FoG detection systems that provide timely biofeedback cues and hence help them regain control over their gait. Unfortunately, the systems proposed thus far are bulky and obtrusive when worn. The objective of this paper is to demonstrate the first integration of an FoG detection system into a single sensor node. To achieve such an integration, features with low computational load are selected and dedicated hardware is designed that limits area and memory utilization. Classification is achieved with a neural network that is capable of learning in real time and thus allows the system to adapt to a patient during runtime. A small form factor FPGA implements the feature extraction and classification, whereas a custom PCB integrates the system into a single node. The system fits into a 4.5 × 3.5 × 1.5 cm^3 housing case, weighs 32 g, and achieves 95.6% sensitivity and 90.2% specificity when adapted to a patient. Biofeedback cues are provided either through auditory or somatosensory means and the system can remain operational for longer than 9 h while providing cues. The proposed system is highly competitive in terms of classification performance and excels with respect to wearability and real-time patient adaptivity.**

***Index Terms*—FPGA, freezing of gait, machine learning, patient adaptivity, parkinson's disease, wearable devices.**

I. Introduction

PARKINSON'S disease (PD) is a chronic, progressive and neurodegenerative condition marked by motor related impairments. It is estimated that at least 8.4 million individuals currently suffer from PD, the majority of them being 50 years old or more [1]. Since the disease progresses rather slowly, the struggle of managing the motor related impairments lasts several years. Eventually, PD patients get overwhelmed and require more sick leaves, begin to show signs of absenteeism and opt to retire earlier [2], [3]. Such losses in productivity, coupled with medical health care expenses, are furthermore the cause of substantial societal costs [2], [3]. A primary goal of this research is to help PD patients manage their motor related impairments effectively such that they remain integrated in their social environment for a prolonged period of time.

The motor impairment that arguably poses the biggest challenge to accomplish that goal is known as freezing of gait (FoG). As the name suggests, it describes a patient's inability to initiate, sustain or generally control their gait. An episode is usually confined to a short period of time followed by the patient regaining control and resuming their regular gait. Roughly half of all PD patients are burdened by FoG at least twice a month, while a third experiences episodes on a daily basis [4]. As a direct result of FoG, patients experience a deterioration in mobility, bodily comfort, activity in daily living and emotional well-being [5]. They are furthermore at risk of falls due to FoG [6], [7]. Falls among elderly people result in 20-30% of cases in injuries [8]. This makes the occurrence of FoG a serious health concern for PD patients. It is thus desirable to provide a means for mitigating the effect of FoG and thereby augmenting a patient's quality of life [9].

Wearable, biofeedback cueing devices are one such promising proposal to accomplish exactly that. These systems detect the occurrence of a FoG episode in real-time and provide biofeedback cues to the patient in an attempt to help them overcome the episode and hence reduce the risk of falls. The supply of such biofeedback cues to a PD patient has shown to reduce the overall severity of FoG [21]–[23], as well as shorten the total duration of FoG by 34% [24]. Unfortunately, a plethora of research focuses on FoG classification algorithms built only in software without providing a wearable hardware implementation for biofeedback cueing. The recent proposals for machine learning (ML) FoG classifiers are prime examples of FoG detection systems that are solely designed in software [25], [26]. Mapping these algorithms onto hardware is not straightforward since wearable, battery-powered electronics have substantial power and area constraints that need to be met. It is thus not surprising that the few FoG detection systems that do provide a hardware implementation are rather bulky and obtrusive to the patient when worn. Nevertheless, there are several recent studies that

Manuscript received December 5, 2018; revised February 18, 2019; accepted April 26, 2019. Date of publication May 1, 2019; date of current version May 24, 2019. This work was supported in part by NUS-NNI 2016 under Grant R263000C36133 and in part by NMRC/CISSP/2014/2015. This paper was recommended by Associate Editor R. Jafari. *(Corresponding author: Val Mikos.)*

V. Mikos, C.-H. Heng, A. Tay, and S.-C. Yen are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 (e-mail: mikos.val@gmail.com; elehch@nus.edu.sg; arthurtay@nus.edu.sg; shihcheng@nus.edu.sg).

N. S. Y. Chia, K. M. L. Koh, and W. L. Au are with the Department of Neurology, National Neuroscience Institute, Singapore 308433 (e-mail: Nicole_CHIA@nni.com.sg; karenkoh7@gmail.com; au.wing.lok@singhealth.com.sg).

D. M. L. Tan is with the Department of Physiotherapy, Singapore General Hospital, Singapore 169608 (e-mail: dawn.tan.m.l@sgh.com.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBCAS.2019.2914253

TABLE I
OVERVIEW OF VARIOUS PROPOSED FOG DETECTION SYSTEMS IN LITERATURE (AS OF LATE 2018)

Work	Hardware					Algorithm			Performance			
	BF	📱	N_s	Weight	Battery	ML	N_f	Interval	Data	Sens	Spec	Split
Bächlin et al. (2010, [10]–[12])	A	✗	2	297g	300 mAh 6 hours	✗	2	16 ms	10	73.1% 87.1%	81.6% 86.9%	✗ 🧍
Mazilu et al. (2012, [13])	A,S	✓	3	174 g	300 mAh 6 hours	✓	7	16 ms	10	62.1% 98.4%	95.2% 99.7%	✓ 🧍
Pepa et al. (2015, [14])	A	✓	1	137 g	📱	✗	3	400 ms	18	82.3%	76.8%	✗
Kim et al. (2015, [15])	A,S	✓	1	130 g	📱	✓	12	1000 ms	9	86.0%	91.7%	✓
Lorenzi et al. (2016, [16], [17])	A	✗	2	25 g	90 mAh 2 hours	✗	1	16 ms	16	94.5%	96.7%	✗
Ahn et al. (2017, [18])	V	✗	1	212 g	900 mAh 6 hours	✗	1	stride	10	97.0%	88.0%	✗
Punin et al. (2017, [19])	S	✓	3	170 g	500 mAh N.A.	✗	2	750 ms	8	86.7%	60.6%	✗
Kita et al. (2017, [20])	A	✓	3	155 g	90 mAh 2 hours	✗	1	400 ms	32	97.6%	93.4%	✗
This paper (2019)	A,S	✗	1	32 g	800 mAh 9+ hours	✓	3	20 ms	63	95.6%	90.2%	✓

LEGEND: BF: biofeedback auditory (A), sematosensory (S), visual (V) 📱: smartphone used to run algorithm, N_s: number of sensors nodes worn, **Weight**: total hardware weight. **Battery**: capacity and lifetime of bottleneck node. **ML**: machine learning based or not, N_f: number of features employed, **Interval**: time between successive classifications. **Data**: number of Parkinson's disease patients, **Sens**: sensitivity, **Spec**: specificity, **Split**: Split dataset into training and test set, or cross-validated. 🧍: indicates patient-specific classification. **DISCLAIMER:** Weight either given in reference, obtained through personal correspondence or estimated.

have successfully mapped complex, real-time capable classification algorithms onto dedicated hardware for wearable health care applications. For example, FPGA based designs were proposed for cardiovascular disease monitoring [27], [28], seizure detection [29] and assistive technologies [30]. Learning on wearable hardware with support vector machines has also been proposed to provide patient adaptivity in seizure detection [29]. In light of these recent advances, this paper demonstrates the first ML based FoG detection system integrated into an FPGA based wearable sensor node. It implements a neural network classification algorithm and allows for real-time, online learning on the sensor node to provide patient adaptivity. The paper details the design efforts and the compromises that made the employment of small, wearable hardware possible while remaining as competitive as possible in terms of classification performance.

The paper first provides a brief overview of published FoG detection systems in Section II and illustrates the research gap that ought to be addressed. In Section III, features are selected that minimize the computational load on the hardware while retaining a high classification accuracy. Section IV then demonstrates the proposed dedicated hardware that allows for learning of the algorithm on wearable hardware in real-time. The overall FoG detection system and thus the successful mapping of the functionality onto a single sensor node is demonstrated in Section V followed by a performance evaluation in Section VI. Finally, the main findings are summarized in the conclusive Section VII.

II. REVIEW OF FREEZING OF GAIT DETECTION SYSTEMS IN LITERATURE

Table I summarizes hardware based FoG detection systems found in literature, all of which employ inertial measurement units (IMUs) to classify FoG. It distinguishes three parts that are of particular interest when discussing FoG detection systems: hardware, algorithm and performance.

The hardware summary shows that FoG detection systems predominantly deliver biofeedback (BF) using auditory (A) or sematosensory (S) cues [10]–[15], [17], [19], [20], with only a single implementation opting to provide visual (V) cues [18]. The hardware that computes the classification result and delivers these cues is commonly a smartphone [13]–[15], [19], [20]. Exceptions are Bächlin *et al.*'*s* waist-worn, portable PC using an Intel XScale family processor [10]–[12], Ahn *et al.'s* Epson BT-200 smart glasses providing visual cues [18] and Lorenzi *et al.*'*s* proposed dedicated hardware for FoG detection [17]. All non-dedicated hardware implementations and smartphone reliant FoG detection systems weigh 130 g or more, while Lorenzi *et al.*'*s* dedicated hardware reduces weight by more than 5× to a miniscule 25 g. Such compact size is achieved at the expense of limited battery life of 2 hours and its inability to process any ML algorithm. In addition, a wireless communication between multiple nodes needs to be maintained which could incur reliability issues. Although there are systems that integrate the overall design into a single node, they rely on bulky, usually waist-worn smart devices to do so

[14], [15], [18]. This calls for an energy efficient hardware innovation with integrated FoG detection on a single sensor node.

Regarding the algorithmic aspect, most systems rely on the computationally inexpensive threshold based approach to FoG classification. The only ML based FoG classifiers are Mazilu *et al.*'*s* random forest [13] and Kim *et al.*'*s* boosted ensemble classifier [15], which require the computational power of a smartphone. The recent algorithmic advances in ML based FoG classification employing neural networks [25] or support vector machines [26] have yet to be implemented in wearable hardware. The reported systems also differ in their interval choice between two successive classifications, with some classifying every IMU data sample [10]–[13], [17] and others adopting larger time intervals [14], [15], [19], [20]. Nevertheless, too large intervals between successive classifications should be avoided to ensure a timely FoG classification.

Finally, FoG detection systems are only viable if the classification performance is satisfactory. That performance has been evaluated by determining the sensitivity and specificity when classifying datasets consisting of roughly a dozen patients in most cases [10]–[15], [17]–[19] and 32 patients at most [20]. All threshold based approaches were built and evaluated using the complete dataset, and the attained performance metrics are thus best regarded as training performance only. The average of the attained sensitivity and specificity ranges from 73.6% to 95.6%. On the other hand, ML implementations build their model based on a training dataset and evaluate the performance on a disjunct test dataset. The achieved average of sensitivity and specificity of 78.6% and 88.8% are thus more objective as they are obtained from an unseen test dataset. Patient-specific classifiers are also another interesting aspect to explore. Compared to classifiers that generalize on all patients, patient-specific classifiers have shown significant improvements in sensitivity and specificity [10]–[13]. This suggests that the design of patient-specific or patient-adaptive FoG detection systems could be a worthwhile endeavor.

This paper addresses the limitations mentioned above. It demonstrates a ML based classification algorithm, provides real-time patient adaptivity and integrates the functionality with dedicated hardware into a single sensor node. The following sections illustrate the design methods and compromises adopted that enable such an integration while remaining competitive in terms of classification accuracy.

III. Hardware Constrained Feature and Algorithm Selection

The real-time extraction of features in hardware can be costly. While software based implementations can virtually add as many features as desired to a classification algorithm, hardware implementations will face repercussions in terms of power consumption and area requirements. This section illustrates the selection of features and classification algorithm that reduces the number of employed features and thus the computational load on the hardware.

A. Candidate Features and Dataset

When limiting the number of employed features, it becomes crucial to select those that are highly potent in detecting FoG. In order to not overlook potentially potent features during the selection process, a list of candidate features is compiled first from a wide range of proposals found in literature. We have conducted a thorough survey of FoG related features in [31] and employ them here as the set of candidate features. They consist of 117 features extracted from various IMU signal axes (accelerations $a_x, a_y, a_z, |a|$, and angular velocities $\omega_x, \omega_y, \omega_z, |\omega|$). They can be divided into the following three categories. First are features based on frequency domain computations, such as freeze index (FI), freeze band (FB), locomotor band (LB), mean frequency ($\bar{f}$), dominant frequency (DF), power (P) and spectral entropy (S). The second category includes features derived from arithmetic operations on IMU data, namely root mean square ($\sqrt{.^2}$), mean ($\bar{.}$), standard deviation (σ), coefficient of variation (CV), kurtosis (K), maximum acceleration ($\lceil . \rceil$), range of acceleration (R) and stride peak (SP). And lastly are features that rely on spatio-temporal domain algorithms, such as stride time (ST), velocity (V), stride length (SL) and the FoG criterion (FoGC).

The selection of features from this set of candidate features is based on a dataset consisting of 25 PD patients who wore an IMU on their ankles ($f_s = 50$ Hz) during 7-meter timed up-and-go exercises. The dataset amounts to a total of 1 hour and 33 minutes of gait data with 221 FoG episodes. The average duration of a FoG episode was 10.2 s. We employed the same dataset in [31] to select optimal feature sets for various machine learning algorithms. Here, we propose a new feature selection approach specifically for hardware implementations by limiting the number of features and thus the computational load on the hardware.

B. Minimal Feature Set Selection

Although software implementations of ML based FoG classifiers employ as many as sixty-four unique features [25], hardware implementations relied on seven [13] and twelve [15] only. Furthermore, both of these hardware implementations relied on the processing power of a bulky smartphone (see Table I) and it is questionable whether a smaller, battery-powered sensor node can accommodate as many. Thus, it is desirable to select as small a set from the candidate features as possible without conceding any significant loss in terms of classification potency. This is accomplished by the feature selection process illustrated in Fig. 1. First, a conventional wrapper method selects optimal feature sets for various ML algorithms from the set of candidate features. This results in an optimal feature set of N_{opt} features for each ML algorithm. Then, a greedy stepwise search algorithm attempts to form a smaller feature set with similar classification performance as observed with the optimal feature set just derived. Initially, its set of selected features is empty. It then executes a step by adding one feature from the set of candidate features to its set of selected features. The decision about which feature it selects falls onto the feature that maximizes the classification accuracy when added to the set of selected features. If the

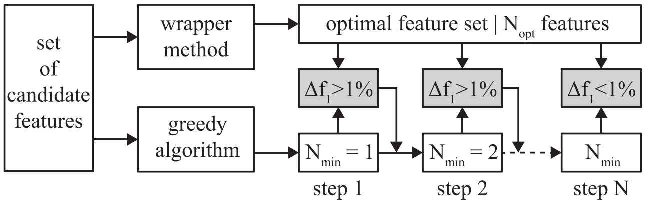


Fig. 1. Feature selection approach to find feature sets with as small a number of features as possible while retaining a high classification performance in terms of f_1-score.

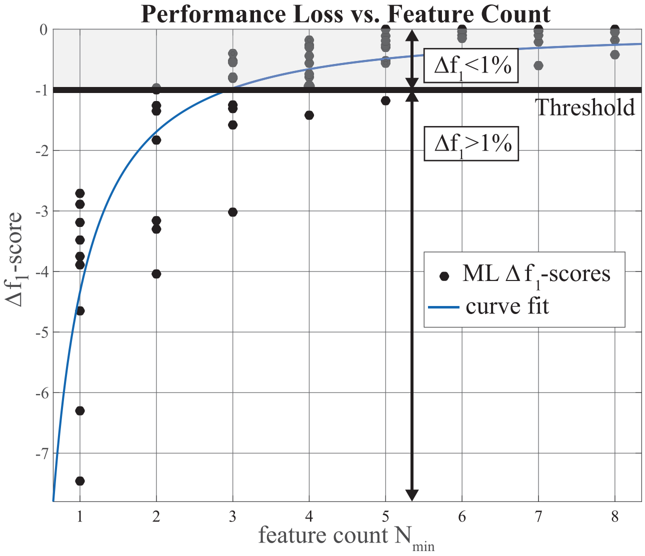


Fig. 2. The gradual improvement of adding more features using a greedy stepwise search. For illustrative purposes, feature count N_{min} is increased even after f_1-score performance has been reached.

classification accuracy with that newly added feature is within 1% of the classification performance observed with the optimal feature set in terms of f_1-score, then the algorithm concludes. Otherwise, it continues to execute further steps by adding more features until the classification performance is comparable to the one observed with the optimal feature set, or until N_{opt} features have been added. Once the greedy algorithm concludes, an alternative feature set consisting of N_{min} features is obtained, where $N_{min} \leq N_{opt} \leq 117$.

C. Selected Features and Classification Algorithm

The proposed minimal feature set selection was implemented using Weka 3.8. The results are illustrated in Fig. 2. The fitting curve to the f_1-scores indicates that, on average, three features suffice to achieve a classification performance similar to the one observed when using the optimal feature set. The number of selected features in the optimal and minimal sets is summarized in Table II for various ML algorithms. Optimal feature sets range in size from two features for naive Bayes up to seventeen for boosted decision trees. Overall, the proposed minimal feature selection procedure succeeded in all but two cases to reduce that number of features. It needs to be noted that even if the minimal feature set concludes with the same number of features as the optimal one, it is most likely inferior in classification performance. This is because the greedy algorithm is only considering the effect of adding one more feature to a set of already selected features, while the wrapper approach is more flexible in that it evaluates various combinations of any number of features to build a feature set. Nevertheless, the minimal feature sets eventually fall within 1% of the optimal f_1-score performance, the exception being the k-nearest neighbor classifier which required $N_{min} = 6 > N_{opt}$ to do so.

Since the core motivation is to minimize the computational load on the hardware, the feature set with the smallest extraction cost ought to be selected rather than the set with the lowest number of features. To that end, the estimated extraction cost of the selected feature sets is provided in Table II (see Appendix A for derivation). It illustrates the number of additions (subtractions), multiplications, divisions (Div) and non-linear (NL) functions needed to extract the derived feature set. Based on these findings, a neural network approach is chosen for our implementation. Besides the low computational load, neural networks can also conduct online learning which is critical to enable patient adaptivity in real-time. Moreover, its potency has been clearly demonstrated in software [25] and porting it into a miniaturized, energy efficient sensor node would prove its feasibility for FoG detection systems. To facilitate online learning and counteract the vanishing gradient problem, one hidden layer with five units is implemented here. The features that are thus extracted for the proposed FoG detection system are as follows. The freeze index (FI), proposed in [32], is given by

$$\mathrm{FI}_{|a|} = \frac{\int_3^8 |A(f)|^2 \, df}{\int_{0.5}^3 |A(f)|^2 \, df}, \tag{1}$$

where $A(f)$ is the 128-point discrete Fourier transform (DFT) of the IMU's most recent acceleration's absolute values $|a|$. During FoG, high frequency related tremors arise due to the loss of control over one's lower limbs. The increase of high frequency components within the spectrum is captured by the numerator, which enlarges the freeze index. On the other hand, regular gait has a distinct frequency peak around 1 Hz. Such frequencies are captured by the denominator, which diminishes the FI. Hence, the FI is expected to increase during FoG episodes, but remain small during regular gait. The second feature, stride peak (SP), was proposed in [31] as

$$\mathrm{SP}^{(i)} = \begin{cases} \omega_z^{(i-1)} & \text{if } \left(\omega_z^{(i-2)} < \omega_z^{(i-1)} > \omega_z^{(i)}\right), \\ \mathrm{SP}^{(i-1)} & \text{otherwise.} \end{cases} \tag{2}$$

Here, $\omega_z^{(i)}$ is the i^{th} sample of the IMU's angular velocity in the patient's frontal plane. Wide swings are inherent to regular gait patterns, but during FoG episodes these swings considerably subside. Smaller SP values thus capture the abatement of swings and indicate a potential FoG episode. For the third and final feature, rather than extracting the standard deviation, σ_{ω_z}, the computational complexity is furthermore reduced by extracting the variance instead. This eliminates one non-linear function which is especially cumbersome to implement in hardware. It is

TABLE II
RESULTING FEATURE SETS AND FEATURE COUNT REDUCTION USING THE PROPOSED FEATURE SELECTION PROCEDURE. THE EXTRACTION COST OF THE FEATURE SETS ARE INDICATED

Classifier	$\boldsymbol{\Delta f_1}$	**#Features**		**Features**	**Extraction Cost**			
		$\boldsymbol{N_{opt}}$	$\boldsymbol{N_{min}}$		**Add**	**Mult**	**Div**	**NL**
Single Models								
Support Vector Machine	$< 1\%$	8	3	$\mathrm{FI}_{\lvert a\rvert}$, FI_{w_z}, LB_{ω_z}	32'922	32'840	2	3
Neural Network	$< 1\%$	12	3	$\mathrm{FI}_{\lvert a\rvert}$, σ_{ω_z}, SP	14'756	14'593	1	2
Logistic Regression	$< 1\%$	11	4	$\mathrm{FI}_{\lvert a\rvert}$, $\sqrt{\omega_z^2}$, $\overline{f_{\omega_z}}$, SP	19'206	19'153	2	3
Naive Bayes	$< 1\%$	2	2	$\mathrm{FI}_{\lvert a\rvert}$, LB_{ω_z}	18'370	18'320	1	2
k-Nearest Neighbor	$> 1\%$	4	4	$\mathrm{FI}_{\lvert a\rvert}$, LB_{ω_z}, R_{ω_y}, $\lceil\omega_z\rceil$	18'774	18'320	1	2
Decision Tree (C4.5)	$< 1\%$	13	3	FI_{a_x}, LB_{ω_z}, $\lceil\omega_z\rceil$	18'558	18'320	1	2
Ensemble Methods								
Bagging (C4.5)	$< 1\%$	5	4	$\mathrm{FI}_{\lvert a\rvert}$, LB_{ω_z}, SL, DF_{a_y}	23'013	24'450	53	3
Boosting (C4.5)	$< 1\%$	17	4	$\mathrm{FI}_{\lvert a\rvert}$, LB_{ω_z}, SL, ST	21'933	23'379	53	2
Random Forest	$< 1\%$	8	4	FI_{ω_z}, LB_{a_x}, SP, ST	18'394	18'338	1	2

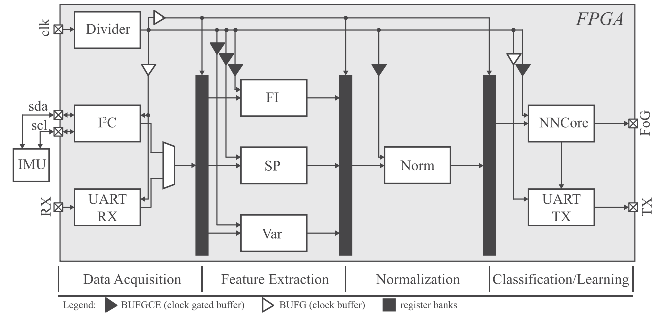


Fig. 3. Overview of the proposed VLSI design of a neural network and feature extraction that is mapped onto an FPGA.

also computed from the IMU's angular velocity readings ω_z, as in

$$\mathrm{Var}_{\omega_z} = \frac{1}{N} \sum_{j=i-N+1}^{i} \left(\omega_z^{(j)} - \overline{\omega_z}\right)^2 \quad \Big| \quad N = 64 \tag{3}$$

The window length N has been deliberately set to a power of 2, as this reduces the division in (3) to a simple bit shift operation in hardware. During regular gait, wide swings engender large variations in observed angular velocity values, while a more rigid, constrained gait is confined to smaller variations. This is again exploited to detect FoG. With the selected features and ML algorithm, the paper next demonstrates the design of hardware for a neural network that is capable of learning in real-time on a sensor node.

IV. Hardware Design of a Patient Adaptive Neural Network Classifier

One of the main goals of this paper is to transfer learning from a desktop PC or cloud computer onto a wearable, battery-powered sensor node using dedicated hardware. It is necessary to limit area and power consumption when mapping the learning algorithm onto such wearable hardware. The proposed VLSI design of a neural network capable of operating under these limitations is illustrated in Fig. 3. The goal is to fit the design onto a small form factor FPGA for later integration into a wearable sensor node.

A. Data Acquisition and Feature Extraction

The proposed design first acquires a patient's gait data from an off-chip IMU using the I^2 C protocol. A UART interface is furthermore supported in order to allow for the receiving and sending of data via BLE. Once a data sample from the IMU has been acquired, the three aforementioned features (FI, SP, Var) are extracted. We have demonstrated the extraction of these three features in detail in our previous work [33]. The computations of (2) and (3) are relatively straightforward and do not require substantial hardware resources. However, the extraction of the FI with (1) is computationally expensive due to the required DFT. Area utilization has been significantly reduced by relying on a spectrum analyzer proposed in [34] to compute the DFT. The spectrum analyzer was implemented with as little as three adders and two multipliers that are time-shared among the necessary computations. Once the features are extracted, they are normalized and routed as a feature vector (FV) to the neural network core shown in Fig. 4. The neural network core consists of four computational blocks using three RAM memories to classify the feature vector for biofeedback generation and provide patient adaptivity through real-time online learning.

B. Real-time Classification For Biofeedback Cueing

The feed forward processing unit (FFPU) is the only processing block required for classification. All other blocks within the core are accommodating the online learning functionality. Forward propagating a sample in a neural network requires the computation of the induced field $z_j^{(l)}$ and the corresponding activation $a_j^{(l)}$ for every j^{th} unit of every l^{th} layer, as in

$$z_j^{(l)} = \sum_{i=0}^{u^{(l-1)}} \theta_{j,i}^{(l-1)} \cdot a_i^{(l-1)}, \qquad \forall j \in [1, u^{(l)}] \tag{4}$$

$$a_j^{(l)} = \sigma\left(z_j^{(l)}\right), \qquad \forall j \in [1, u^{(l)}] \tag{5}$$

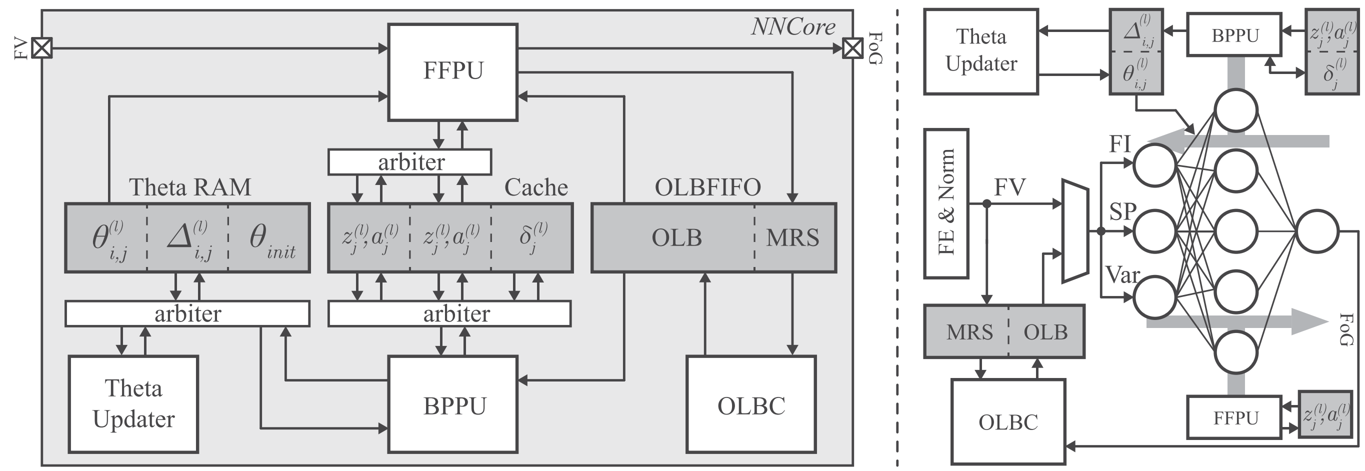


Fig. 4. Neural network accelerator with online learning capability. *Left*: Block diagram of the core processor consisting of three RAM blocks (Theta RAM, Cache, OLBFIFO) and four main computation blocks (FFPU, BPPU, OLBC, Theta Updater). *Right*: A computational flow diagram illustrating the neural network along with the role of the various RAM and computational blocks.

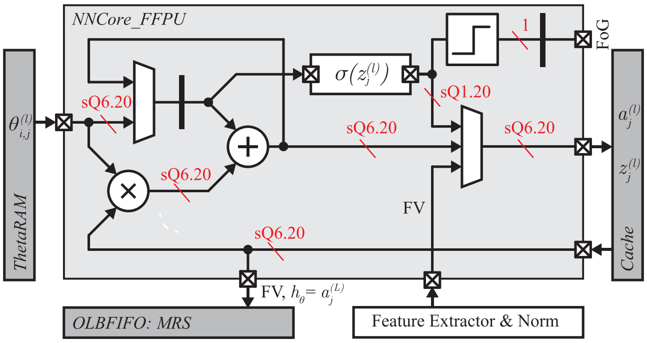


Fig. 5. The FFPU relies on two time-shared DSP blocks and an LUT storing the sigmoid activation function.

where $u^{(l)}$ denotes the number of units in layer l, $\sigma(.)$ the sigmoid function and $\theta^{(l-1)}_{j,i}$ the neural network parameter from unit i in layer $l-1$ to unit j in layer l. Note that every zeroth unit is a bias unit, as in $a^{(l)}_0 = 1$ for all layers l. The VLSI implementation of the FFPU is shown in Fig. 5. It time-shares one adder and multiplier to reduce area utilization and implements the sigmoid function, $\sigma(.)$, with a lookup table (LUT). The LUT stores 256 entries with 22-bit fixed point precision (sQ1.20), which was sufficient to ensure accurate classification. Since the neural network parameters $\theta^{(l)}_{j,i}$, induced fields $z^{(l)}_j$ and activations $a^{(l)}_j$ need to be shared among multiple processing blocks, the RAM blocks are implemented with dual ports to allow for simultaneous access. The FFPU computes the fields $z^{(l)}_j$ and activations $a^{(l)}_j$ unit by unit and writes the results into the cache. An arbiter ensures that the results are written into a section of the cache that other processing blocks are currently not reading from. Once the FFPU computes the activation for the output unit in the final layer L, $a^{(L)}_j$, it is rounded to obtain the classification result and outputted as the signal *FoG* to control biofeedback cueing. That activation value is referred to as the FFPU prediction score, $h_{\boldsymbol{\theta}} \in (0, 1)$, and is written into the most recent samples (MRS) section of the OLBFIFO RAM along with the corresponding feature vector for online learning purposes.

C. Online Learning Batch Creator (OLBC)

Once a feature vector has been classified by the FFPU, the neural network core begins with learning. First and foremost, the learning algorithm requires data to be trained with. The task of assembling such training data is handled by the online learning batch creator (OLBC). The OLBC reads the feature vectors stored in the most recent samples section of the OLBFIFO RAM to create and manage the online learning batch (OLB). That online learning batch is then used to train the neural network. Unfortunately, the batch cannot store the complete training dataset as software implementations do. The available on-chip memory on wearable hardware is far from sufficient to do so. Learning on wearable hardware thus requires to be done with smaller learning batches and feature vectors need to be discarded in a first-in first-out (FIFO) fashion to make space for new training data. Smaller learning batches result in a more stochastic learning progress instead of a smooth one, but should not prevent the ML algorithm from adopting a potent, final model. Therefore, the online learning batch is assembled with 32 feature vectors only, with one half reserved for feature vectors representing FoG and one half for feature vectors representing regular gait. The OLBFIFO RAM with its two sections, the online learning batch and the most recent samples, is illustrated in Fig. 6. It shows the online learning batch's 32 feature vectors and the schematic of the OLBC that decides which feature vectors from the most recent samples section are added to the online learning batch and which are not. The inclusion of feature vectors into the online learning batch depends on whether supervised or unsupervised learning is selected.

1) Supervised Learning: For supervised learning, the correct label (FoG or no FoG) of the feature vector is known and supplied along with the feature vector. This is possible if the patient is part of the dataset that the neural network is being trained with or if

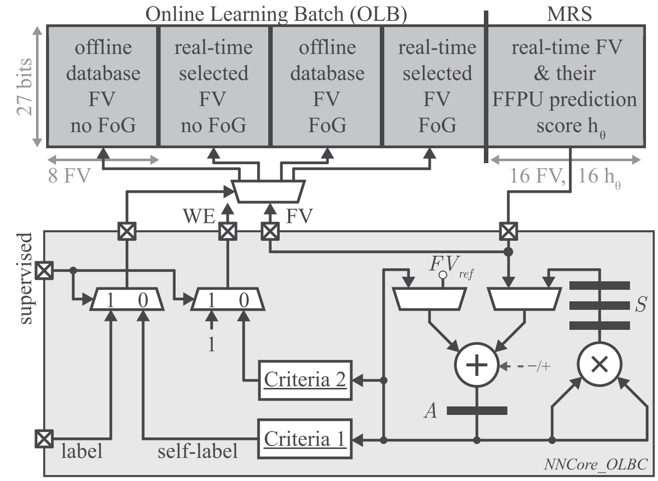

Online Learning Batch (OLB)
MRS
offline database FV no FoG
real-time selected FV no FoG
offline database FV FoG
real-time selected FV FoG
real-time FV & their FFPU prediction score h_θ
27 bits
8 FV
16 FV, 16 h_θ
WE
FV
supervised
FV_ref
1 0
1 0
1
S
Criteria 2
Criteria 1
−/+
A
label
self-label
NNCore_OLBC

Fig. 6. The online learning batch creator reads the feature vectors from the OLBFIFO’s most recent samples (MRS) section and decides if they are added into the online learning batch (OLB).

the labels are provided in real-time, e.g. through a BLE enabled push button while the patient is walking. Fig. 6 shows that if supervised learning is selected, the OLBC routes the provided label through to the online learning batch multiplexer and sets the write enable (WE) to 1. Thus, when labels are provided the OLBC indiscriminately adds feature vector after feature vector into the online learning batch. According to the provided label, the feature vector is written into any section of the online learning batch labeled “no FoG” or “FoG”.

2) Unsupervised Learning: If no label is provided, the neural network core allows for a self-training mechanism based on our proposed software algorithm in [35]. It demonstrates two criteria that the OLBC can employ to self-label the feature vectors with a high degree of confidence. Criteria 1 computes the mean FFPU prediction score, $\overline{h_{\boldsymbol{\theta}}} \in (0, 1)$, of the most recent feature vectors. If that mean is highly positive, or highly negative, the OLBC self-labels the feature vectors according to

$$\hat{\boldsymbol{y}} = \begin{cases} \mathbf{1}, & \overline{h_{\boldsymbol{\theta}}\,(FV)} > 0.85 \\ \mathbf{0}, & \overline{h_{\boldsymbol{\theta}}\,(FV)} < 0.25 \\ \text{discard}, & \text{otherwise}. \end{cases} \tag{6}$$

A total of 16 FV-$h_{\boldsymbol{\theta}}$ pairs are available in the most recent samples section of the OLBFIFO RAM. Thus, the OLBC computes the mean by accumulating the individual prediction scores in register *A* and bit shifting the result. Criteria 2 triggers the write enable for those feature vector that are not too far off from a known reference FV_{ref}. For example, if a feature vector is self-labeled as FoG using criteria 1, then the feature vector needs to satisfy

$$(\mathrm{FI} - \mathrm{FI}_{ref})^2 + (\mathrm{SP} - \mathrm{SP}_{ref})^2 + (\mathrm{Var} - \mathrm{Var}_{ref})^2 < lim \tag{7}$$

in order to be included into the online learning batch. The OLBC stores the squared terms of (7) in register *S* before accumulating them in register *A*. The motivation behind the two criteria are thoroughly described in [35]. If both criteria are met, the OLBC proceeds to write the feature vector into the online learning batch’s real-time section of “FoG” or “no FoG” according to the determined label. Note that for unsupervised learning, the online learning batch reserves half of all feature vectors for offline database feature vectors that have a known label. These feature vectors can only be overwritten during supervised learning. Hence, 50% of the online learning batch does not change during unsupervised online learning and at least 50% of the online learning batch has definitely a correct label assigned to it.

D. Online Batch Learning With Backpropagation

With training data assembled in the online learning batch, the neural network can begin with learning. Learning in neural networks is the search for suitable parameters $\boldsymbol{\theta}$ that lead to a high classification accuracy on the online learning batch training data. Here, a stochastic gradient descent algorithm searches for suitable parameters by iteration of

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - \alpha \frac{\partial}{\partial \boldsymbol{\theta}_n} J(\boldsymbol{\theta}) \tag{8}$$

where α is the learning rate, n the iteration index and $J(\boldsymbol{\theta})$ the cost function of classifying training samples incorrectly. For the proposed implementation, the cross-entropy cost function is adopted and one gradient descent iteration is conducted per assembled online learning batch. Hence, learning requires the computation of the partial derivative of the cost function. The backpropagation processing unit (BPPU) is designated to compute these partial derivatives for all units in the network. For every feature vector in the online learning batch, it quantifies the error $\delta_j^{(L)}$ at the output layer L as the difference between the FFPU’s prediction score , $h_{\boldsymbol{\theta}} = a_j^{(L)}$, and the correct label y_j. It then backpropagates that error to compute the relative error of every unit according to

$$\delta_j^{(l)} = \begin{cases} a_j^{(l)} - y_j & l = L \\ \sigma'(z_j^{(l)}) \cdot \sum_{i=1}^{u^{(l+1)}} \theta_{i,j}^{(l)} \cdot \delta_i^{(l+1)} & \forall l \in [2, L-1] \end{cases} \tag{9}$$

Backpropagation of the errors using (9) requires the intermediate results $a_j^{(l)}$ and $z_j^{(l)}$ of the forward propagated feature vector. Therefore, a feature vector from the online learning batch is first forward propagated by the FFPU to obtain these intermediate results and store them in the shared cache. The FFPU then forward propagates the next feature vector from the online learning batch, while the BPPU begins to backpropagate the first feature vector using the results stored in the cache. The FFPU and BPPU thus always work in parallel on two subsequent samples when learning. Once the BPPU computes the errors $\delta^{(l)}$ for all required units in the neural network, the gradient is computed through

$$\Delta_{i,j}^{(l)} = \Delta_{i,j}^{(l)} + \delta_i^{(l+1)} \cdot a_j^{(l)}, \quad \forall i, j, l \tag{10}$$

These gradients represent an estimate of the cross-entropy cost function’s partial derivatives required for the gradient descent algorithm. The partial derivatives can therefore be obtained by

$$\frac{\partial}{\partial \theta_{i,j}^{(l)}} J(\theta_{i,j}) := \frac{1}{m} \Delta_{i,j}^{(l)} \;\Big|\; m = 32 \tag{11}$$

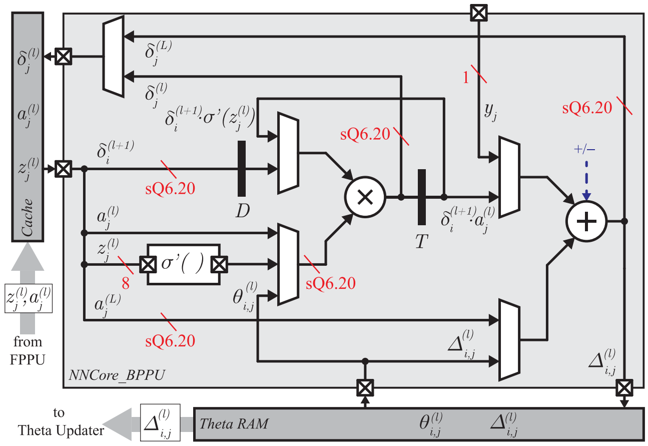


Fig. 7. A time-shared implementation of the BPPU to compute the gradients necessary for one subsequent gradient descent iteration.

The BPPU concludes by writing these derivatives into Theta RAM. The Theta Updater then reads these values and executes one iteration of the stochastic gradient descent in (8) to conclude learning for the currently assembled online learning batch.

The proposed design of the BPPU computing (9) and (10) is shown in Fig. 7. As for the FFPU, the BPPU time-shares one adder and multiplier to reduce area utilization at the expense of increased computation time. The execution of one gradient descent iteration, however, has to conclude before a new feature vector is supplied to the core which creates a new online learning batch. The time intensive computation of the non-linear derivative of the sigmoid function, $\sigma'(.)$, is thus implemented with a LUT to reduce its computation time to a single clock cycle. The LUT stores 256 entries with 22-bit fixed point precision (sQ1.20), which was sufficient to ensure accurate learning. The register D stores the error $\delta_i^{(l+1)}$, while a register T stores the temporary results $\delta_i^{(l+1)} \cdot \sigma'(z_j^{(l)})$ when computing (9), or $\delta_i^{(l+1)} \cdot a_j^{(l)}$ when computing (10). Note that the summation term in (9) simplifies to a single term if $u^{(l+1)} = 1$. This is the case in the neural network discussed here when computing the errors for the hidden layer, since $u^{(L)} = 1$.

V. Wearable Freezing of Gait Detection System

This section illustrates the proposed FoG detection system and shows that it has been successfully integrated into a single sensor node with real-time learning capability as a result of the efforts discussed in the previous sections.

A. Overview of the Freezing of Gait Detection System

The FoG detection system in its entirety is shown in Fig. 9. It integrates all fundamental elements of FoG detection systems into a single, ankle-worn sensor node, namely data acquisition, feature extraction, FoG classification and biofeedback generation. Data is acquired through an IMU (LSM9DS1, *STMicroelectronics*), while an FPGA implements the feature extraction and neural network architecture for classification and real-time learning (Artix-7 35T, *Xilinx*). A 32 Mb flash memory stores the FPGA configuration bitstream and ensures that the FPGA is properly configured upon power up. Biofeedback cues can be delivered in two different ways. Sematosensory cues are generated by a vibration motor that is attached to the inside of the strap and connects to the sensor node with wires. Auditory cues on the other hand are delivered through BLE capable earphones. Four DIP switches define the system operation modes, which determine the cueing type and enable online learning. The system is enclosed in a 4.5 × 3.5 × 1.5 cm^3 housing case and weighs only 32 g, which is an order of magnitude improvement compared to other reported systems without dedicated hardware. Power is supplied to the system using a chargeable Li-ion battery with 800 mAh capacity.

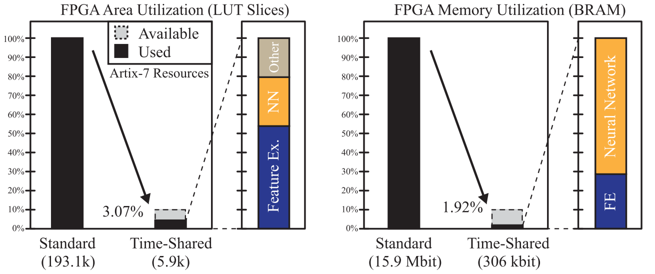
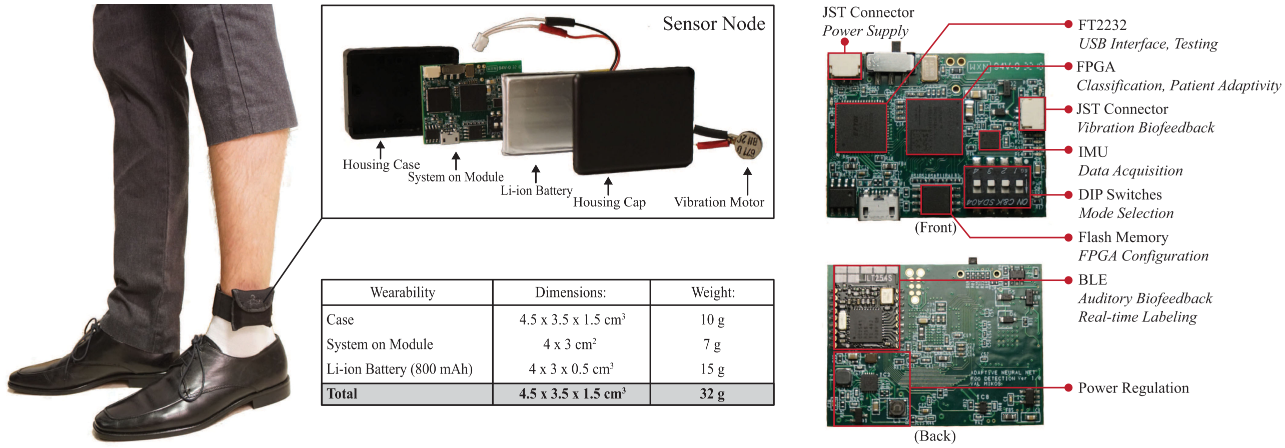


Fig. 8. Area and memory utilization of the proposed design and its ability to fit onto small form factor FPGAs such as the Artix-7 35T (10 × 10 mm^2).

B. Meeting the FPGA Resource Limitations

In our attempt to design a neural network based FoG detection system, a substantial focus has been put on limiting the design's overall resource requirements. There are three main design efforts that have contributed to that goal. First, the computational load on the FPGA has been limited by reducing the number of features that are extracted. Second, a substantial part of the neural network classification and learning computations employ time-shared circuitry. And third, the learning algorithm operates sequentially with batches consisting of 32 feature vectors that are discarded as new feature vectors arrive, circumventing the storage of a large dataset.

The proposed design's area and memory utilization are illustrated in Fig. 8. It indicates that the largest area consumption comes from the extraction of features (54% of total), mostly due to the required DFT of the FI in (1). Had the number of features not been reduced, it would have certainly been more challenging to fit the overall design onto the available resources of the Artix-7 FPGA. The neural network on the other hand requires comparatively little area as the majority of the functionality is computed with time-shared resources. The figure indicates what repercussions would result if no time-sharing had been employed. If the standard, vectorized approach to neural network computation were implemented, the required area would exceed the limitations of the Artix-7. A Kintex-7 480T FPGA would be required for the standard approach, which has a form factor of 31 × 31 cm^2 and is thus substantially larger than the Artix-7



Wearability	Dimensions:	Weight:
Case	4.5 x 3.5 x 1.5 cm^3	10 g
System on Module	4 x 3 cm^2	7 g
Li-ion Battery (800 mAh)	4 x 3 x 0.5 cm^3	15 g
Total	**4.5 x 3.5 x 1.5 cm^3**	**32 g**

Fig. 9. Overview of the proposed FoG detection system. A single, ankle-worn system on module integrates all fundamental elements of FoG detection systems and biofeedback cueing into a single sensor node.

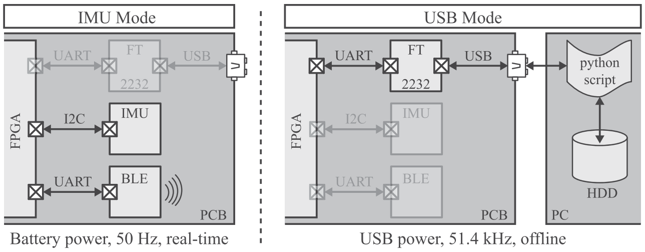


Fig. 10. Learning modes on the FoG detection system sensor node.

with 10 × 10 cm^2. Giving up computational speed by time-sharing circuitry has thus enabled the employment of FPGAs with smaller form factors, reducing the required FPGA LUT slices to 3.07%. Similarly, if the standard approach to learning were implemented, no FPGA would have sufficient on-chip memory to store the whole dataset. The proposed design of relying on a small online learning batch which discards training data with a FIFO replacement method for new data significantly reduces memory utilization. Compared to the standard, vectorized implementation in software, memory utilization for the time-shared design is reduced to 1.92%. Hence, online learning on small form factor FPGAs is made possible.

C. Learning on the Sensor Node

The sensor node allows for two different learning modes, as illustrated in Fig. 10. The first mode, USB mode, enables fast learning, while the second mode, IMU mode, enables power efficient, real-time learning for patient adaptivity.

1) USB Mode: This mode requires the sensor node to connect via USB cable to a desktop PC that stores a FoG dataset. A python script then transmits that data to the FPGA using the USB to UART bridge (FT2232). The FPGA accepts the incoming data and manages its online learning batch to conduct learning. Since a wired connection to a desktop PC is required, learning in this mode cannot be conducted during a patient's system usage. However, power consumption thus becomes inconsequential, as power is supplied by the USB cable rather than the battery. As a direct result of that, learning can be conducted at a high data rate without having to worry about battery lifetime. The maximum data rate is limited by the time it takes the FPGA to compute one learning cycle and accept new incoming data samples. The proposed design requires $19.4 \mu s$ to do so, hence a data rate 51.4 kHz can be achieved.

2) IMU Mode: In this mode, the IMU data collected when the system is worn by a patient is directly used for online learning. The sampling frequency of IMU is set to 50 Hz to limit the dynamic power consumption. Learning in both modes can be supervised and unsupervised. For supervised learning in IMU mode, the labels have to be transmitted in real-time, possibly through a BLE enabled push button.

VI. System Performance Evaluation

This section evaluates and discusses the proposed FoG system's viability in terms of learning performance, classification accuracy, power consumption and comfort for wearability.

A. FoG Dataset and Methodology

We base our evaluation on a dataset consisting of 63 patients that wore IMUs on their ankles ($f_s = 50$ Hz) while performing 7-meter timed up-and-go exercises and random walks. A subset of 25 patients from this dataset has been previously used to select features in Section III. The patient data has been labeled by two independent raters qualified to identify FoG by analyzing video recordings of the exercises. A third rater acted as an arbiter in case of disagreements between the first two raters. The dataset amounts to 3 hours and 50 minutes of gait data with 485 observed FoG episodes. The average duration of a FoG episode was 10.2 s. This makes the acquired FoG dataset the largest to date (see Table I). The study was approved by the Singhealth Centralized Institutional Review Board (CIRB 2011/255/A).

The dataset is split into a training and test set to obtain a reliable representation of the system's performance on unseen patients. The 25 patients used previously to select features form

TABLE III
TRAINING ACCURACY AND TIME OF THE NEURAL NETWORK WHEN EXECUTED ON A CONVENTIONAL DESKTOP PC (INTEL I5-4690 @3.5 GHZ) AND THE PROPOSED SENSOR NODE

	Sens	Spec	Training Time		
			Speedup	Actual	Mode
Software (Desktop PC)	97.3	89.9	1×	5min 54s	-
Hardware (Sensor Node)	97.3	89.8	22.7×	16s	USB
			0.02×	4h 29min	IMU

TABLE IV
CLASSIFICATION PERFORMANCE ON THE SENSOR NODE

	Database Model	Patient Adaptivity	
		Unsupervised	Supervised
Sens [%]	86.2	87.4	95.6
Spec [%]	87.7	86.9	90.2
Average [%]	**87.0**	**87.2**	**92.9**

the training set, while the remaining 38 are grouped into a test set. Learning performance is evaluated by the time it takes for the neural network to learn a "database model" using the training set. Classification performance on the other hand observes the ability of that database model to classify patients from the test set. To allow for comparisons within the FoG literature, the classification performance is assessed by sensitivity and specificity. Finally, patient adaptivity is evaluated by feeding the database model a stream of data from a particular test patient and let the neural network conduct learning as it receives that data. The classification performance when using patient adaptivity is then assessed by evaluating the resulting adapted model on unseen data of the same patient. This is done for all patients in the test set and the average is taken to summarize the efficacy of patient adaptivity.

B. Learning Performance

Table III summarizes the resulting training accuracy and training time when learning a database model using either the proposed sensor node or a desktop PC. The first thing to note is that learning on the sensor node leads to virtually the same database model with only a slight difference in training accuracy. The sensor node can learn such a model about 22.7 times faster than a desktop PC when using the USB learning mode. This is despite the fact that the proposed design time-shares most hardware resources and is only running at a frequency of 12 MHz, whereas the desktop PC is clocked at 3.5 GHz.

On the other hand, learning in IMU mode will take 45.5 times longer than a desktop PC. This is due to the data rate of the IMU which has been set low in order to limit dynamic power consumption. Hence, the sensor node is meant to learn a patient-independent database model in USB mode first and only learn in IMU mode for adapting that model to a patient in real-time. Such patient-specific adaptations are usually gradual enough to not require many learning samples, which justifies the use of lower data rates in IMU mode.

C. Classification Performance

Table IV demonstrates the attained classification performance on the test set of 38 patients. The patient-independent database model achieves 87.0% in average of sensitivity and specificity. Other ML based FoG detection systems required powerful smartphones to reach comparable performance values (78.6% [13], 88.8% [15], Table I). Once the sensor node's ability to adapt to patients is accounted for, the classification performance increases up to 92.9%. Such classification performance is comparable to those of recently reported ML based FoG classifiers that are purely based on software implementations (90.7% [25] and 96.1%[26]). These comparisons clearly demonstrate the effectiveness of the proposed patient-adaptive sensor node.

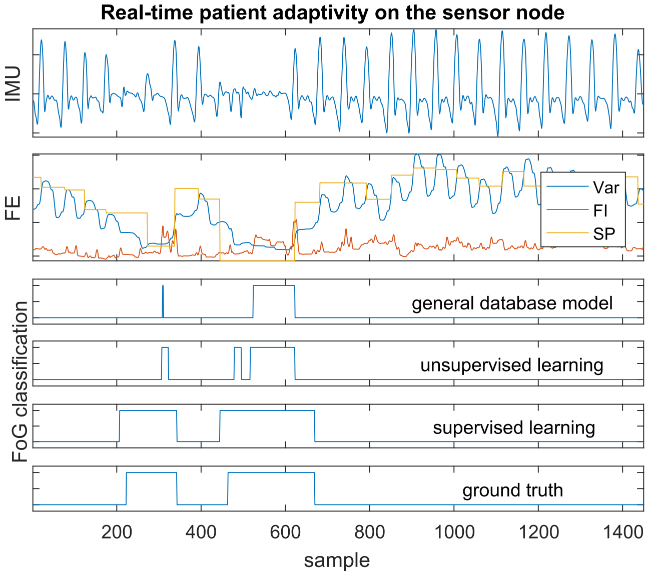


Fig. 11. Patient adaptivity provided through real-time learning capability on the sensor node.

The sensor node's learning ability is illustrated in Fig. 11. It shows how the database model is improved by adapting it to the patient currently wearing the system. As was to be expected, a supervised learning approach yields superior classification performance compared to the unsupervised counterpart. In fact, the unsupervised method barely shows any noticeable improvement on average when compared to the database model. Nevertheless, unsupervised learning has the potential to significantly improve the classification performance of patients which are not classified well with the database model and which have no means of labeling their gait. For example, patient 40 attained only 64.1% sensitivity and 97.6% specificity using the database model. With the use of unsupervised learning, the sensitivity improved to 85.3% with only a slight deterioration of specificity to 94.7%.

D. Wearability and Power Consumption

The proposed FoG detection system is the first to provide dedicated hardware integrated into a single sensor node. With an overall weight of 32 g, this is to date perhaps the least obstructive system yet. The 25 g reported in [17] is achieved using a battery with a 90 mAh capacity that allows for only 2 hours of operation. If a battery with similar capacity as ours was used, the weight

TABLE V
POWER CONSUMPTION OF THE SENSOR NODE

Operating Mode	BLE	Vibration	Power	Lifetime
Classification & Learning	✗	✗	80 mW	10 hours
Auditory Biofeedback	✓	✗	86 mW	9.3 hours
Vibration Biofeedback	✗	✓	96 mW	8.3 hours
Auditory & Vibration Biofeedback	✓	✓	102 mW	7.8 hours

would increase to 36 g. The power consumption for the different operating modes of the sensor node are shown in Table V. It shows that the node can operate continuously for 10 hours and 9.3 hours if auditory biofeedback is provided. The excitation of the vibration motor is more power intensive than sending BLE data, which explains the slightly lower lifetime of 8.3 hours when opting for sematosensory biofeedback. Any of these operating modes has a lifetime that exceeds previously published FoG detection systems, a possible exception being the smartphone based systems (see Table I). In any case, these figures impressively demonstrate the viability of the proposed design.

VII. CONCLUSION

This paper demonstrated the various design steps conducted to integrate a FoG detection system into a single sensor node. In order to achieve such an integration, certain design efforts and compromises were adopted. First, the computational load on the hardware was reduced by relying on features that incur minimum computational cost. It was shown that, on average, three features are most likely sufficient to build a ML based FoG classifier with comparable classification accuracy as one built with a larger, optimal feature set. Second, dedicated hardware for a neural network was designed such that it time-shared resources to reduce area utilization. The advantage of time-sharing was demonstrated by the ability to map the proposed design onto FPGAs with smaller form factors. Third, online learning with small batches of 32 samples was proposed to enable learning on the sensor node, where samples are discarded with a FIFO replacement method as new samples arrive. This significantly reduces memory requirement and allows the design to fit onto the chosen FPGA. With these design efforts, the proposed FoG detection system was successfully integrated into a sensor node that allows for patient adaptivity in real-time. The classification accuracy is comparable to recent software implemented ML based FoG classifiers, and exceeds other FoG detection systems when evaluated on an unseen test set. Specifically, the system achieves 92.9% in average of sensitivity and specificity when exploiting its patient adaptive learning capability. Due to its small form factor, the design is one of the least obstructive FoG detection systems when worn and operates for more than 9 hours while providing auditory biofeedback cues.

APPENDIX A
ESTIMATED FEATURE EXTRACTION COST

IMU sampling frequency is $f_s = 50$ Hz. Window lengths (N) are taken from [31].

A. Frequency Domain Features

Features in Table II in this category are freeze index (FI), locomotor band (LB), mean frequency ($\bar{f}$) and dominant frequency (DF). They all rely on the computation of a discrete Fourier transform for the time-series data x_n

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{2\pi i}{N}kn}, \tag{12}$$

which requires $2Nk$ real additions, $2Nk$ real multiplications and has 1 non-linear function.

1) Freeze Index: $N = 220$ for $|a|, \omega_z, a_x$. The two bands thus use bins $k = [3, 35]$, or 33 bins. $k - 1 = 32$ to add the bins together. Total cost is $2Nk + (k - 1)$ additions, $2Nk$ multiplications, 1 division, 1 non-linear function.

2) Locomotor Band: $N = 100$ for ω_z, a_x. Band thus uses bins $k = [1, 5]$, or 5 bins. $k - 1 = 4$ to add the bins together. Total cost is $2Nk + (k - 1)$ additions, $2Nk$ multiplications, 1 non-linear function.

3) Mean Frequency: $N = 120$ for ω_z. Band thus uses bins $k = [1, 19]$, or 19 bins. $k - 1 = 18$ to add the bins together, 1 division for mean. Total cost is $2Nk + (k - 1)$ additions, $2Nk$ multiplications, 1 division, 1 non-linear function.

4) Dominant Frequency: $N = 60$ for a_y. Band thus uses bins $k = [1, 9]$, or 9 bins. Needs 6 comparisons that require 2 additions/subtraction each [36, p. 331]. Total cost is $2Nk + 12$ additions, $2Nk$ multiplications, 1 non-linear function.

B. Arithmetic IMU Data Features

Features in Table II in this category are standard deviation (σ), root mean square ($\sqrt{.^2}$), maxima ($\lceil . \rceil$), range (R) and stride peak (SP).

1) Standard Deviation: $N = 80$ for ω_z, use $N = 64$ to eliminate divisions via bit shift operation. Total cost is $3N$ additions, N multiplications, 1 non-linear function (square root).

2) Root Mean Square: $N = 80$ for ω_z, use $N = 64$ to eliminate division via bit shift operation. Total cost is N additions, N multiplications, 1 non-linear function (square root).

3) Maxima: $N = 95$ for ω_z. Need 94 comparisons that require 2 additions/subtraction each [36, p. 331]. Total cost is $2(N - 1)$ additions.

4) Range: $N = 55$ for ω_y. Same as maxima, but once for maxima and minima. Hence, total cost is $4(N - 1)$ additions.

5) Stride Peak: According to [31], a 4th order Butterworth filter is needed

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}}. \tag{13}$$

When implemented in the canonical form results in 8 additions and 9 multiplications with no divisions. Need 2 comparison that require 2 additions/subtraction each [36, p. 331]. The total cost is 12 additions, 9 multiplications.

C. Spatio-Temporal Features

Features in Table II in this category are stride time (ST) and stride length (SL).

1) Stride Time: Requires to find peaks, as for stride peak. Computation of time duration is neglected here, hence total cost is 12 additions, 9 multiplications.

2) Stride Length: Extraction is based on Madgwick *et al.*'s algorithm [37] using quaternions. The reader may contact the authors for a run down of the computational expense. The total cost is 5050 multiplications, 3551 additions and 52 divisions.

References

[1] T. Pringsheim, N. Jette, A. Frolkis, and T. D. Steeves, "The prevalence of Parkinson's disease: A systematic review and meta-analysis," *Movement Disorders*, vol. 29, no. 13, pp. 1583–1590, 2014.

[2] Y. J. Zhao *et al.*, "Economic burden of Parkinson's disease in Singapore," *Eur. J. Neurology*, vol. 18, no. 3, pp. 519–526, 2011.

[3] Y. J. Zhao *et al.*, "Estimating the lifetime economic burden of Parkinson's disease in Singapore," *Eur. J. Neurology*, vol. 20, no. 2, pp. 368–374, 2013.

[4] M. Macht *et al.*, "Predictors of freezing in Parkinson's disease: A survey of 6,620 Patients," *Movement Disorders*, vol. 22, no. 7, pp. 953–956, 2007.

[5] S. T. Moore, H. G. MacDougall, J.-M. Gracies, H. S. Cohen, and W. G. Ondo, "Long-term monitoring of gait in Parkinson's disease," *Gait Posture*, vol. 26, no. 2, pp. 200–207, 2007.

[6] B. R. Bloem, J. M. Hausdorff, J. E. Visser, and N. Giladi, "Falls and freezing of gait in Parkinson's disease: A review of two interconnected, episodic phenomena," *Movement Disorders*, vol. 19, no. 8, pp. 871–884, 2004.

[7] M. D. Latt, S. R. Lord, J. G. Morris, and V. S. Fung, "Clinical and physiological assessments for elucidating falls risk in Parkinson's disease," *Movement Disorders*, vol. 24, no. 9, pp. 1280–1289, 2009.

[8] WHO and Ageing and Life Course Unit, *WHO Global Report on Falls Prevention in Older Age*. Geneva, Switzerland: World Health Organization, 2008.

[9] I. Farag *et al.*, "Economic evaluation of a falls prevention exercise program among people with Parkinson's disease," *Movement Disorders*, vol. 31, no. 1, pp. 53–61, 2016.

[10] M. Bächlin, J. M. Hausdorff, D. Roggen, N. Giladi, M. Plotnik, and G. Tröster, "Online detection of freezing of gait in Parkinson's disease patients: A performance characterization," in *Proc. 4th Int. Conf. Body Area Netw.*, 2009, Art. no. 11.

[11] M. Bächlin, M. Plotnik, D. Roggen, N. Giladi, J. M. Hausdorff, and G. Tröster, "A wearable system to assist walking of Parkinson's disease patients," *Methods Inf. Med.*, vol. 49, pp. 88–95, 2010.

[12] M. Bächlin *et al.*, "Wearable assistant for Parkinson's disease patients with the freezing of gait symptom," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 436–446, Mar. 2010.

[13] S. Mazilu *et al.*, "Online detection of freezing of gait with smartphones and machine learning techniques," in *Proc. 6th Int. Conf. Pervasive Comput. Technologies Healthcare*, May 2012, pp. 123–130.

[14] L. Pepa, F. Verdini, M. Capecci, and M. Ceravolo, "Smartphone based freezing of gait detection for Parkinsonian patients," in *Proc. IEEE Int. Conf. Consumer Electron.*, 2015, pp. 212–215.

[15] H. Kim *et al.*, "Unconstrained detection of freezing of gait in Parkinson's disease patients using smartphone," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2015, pp. 3751–3754.

[16] D. Comotti, M. Galizzi, and A. Vitali, "Nememsi: One step forward in wireless attitude and heading reference systems," in *Proc. Int. Symp. Inertial Sensors Syst.*, 2014, pp. 1–4.

[17] P. Lorenzi, R. Rao, G. Romano, A. Kita, and F. Irrera, "Mobile devices for the real-time detection of specific human motion disorders," *IEEE Sensors J.*, vol. 16, no. 23, pp. 8220–8227, Dec. 2016.

[18] D. Ahn *et al.*, "Smart gait-aid glasses for Parkinson's disease patients," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 10, pp. 2394–2402, Oct. 2017.

[19] C. Punin, B. Barzallo, M. Huerta, A. Bermeo, M. Bravo, and C. Llumiguano, "Wireless devices to restart walking during an episode of FOG on patients with Parkinson's disease," in *Proc. IEEE Ecuador Technical Chapters Meet.*, 2017, pp. 1–6.

[20] A. Kita, P. Lorenzi, R. Rao, and F. Irrera, "Reliable and robust detection of freezing of gait episodes with wearable electronic devices," *IEEE Sensors J.*, vol. 17, no. 6, pp. 1899–1908, Mar. 2017.

[21] A. Nieuwboer *et al.*, "Cueing training in the home improves gait-related mobility in Parkinson's disease: The rescue trial," *J. Neurology, Neurosurgery Psychiatry*, vol. 78, no. 2, pp. 134–140, 2007.

[22] A. Delval *et al.*, "Auditory cueing of gait initiation in Parkinson's disease patients with freezing of gait," *Clin. Neurophysiology*, vol. 125, no. 8, pp. 1675–1681, 2014.

[23] P. J. McCandless, B. J. Evans, J. Janssen, J. Selfe, A. Churchill, and J. Richards, "Effect of three cueing devices for people with Parkinson's disease with gait initiation difficulties," *Gait Posture*, vol. 44, pp. 7–11, 2016.

[24] O. A. Cando, K. R. Hidalgo, and B. C. Palacios, "A low-cost vibratory stimulus system to mitigate freezing of gait in Parkinson's disease," in *Proc. IEEE ANDESCON*, 2016, pp. 1–4.

[25] J. Camps *et al.*, "Deep learning for freezing of gait detection in Parkinson's disease patients in their homes using a waist-worn inertial measurement unit," *Knowledge-Based Syst.*, vol. 139, pp. 119–131, 2018.

[26] C. Ahlrichs *et al.*, "Detecting freezing of gait with a tri-axial accelerometer in Parkinson's disease patients," *Med. Biol. Eng. Comput.*, vol. 54, no. 1, pp. 223–233, 2016.

[27] J. P. Dominguez-Morales, A. F. Jimenez-Fernandez, M. J. Dominguez-Morales, and G. Jimenez-Moreno, "Deep neural networks for the recognition and classification of heart murmurs using neuromorphic auditory sensors," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 24–34, Feb. 2018.

[28] X. Tang, Q. Hu, and W. Tang, "A real-time QRS detection system with PR/RT interval and ST segment measurements for wearable ECG sensors using parallel delta modulators," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 4, pp. 751–761, Aug. 2018.

[29] L. Feng, Z. Li, and Y. Wang, "VLSI design of SVM-based seizure detection system with on-chip learning capability," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 171–181, Feb. 2018.

[30] A. Jafari, N. Buswell, M. Ghovanloo, and T. Mohsenin, "A low-power wearable stand-alone tongue drive system for people with severe disabilities," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 58–67, Feb. 2018.

[31] V. Mikos *et al.*, "Optimal window lengths, features and subsets thereof for freezing of gait classification," in *Proc. Int. Conf. Intell. Informat. Biomed. Sci.*, 2017, pp. 1–8.

[32] S. T. Moore, H. G. MacDougall, and W. G. Ondo, "Ambulatory monitoring of freezing of gait in Parkinson's disease," *J. Neurosci. Methods*, vol. 167, no. 2, pp. 340–348, 2008.

[33] V. Mikos *et al.*, "A neural network accelerator with integrated feature extraction processor for a freezing of gait detection system," in *Proc. Asian Solid-State Circuits Conf.*, 2018, pp. 59–62.

[34] B. Widrow, P. Baudrenghien, M. Vetterli, and P. Titchener, "Fundamental relations between the LMS algorithm and the DFT," *IEEE Trans. Circuits Syst.*, vol. 34, no. 7, pp. 814–820, Jul. 1987.

[35] V. Mikos *et al.*, "Real-time patient adaptivity for freezing of gait classification through semi-supervised neural networks," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl.*, Dec. 2017, pp. 871–876.

[36] N. H. Weste, D. Harris, and A. Banerjee, *CMOS VLSI Design*, 3rd ed. London, U.K.: Pearson, 2010.

[37] S. Madgwick, A. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *Proc. IEEE Int. Conf. Rehabil. Robot.*, Jun. 2011, pp. 1–7.

Val Mikos (S'16) received the B.Sc. and M.Sc. degrees in electrical engineering and information technology from the Swiss Federal Institute of Technology, Zürich, Switzerland, in 2012 and 2014, respectively. Since 2015, he has been working toward the Ph.D. degree with the National University of Singapore, Singapore.

His research interests include VLSI design for machine learning and system design for battery-powered, miniaturized hardware.

Chun-Huat Heng (S'96–M'04–SM'13) received the B.Eng. and M.Eng. degrees from the National University of Singapore (NUS), Singapore, in 1996 and 1999, respectively, and the Ph.D. degree from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2003.

From 2001 to 2004, he was with Wireless Interface Technologies, San Diego, CA, USA, which was later acquired by Chrontel. Since 2004, he has been with the NUS, Singapore, where he is currently an Associate Professor. He has been involved in CMOS integrated circuits involving synthesizer, delay-locked loop, and transceiver circuits.

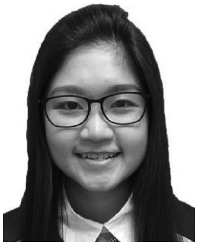
Dr. Heng is currently a Technical Program Committee Member for the International Solid-State Circuits Conference and the Asian Solid-State Circuits Conference. He was a recipient of the NUS Annual Teaching Excellence Award in 2008, 2011, and 2013, the Faculty Innovative Teaching Award in 2009, and the ATEA Honor Roll in 2014. He was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II.

Arthur Tay received the B.Eng. (first class hons.) and Ph.D. degrees in electrical engineering from the National University of Singapore, Singapore, in 1995 and 1998, respectively.

He was a visiting scholar with the Information System Laboratory, Stanford University, Stanford, CA, USA, from 1998 to 2000. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore. His research interests include applications of mathematical system science tools in healthcare, semiconductor manufacturing, and process control.

Shih-Cheng Yen received the B.S.E. and M.S.E. degrees in 1993, and the Ph.D. degree in 1998, all from the Department of Bioengineering, University of Pennsylvania, Philadelphia, PA, USA.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, National University of Singapore. His research interests include neural coding, neuroprosthetic devices, and telehealth solutions.



Nicole Shuang Yu Chia received the B.A. degree (*magna cum laude*) in psychology and communication from the State University of New York, Buffalo, NY, USA, in 2015.

Since 2016, she has been working on Parkinson's disease and movement disorders research with the National Neuroscience Institute, Singapore. Her research interests include telerehabilitation, gait disorders, and neuropsychology.

Karen Mui Ling Koh received the Bachelor's degree in health sciences (Physiotherapy) from The University of Sydney, Sydney, Australia, in 1999. Her research interest includes rehabilitation.

Dawn May Leng Tan received the Doctor of Clinical Physiotherapy degree from the University of Melbourne, Parkville, VIC, Australia, in 2011.

Dr. Dawn is currently a Senior Principal Physiotherapist with the Singapore General Hospital and works with patients with neurological and vestibular disorders in the outpatient setting. She is also an Assistant Professor with the Singapore Institute of Technology, and teaches in the undergraduate physiotherapy degree program. Her areas of interests include stroke, Parkinson's disease and vestibular rehabilitation, and promoting evidence-based practice (EBP) such as conducting systematic reviews and running EBP workshops. She is a recipient of The Efficiency Medal from Singapore's Ministry of Health.

Wing Lok Au received the Bachelor's of Medicine, Bachelor's of Surgery (MBBS) degree from the National University of Singapore, Singapore, in 1994.

He is currently the Deputy Medical Director (Clinical) of National Neuroscience Institute of Singapore, Singapore, and the Academic Deputy Chair of the Neuroscience Academic Clinical Programme, SingHealth-DukeNUS, Singapore. He received his neurology training at the National Neuroscience Institute of Singapore, and underwent two years of Fellowship training in movement disorders at the Pacific Parkinson's Research Centre, Vancouver, Canada, from 2004 to 2005. In 2012, he went for further training in neurophysiology in movement disorders at the Toronto Western Hospital, Toronto, Canada. His research interests include functional neuroimaging, neurophysiology, and biosensor devices in movement disorders. He was a recipient of the Distinguished Team Award, SingHealth Excellence Award in 2015, and Outstanding Clinician Award, GCEO Excellence Award, SingHealth in 2018. He has been a Fellow of the Academy of Medicine (Singapore) since 2004, and a Fellow of the Royal College of Physicians (Edinburgh) since 2007.