sensors

Article

Motion Artifact Suppression for Insulated EMG to Control Myoelectric Prostheses

Theresa Roland

Institute of Biomedical Mechatronics, Johannes Kepler University, 4040 Linz, Austria; theresa.roland@jku.at

Received: 15 January 2020; Accepted: 10 February 2020; Published: 14 February 2020

Abstract: Myoelectric prostheses help amputees to regain independence and a higher quality of life. These prostheses are controlled by electromyography, which measures an electrical signal at the skin surface during muscle contractions. In this contribution, the electromyography is measured with innovative flexible insulated sensors, which separate the skin and the sensor area by a dielectric layer. Electromyography sensors, and biosignal sensors in general, are striving for higher robustness against motion artifacts, which are a major obstacle in real-world environment. The motion artifact suppression algorithms presented in this article, prevent the activation of the prosthesis drive during artifacts, thereby achieving a substantial performance boost. These algorithms classify the signal into muscle contractions and artifacts. Therefore, new time domain features, such as Mean Crossing Rate are introduced and well-established time domain features (e.g., Zero-Crossing Rate, Slope Sign Change) are modified and implemented. Various artificial intelligence models, which require low calculation resources for an application in a wearable device, were investigated. These models are neural networks, recurrent neural networks, decision trees and logistic regressions. Although these models are designed for a low-power real-time embedded system, high accuracies in discriminating artifacts to contractions of up to 99.9% are achieved. The models were implemented and trained for fast response leading to a high performance in real-world environment. For highest accuracies, recurrent neural networks are suggested and for minimum runtime (0.99–1.15 µs), decision trees are preferred.

Keywords: motion artifacts; insulated/capacitive EMG; artificial intelligence; neural network; time domain features; myoelectric prosthesis

1. Introduction

Electromyography (EMG) sensors are applied, beyond others, for myoelectric prosthesis control, diagnostic purposes and exoskeletons. The state-of-the-art dry sensors, which require a conductive connection to the skin, are typically applied for myoelectric prosthesis control. This research group has already published the developed flexible insulated EMG sensors [1,2] to overcome the disadvantages associated with dry conductive EMG sensors. The insulated sensors avoid pressure marks and are independent of sweat and the skin condition in general.

Various research has presented the control of high-level dexterity prostheses [3–5]. Well-established EMG features have been implemented [6–8] and different signal processing algorithms have been developed [9–11]. Although these algorithms achieve high accuracies for distinguishing hand movements, they often lack robustness in real-world environment [12,13]. However, in terms of practical application, robustness is preferred over technologically complex and unreliable systems [14].

The main reason for the lacking robustness are the motion artifacts [15], which have various origins. They typically occur in low frequency range. However, these artifacts also appear in the frequency range of the contraction EMG, which limits the possibilities for filtering. To achieve high robustness, real-time motion artifact suppression algorithms were implemented on an ultra-low-power microcontroller (μC) in this contribution. These algorithms aimed at an activation of the prosthesis drive solely during an actual contraction. The aim of the algorithm was the distinction of the two classes: contraction and artifact.

An algorithm with a frequency domain feature has already been presented by Roland et al. [16]. Frequency domain features require a Fourier transform, which leads to a long time span between decisions. The presented algorithm in Roland et al. [16] has a weakness in detecting short contractions. At the beginning and the end of a contraction, the muscle tissue moves relatively to the skin surface, which causes a motion artifact. As short contractions mainly consist of the beginning and the end of a contraction, a motion artifact is overlaid to the contraction EMG signal. However, these short contractions are of high importance to enable co-contractions, which switch the prosthesis movement mode.

In this article, only time domain (TD) features were implemented and instead of the linear separator in Roland et al. [16] different artificial intelligence models, such as neural networks (NN), were investigated. The algorithms were designed to confidently detect the short contractions. Additionally, the algorithms aimed at a robust detection of strong and weak contractions to allow a proportional control of the myoelectric hand prosthesis.

These algorithms were developed with MATLAB® aiming at high accuracy but low calculation resources. New features were designed, and different artificial intelligence models were trained, and the resulting models were implemented on an ultra-low-power μC. The algorithms were designed for the insulated EMG sensor, however, they can also be applied to increase the robustness of conductive EMG sensors or even of other biosignal sensors.

2. Methods

Figure 1 shows the signal flow graph of the capacitive EMG signal. The data acquisition, filtering, rectification and smoothing were described in detail by Roland et al. [1,2]. The block diagram of the digital signal processing is plotted in Figure 2. Additional data pre-processing, the feature calculation, the decision algorithm (logistic regressions, decision trees, NN or recurrent neural networks (RNN)) and the data post-processing as well as the signal delay are described in this section.

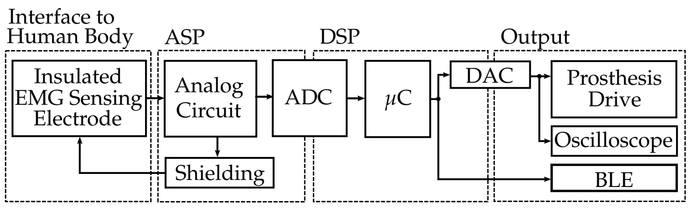

Figure 1. First, the EMG signal coupled from the human body is amplified and filtered by the analog circuit (ASP). In the μC, the signal is then digitally filtered and evaluated (DSP). The contraction EMG signal is provided at the prosthesis drive via the digital-to-analog converter (DAC). For experimental applications, the signal can be connected to the oscilloscope or Bluetooth (BLE) module (Adapted from Roland et al. [2]).

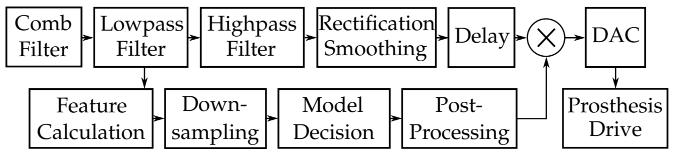

Figure 2. Signal flow of the digital signal processing. The decision of the model (0 for artifacts or 1 for contractions) is multiplied with the delayed processed EMG signal.

2.1. Data Acquisition and Pre-Processing

Insulated EMG data were acquired with the sensor developed by Roland et al. [1,2]. Figure 3 shows flexible coupling electrodes, which are an assembly of textiles or a flexprint. The EMG sensing system filtered the signal with a first-order analog bandpass (f_{CL}= 11 Hz, f_{CU} = 1064 Hz), a digital comb (50 Hz and harmonics) and a first-order digital lowpass (f_C = 531 Hz). In the feature calculation path, a second order digital high pass (f_C = 60 Hz) was implemented. The optimal choice of these above-mentioned parameters was investigated in Roland et al. [1,2]. The parameter values were selected according to these findings. The flexible sensor was placed at the musculus extensor digitorum at the human forearm of one subject. Four different types of data were measured. Data for strong and weak isometric contractions, random sequences of short contractions and measurements of many artifacts were acquired. Therefore, various different motion artifacts were generated, such as sensor lift-off, external mechanical shocks or vibrations.

The data were measured with a digital oscilloscope (HS3 of TiePie Engineering) at the μC's digital-to-analog converter. The 10 s measurements were sampled by the oscilloscope at 10 kHz to fulfill the Nyquist-Shannon sampling theorem [17]. A measured contraction and a motion artifact are presented in Figure 4. The measured EMG data were downsampled by averaging to 2 kHz. According to Roland et al. [2] downsampling to 2 kHz does not reduce the EMG signal quality. This sampling frequency of 2 kHz was applied for the feature calculation. Time windows with small signals, which would not lead to a prosthesis drive activation, were removed from the data set.

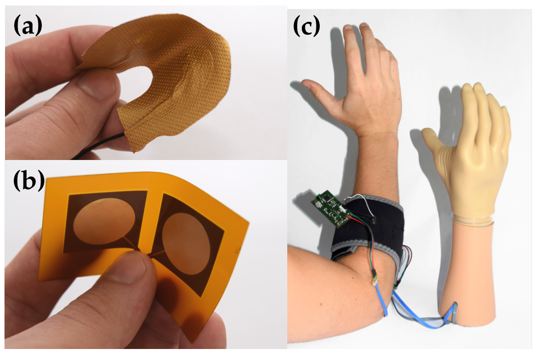

Figure 3. (**a**) Textile capacitive EMG sensing electrode. (**b**) Flexprint capacitive EMG sensing electrode. (**c**) Control of myoelectric hand prosthesis with capacitive EMG measurement setup (Adapted from Roland et al. [1]).

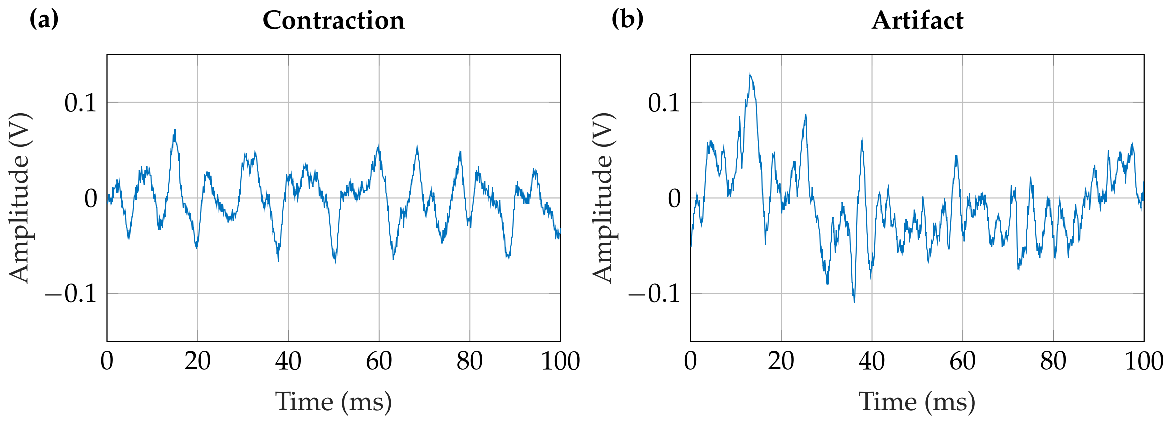

Figure 4. Raw signals measured with the capacitive EMG sensor at a sampling frequency of 10 kHz. (**a**) EMG signal of a strong contraction. (**b**) Exemplary artifact. Please note that artifacts are varying greatly, depending on its origin.

After these pre-processing steps, a total of 381 s of contractions and 489 s of artifacts were the remaining training, test and validation data. These data comprised a wide variety of different contractions and artifacts to achieve high robustness in real-world environment.

The contraction and artifact signals were mixed to 1 s long windows. The contraction and artifact signal windows were alternated, hence a fast reaction to the changing signal class was ensured. The features were calculated over these alternating time windows. Features which were slow at the transients led to errors at the transition from one class to the other. By iterating the classes in the data set, these slow features were consequentially removed by the feature selection. This is essential for real-world performance, where artifacts and contractions are alternating and a fast reaction of the decision is required. For the training and validation set of the RNN, the duration of the windows was varied for each snipped, thereby preventing the RNN to learn the periodic behavior of the training set.

The training (70%) and validation (15%) set amounted to 85% of the data. The validation data were selected out of the 85% of the data when training the models with cross validation, see Section 2.6. The remaining 15% were the test set, which was equal for all models. The training, validation and test set comprised an equal percentage of the strong, short, and weak contractions, and artifacts according to their share on the whole data set. A sample of the features highly correlates to the previous and next sample in time, the adjacent feature values are very similar. Drawing individual samples for the allocation to the training, validation and test data would lead to non-generalizable results. Therefore, continuous time sequences were selected in the allocation in order that the training, validation and test data are independent from each other.

2.2. Feature Calculation

In this contribution, new features and modifications of features for biosignal processing, and well-established features were included to the feature calculation. The TD features were hand-crafted as the models should achieve high accuracies with a shallow model architecture. The feature vector $\boldsymbol{feat}$ was calculated by filtering with a first-order lowpass, an exponentially decaying moving average filter (EMA) [18]:

$$feat_i = (feat_{i-1} + f(x_i))b\,, \tag{1}$$

where $f(x_i)$ denotes a function of the current comb and lowpass filtered value x_i. The function $f(x_i)$ is defined differently for each feature. The coefficient b was selected in the interval $[0, 1]$ (in floating-point format). This b was implemented as a fixed-point with a multiplication and a shift operation to avoid operations with floating-point variables. This implementation requires less calculation resources than a standard moving average filter [2].

The features were designed with the aim to detect differences of contraction EMG and artifacts. These TD features also attempt to detect frequency information, which is relevant for the distinction of the classes, e.g., by the zero-crossing rate. Some features were implemented several times with varied parameters, which enables the extraction of different signal characteristics.

The features were designed with MATLAB® and implemented on the μC in C. The MATLAB® and C implementation were in the same value range and truncation, overflow and saturation were considered. Hence, the values listed in the following section are valid for both implementations.

Some features were saturated at an upper bound (ub) and/or a lower bound (lb) to avoid a drift of the feature values, thus achieving a fast transition between the different classes. The selected parameters of the implemented features are listed in Tables 1–7.

2.2.1. Zero-Crossing Rate (ZCR)

The ZCR (Figure 5a) is a well-established feature for EMG signal analysis, which comprises frequency information. When the EMG signal crosses zero, the feature is increased by 100. This feature update parameter is set to 100 to prevent the feature variable from vanishing right away due to truncation caused by the shift operation in the lowpass filter in Equation (1). Moreover, the value was not selected higher to prevent overflows. The parameter was determined, so that no overflows occurred in the training data. These facts are also valid for the selection of this feature update parameter for the following described features. As also noise with low amplitude contributes to the feature, a hysteresis (*hyst*) was added to detect the crossings only, when the signal exceeds a certain amplitude. An upper bound *ub* was included in *ZCR2S* to avoid a drift of the feature value, thereby achieving fast transitions. The conditions for the *ZCR* were defined as

$$f(x_i) = \begin{cases} 100, & [s = 1]; \quad \text{if } x_i > hyst \land s = 0 \\ 100, & [s = 0]; \quad \text{if } x_i < -hyst \land s = 1 \\ 0, & [s = s]; \quad \text{else}\,, \end{cases} \tag{2}$$

where s denotes the sign of the previous zero-crossing.

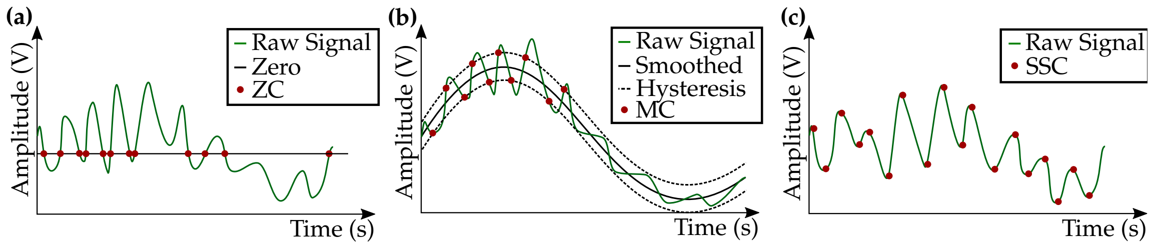

Figure 5. Exemplary sketches of features. Note that different variants of the sketched features were calculated. (**a**) Zero-Crossing (ZC): The crossings of the raw signal through the zero line are detected. (**b**) Mean crossing (MC): The crossings of the raw signal through the smoothed signal (+/− a hysteresis) are detected. (**c**) Slope Sign Change (SSC): The changes of the slope sign of the raw signal are detected.

Table 1. Parameters of *ZCR* implementations.

Feature	b	hyst	ub
*ZCR*1	0.9961	0	$2^{16}-1$
*ZCR*2	0.9961	242	$2^{16}-1$
*ZCR*2*S*	0.9961	242	1000

2.2.2. Mean Crossing Rate (MCR)

The contraction EMG is usually superimposed to a slight artifact at short contractions, due to the relative movement of the muscle tissue to the skin surface at the beginning and the end of a contraction. To detect this superimposed EMG signal, the crossing of the signal through the smoothed signal, as shown in Figure 5b, increases the feature by 100.

A hysteresis *hyst* was included to avoid noise contributing to the feature. A feature variant with a saturation at an upper bound (*ub*) and a lower bound (*lb*) was implemented. The conditions for the *MCR* were defined as

$$f(x_i) = \begin{cases} 100, & [s = 1]; \quad \text{if } x_D > (v + hyst) \land s = 0 \\ 100, & [s = 0]; \quad \text{if } x_D < (v - hyst) \land s = 1 \\ 0, & [s = s]; \quad \text{else}\,, \end{cases} \tag{3}$$

where s denotes the sign of the previous mean crossing and v is defined as

$$v = (smo_i)e\,. \tag{4}$$

The smoothed signal smo_i is calculated with an EMA with the coefficient c:

$$smo_i = (smo_{i-1} + x_i)c\,. \tag{5}$$

The input x_i is delayed and scaled by

$$x_D = (x_{i-delay})d\,. \tag{6}$$

Table 2. Parameters of *MCR* implementations.

Feature	b	c	d	e	delay	hyst	lb	ub
$MCR1$	0.9961	0.8750	6	1	8	1044	0	$2^{16}-1$
$MCR1S$	0.9961	0.8750	6	1	8	1044	2000	3600
$MCR2$	0.9922	0.9688	1	0.0313	8	0	0	$2^{16}-1$

2.2.3. Slope Sign Change (SSC)

When the EMG signal slope is changing (Figure 5c), the feature is increased by 100. To avoid noise contribution, the feature is calculated with the smoothed signal from Equation (5). Variants of the feature require the signal slope to maintain the direction for a minimum number of data points d_{min} and a maximum number of data points d_{max}. This way capturing the desired frequency information. Again, features with a saturation at a lower lb and an upper bound ub were implemented. The conditions were defined as

$$f(x_i) = \begin{cases} 100, & [s=1, \quad w=0, \quad p=0]; \\ & \text{if } v > u \wedge s = 0 \wedge w \geq d_{min} \wedge p < d_{max} \\ 0, & [s=1, \quad w=0, \quad p=0]; \\ & \text{elseif } v > u \wedge s = 0 \wedge w \geq d_{min} \\ 0, & [s=s, \quad w=w+1, \quad p=p]; \\ & \text{elseif } v > u \wedge s = 0 \\ 0, & [s=1, \quad w=0, \quad p=p+1]; \\ & \text{elseif } v > u \\ 100, & [s=0, \quad w=0, \quad p=0]; \\ & \text{elseif } v < u \wedge s = 1 \wedge w \geq d_{min} \wedge p < d_{max} \\ 0, & [s=0, \quad w=0, \quad p=0]; \\ & \text{elseif } v < u \wedge s = 1 \wedge w \geq d_{min} \\ 0, & [s=s, \quad w=w+1, \quad p=p]; \\ & \text{elseif } v < u \wedge s = 1 \\ 0, & [s=0, \quad w=0, \quad p=p+1]; \\ & \text{else}, \end{cases} \tag{7}$$

where s denotes the sign of the previous slope. The variables w and p are counters and u and v are the smoothed signal from Equation (5):

$$u = smo_{i-1}\,, \tag{8}$$

$$v = smo_i\,. \tag{9}$$

Table 3. Parameters of *SSC* implementations.

Feature	b	c	d_{min}	d_{max}	lb	ub
$SSC1$	0.9961	0.9922	0	$2^{16}-1$	0	$2^{16}-1$
$SSC1S$	0.9961	0.9922	0	$2^{16}-1$	2000	5000
$SSC2$	0.9961	0.5	0	$2^{16}-1$	0	$2^{16}-1$
$SSC2S$	0.9961	0.5	0	$2^{16}-1$	0	14000
$SSC3$	0.9961	0.9961	5	$2^{16}-1$	0	$2^{16}-1$
$SSC3S$	0.9961	0.9961	5	$2^{16}-1$	900	2000
$SSC4$	0.9961	0.9961	0	1	0	$2^{16}-1$
$SSC5$	0.9961	0.75	3	200	0	$2^{16}-1$
$SSC5S$	0.9961	0.75	3	200	1500	3000

2.2.4. Waveform Length (WFL)

Different variants of the feature WFL were implemented. The waveform length is calculated by

$$f(x_i) = |x_i - x_{i-1}|d\,. \tag{10}$$

Table 4. Parameters of WFL implementations.

Feature	b	d	ub
$WFL1$	0.9961	0.0078	$2^{16}-1$
$WFL1S$	0.9961	0.0078	1300
$WFL2S$	0.9922	0.0156	1894

2.2.5. Mean Absolute Value (MAV)

The function $f(x_i)$ for the MAV was defined as

$$f(x_i) = |x_i|\,. \tag{11}$$

The $MAV2$ is calculated by means of the $MAV1$ by

$$MAV2_i = (MAV2_{i-1} + |MAV1_i - MAV1_{i-delay}|)c\,. \tag{12}$$

The feature $MAV1S$ is saturated at the upper bound $ub1$ and the $MAV2S$ at $ub2$.

Table 5. Parameters of MAV implementations.

Feature	b	c	delay	ub1	ub2
$MAV1$	0.9961	-	0	$2^{16}-1$	$2^{16}-1$
$MAV1S$	0.9961	-	0	600	$2^{16}-1$
$MAV2$	0.9375	0.9961	8	$2^{16}-1$	$2^{16}-1$
$MAV2S$	0.9375	0.9961	8	4000	6000

2.2.6. Wilison Amplitude (WAM)

When the EMG signal exceeds a defined threshold *thresh*, the feature is increased by 100. The conditions for $WAM1$ were defined as

$$f(x_i) = \begin{cases} 100; & \text{if } |x_i| > thresh \\ 0; & \text{else}\,. \end{cases} \tag{13}$$

For $WAM2$, the smoothed signal smo_i is calculated with Equation (5) and the conditions are

$$f(x_i) = \begin{cases} 10; & \text{if } |(x_{i-delay}d) - smo_i| > thresh \\ 0; & \text{else}\,. \end{cases} \tag{14}$$

Table 6. Parameters of WAM implementations.

Feature	b	c	d	delay	thresh
$WAM1$	0.9922	-	-	0	44
$WAM2$	0.9961	0.9922	16	64	3636

2.2.7. Variance (VAR)

The VAR is the squared EMG signal, which was also implemented with a saturation at the upper bound ub. The VAR is calculated by

$$f(x_i) = {x_i}^2 \,. \tag{15}$$

Table 7. Parameters of VAR implementations.

Feature	b	ub
VAR	0.9961	$2^{16}-1$
$VARS$	0.9961	4000

2.3. Downsampling

For the training of the NN, decision tree, and logistic regression, the features were downsampled from 2 kHz to 40 Hz to accelerate model training. In the implementation the features were not downsampled to achieve short time periods between the decisions. For the RNN, the features were downsampled from 2 kHz to 250 Hz, which led to a higher calculation effort in the learning process. However, an equal sampling frequency in the training and implementation is indispensable for high accuracies of RNNs.

2.4. Correlation of Features

To reduce the number of features before training, the Pearson correlation coefficients [19] of the feature vectors were calculated by pairwise comparison. For an absolute value of the correlation coefficient greater than 0.9, one feature of the pair was eliminated.

2.5. Normalization

For the logistic regression, NN and RNN (but not for the decision tree) the min-max normalization was calculated for each entry of the feature vector $\boldsymbol{feat}$:

$$feat'_i = \frac{(y_{max} - y_{min})(feat_i - feat_{min})}{(feat_{max} - feat_{min})} + y_{min} \tag{16}$$

with $y_{max} = 1$ and $y_{min} = -1$ in floating-point format. The minimum and maximum ($feat_{min}, feat_{max}$) were determined for the feature vector calculated with the training data.

2.6. Training of Models

All models were trained with a 5-fold cross validation. The validation data were randomly drawn time sequences of the strong, short and weak contractions and the artifacts. The decisions within 100 ms after a transition of the target value were not included in the error calculation of the validation data. Wrong decisions, starting from 100 ms after a transition of the target value, were recorded as errors, thus achieving fast models. For the test data, this tolerance was set to 150 ms (cf. signal delay, Section 2.8.3). The models were trained with functions provided by the MATLAB® toolbox "Statistics and Machine Learning".

2.6.1. Logistic Regression

The features were selected with the lasso algorithm [20] assuming a binomial distribution. 30 different regularization penalties (λ) were applied. The range of the λ was defined such that the resulting feature vectors were ranging from a maximum number of regression coefficients to maximum sparsity. A vector with the desired number of features was selected and the non-zero entries were the selected features.

With these features the logistic regression model was trained by the Ridge Regression algorithm [21]. Finally, the model with the highest accuracy was selected.

2.6.2. Decision Tree

Binary decision trees were trained with different numbers of maximum splits, which avoids overfitting. The split and the respective feature was selected by calculating the Gini Diversity Index [22].

2.6.3. Neural Network (NN)

The features for the shallow NNs with one hidden layer were chosen by sequential backwards selection [23]. The activation function for the hidden unit was *SATLINS*, and *PURELIN* for the output layer. These activation functions allow a fast calculation on an embedded system. The NNs were trained by the Levenberg-Marquardt algorithm [24,25], aiming at the minimization of the cross-entropy loss [26]. The model was trained with a maximum number of epochs of 1000 and a maximum of 6 validation failures.

2.6.4. Recurrent Neural Network (RNN)

For the RNN (Figure 6), the maximum number of epochs was set to 100. All other hyperparameters were set as described in Section 2.6.3.

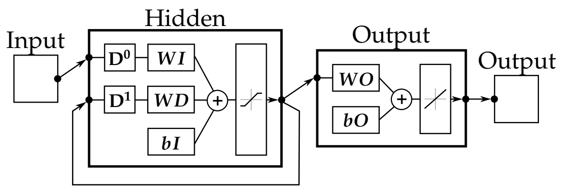

Figure 6. Shallow NN with one recurrent path with a delay of one ($\mathbf{D^1}$).

2.7. *Offline Data Post-Processing*

The model decision was further processed to increase the stability against short wrong decisions. The system should be prevented from incorrect reactions due to individual false classified samples. To allow a transition of the decision, a minimum number of equal consecutive decisions was required. The transition at the output was only conducted after (excl.) n_{slope} equal decisions. For the offline post-processing n_{slope} was set to 2 for the NNs, decision trees and logistic regressions, and to 3 for the RNNs. This parameter introduces an additional delay, which is depending on the model decision frequency and the value of n_{slope}. The parameter n_{slope} was selected in a way to achieve similar delays for the different implementations. Despite the additional delay, n_{slope} is essential for a high stability of the system. This parameter value was selected, as it is a good trade-off between delay and stability, however, it can be adapted to the individual amputee's preference.

2.8. *Implementation*

The features and the models were implemented on an ultra-low-power μC, the ATSAML21E18B from Microchip Technology Inc. (Chandler, AZ, USA) with a 32 bit ARM® CORTEX®-M0+ processor, 256 kB flash, and 32 kB SRAM main memory. The clock frequency was set to 48 MHz to test the implementations. The clock frequency can be reduced (depending on the runtime of the selected model) to decrease power consumption [2].

The analog digital sampling of the EMG signal has been described by Roland et al. [2]. The digital signal processing sampling frequency was set to 2 kHz. For the RNN (250 Hz), the 48 MHz clock provides 192,000 clock cycles (4 ms) between two decisions. The NN, decision tree, logistic regression, and the feature calculation (2 kHz) provide 24,000 clock cycles (500 µs) between two samples.

2.8.1. Fixed-Point Representation

All algorithms were implemented in fixed-point format, as described by ARM Ltd. (Cambridge, UK) [27], which is faster than a floating-point implementation on this μC. For high precision and to avoid overflows or saturations the optimal value range of the variables was selected to exploit the full operating range of the data type. The quantization effects and possible overflows or saturations had already been considered in the offline design. The high performance of the μC 32 bit hardware multiplier was exploited with this fixed-point implementation. All divisions were replaced by shift operations to require low calculation resources.

2.8.2. Online Data Post-Processing

As mentioned above, the time between two decisions was deviating in the offline and online implementations for the NN, decision tree and logistic regression. Therefore, different post-processing parameters were selected. The parameter n_{slope} was set to 20 in the implementation of the NN, decision tree and logistic regression. For the RNN, n_{slope} was set to 3, equal to the offline design.

2.8.3. Signal Delay

The filtered EMG signal and the decision were multiplied at the output; therefore, they should correspond to each other in time. Hence, the input signal was delayed, so that it corresponds to the current decision. The smoothing had a time constant T of 51 ms [2], which introduced a delay. An additional delay was set to 100 ms in the implementation.

3. Results

Table 8 lists the accuracies of the resulting models. Please note that these accuracies were calculated for test data with many transitions. Additionally, many artifacts were in a similar frequency and amplitude range than the contractions. Furthermore, transitions longer than 150 ms were recorded as errors and small signals were removed from the data set. Consequently, these challenging data led to higher error rates in the evaluation but to a highly robust behavior in practical application.

The runtime of the implemented features is shown in Table 9 and of the models in Table 10. The runtime of the decision tree is listed for the shortest and longest branch.

The time constant T of the EMA for the features $MCR2$, $WFL2S$, $WAM1$ was 63.5 ms, and 127.5 ms for all other features.

For the logistic regression, NN and RNN, the feature calculation was followed by the normalization to ensure that the features were in an equal value range. For the decision tree, no normalization was required. The coefficients were determined with the features calculated for the training data. These quantized coefficients were implemented on the μC. The resulting runtime for each feature normalization was 0.53 µs.

Table 8. Accuracies of quantized models in %.

Model\Num. Features	**3**	**6**	**9**	**12**
NN (3 Hidden Units)	97.24	97.30	97.05	98.95
NN (6 Hidden Units)	97.24	98.95	98.89	98.87
NN (9 Hidden Units)	95.79	97.28	98.68	99.06
RNN (3 Hidden Units)	77.47	96.51	98.66	99.22
RNN (6 Hidden Units)	91.63	97.06	99.16	98.66
RNN (9 Hidden Units)	93.20	98.33	99.91	99.67
Decision Tree	96.96	97.93	98.76	98.18
Log. Regression	94.34	95.22	95.60	95.12

Table 9. Runtime of input features in μs at a 48 MHz clock.

$ZCR1$	1.97	$ZCR2$	2.05	$ZCR2S$	2.05	$MCR1$	3.62
$MCR1S$	3.62	$MCR2$	3.55	$SSC1$	2.42	$SSC1S$	2.42
$SSC2$	2.42	$SSC2S$	2.42	$SSC3$	2.42	$SSC3S$	2.42
$SSC4$	2.42	$SSC5$	2.42	$SSC5S$	2.42	$WFL1$	2.57
$WFL1S$	2.57	$WFL2S$	2.57	$MAV1$	2.05	$MAV1S$	2.05
$MAV2$	5.42	$MAV2S$	5.42	$WAM1$	2.95	$WAM2$	3.77
VAR	2.27	$VARS$	2.12				

Table 10. Runtime of models in μs at a 48 MHz clock.

Model\Num. Features	**3**	**6**	**9**	**12**
NN (3 HU)	12.16	15.16	17.78	20.77
NN (6 HU)	18.53	24.14	29.39	35.28
NN (9 HU)	24.89	33.13	41.75	49.99
RNN (3 HU)	19.65	23.02	26.39	29.76
RNN (6 HU)	38.75	45.49	52.23	58.98
RNN (9 HU)	64.22	74.56	84.82	94.93
Decision Tree	0.99–1.15	0.99–1.30	1.15–1.60	1.15–1.67
Log. Regression	1.67	2.12	2.87	3.62

Selected Models

The choice of the final model depends on the requirements. For a highly robust system, the author suggests the RNN with 9 hidden units and 9 features. Figure 7 shows the output of the selected model for a window of the test data. The confusion matrix of the test data for this RNN is presented in Table 11.

For an application with minimum calculation effort but high performance, the decision tree with 3 features ($SSC3$, $ZCR2$, $VAR1S$) and 4 splits is the best choice. In comparison to the RNN, the decision tree has a longer time lag at the transitions (see Figure 8), which leads to lower accuracies. By increasing the signal delay, the accuracy could be improved; however, in this article, long transitions were recorded as an error, because it was aimed at a fast system. The confusion matrix of the selected decision tree is presented in Table 12. Please note that the decision tree has less test samples due to a lower sampling frequency, see Section 2.3. However, the durations of the test data time sequences of the decision tree and RNN are equal.

Table 11. Confusion matrix for selected RNN model (9 hidden units, 9 input features).

$N_{TestSamples}$ **=16,008**	**Predicted Contraction**	**Predicted Artifact**
Actual Contraction	7665	3
Actual Artifact	12	8328

Table 12. Confusion matrix for selected decision tree (3 input features).

$N_{TestSamples}$ **= 4774**	**Predicted Contraction**	**Predicted Artifact**
Actual Contraction	2215	65
Actual Artifact	80	2414

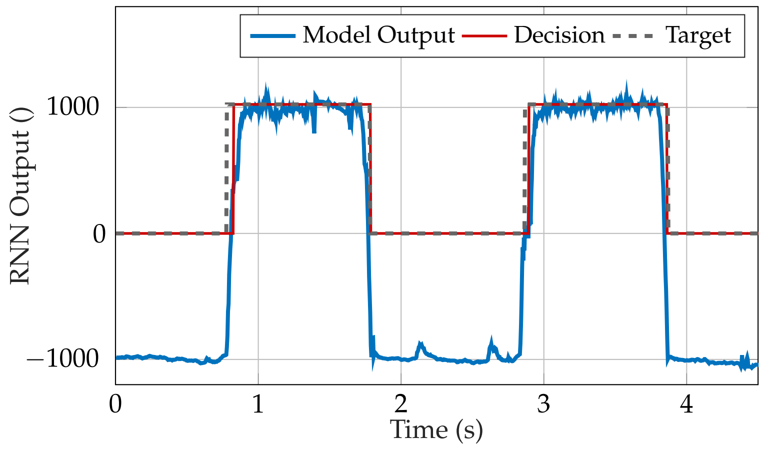

Figure 7. Output, decision and target of the selected RNN for a window of the test data, corrected by the signal delay (Section 2.8.3). The target value is 1024 (= 1 in floating-point format) for contractions and 0 for artifacts. A decision of 1024 leads to an activation of the prosthesis drive. When an artifact is detected, the current output value is maintained or the prosthesis drive is turned off, depending on the selected policy.

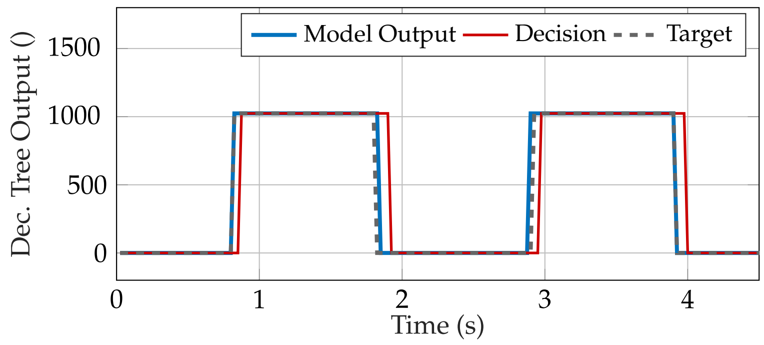

Figure 8. Output, decision and target of the selected decision tree for a window of the test data, corrected by the signal delay (Section 2.8.3). The target value is 1024 (= 1 in floating-point format) for contractions and 0 for artifacts. A decision of 1024 leads to an activation of the prosthesis drive. Decisions for artifacts maintain the current output value or turn off the prosthesis drive, depending on the selected policy.

4. Conclusions

Prosthesis drive activation due to artifacts is highly unpleasant for amputees. Avoiding incorrect activation leads to higher performance and thereby increases the acceptance of the myoelectric prostheses. With the presented models, the artifacts and contraction EMG signals were successfully distinguished with high accuracies. These algorithms substantially improved the robustness of the flexible insulated EMG sensors in real-world environment. Activation of the prosthesis drive due to artifacts is now effectively prevented by the implemented models. This high performance was achieved also by the algorithms' fast reaction to signal changes. All the models were designed to require low calculation effort for implementation on low-power wearable systems. With the shallow models, high accuracies were accomplished, although many transitions of the classes were included in the test data. The algorithms led to a short time between decisions as no Fourier transform had to be calculated. The presented models use hand-crafted time domain features only. Due to the wisely designed features, high accuracies were achieved with shallow models.

The author suggests an RNN or a regular NN for highest accuracies (99.91%/99.06%). For an implementation, which requires very little calculation resources, the author suggests a decision tree (98.76% accuracy).

As mentioned above, the linear separation algorithm presented by Roland et al. [16] has difficulties in detecting short contractions. This algorithm is now outperformed by the new models. Additionally, the need for the calculation of the short-time Fourier transform is obviated.

In the next step, models will be trained on amputees and the EMG sensor system with the presented motion artifact suppression will be applied to real-world prosthesis control. Furthermore, the potential of improving the performance by a random forest or a convolutional NN will be evaluated. The accuracies of ensemble models will be investigated. The runtime of the features and the models will be included to the cost term and a global optimization problem, which also comprises the feature parameters, will be solved. Additionally, the application of these algorithms to the state-of-the-art conductive EMG sensors and to other biosignal sensors (e.g., EEG or ECG) will be investigated. Furthermore, a multiple subject dataset will be acquired, and the presented models will be trained with the new data. The effect of inter-subject variability in the EMG signal will be evaluated. It will be examined whether a model trained on multiple subjects or a model trained on the individual EMG data leads to the best performance.

Author Contributions: T.R. conceived and designed the experiments; T.R. performed the experiments; T.R. trained, implemented and tested the models; T.R. wrote the paper. The author have read and agreed to the published version of the manuscript.

Funding: The author would like to thank the Linz Institute of Technology (LIT) for their financial support.

Acknowledgments: The author would like to thank the Open Access Funding by the University of Linz.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Roland, T.; Wimberger, K.; Amsuess, S.; Russold, M.; Baumgartner, W. An Insulated Flexible Sensor for Stable Electromyography Detection: Application to Prosthesis Control. *Sensors* **2019**, *19*, 961. [CrossRef] [PubMed]
2. Roland, T.; Amsuess, S.; Russold, M.; Baumgartner, W. Ultra-Low-Power Digital Filtering for Insulated EMG Sensing. *Sensors* **2019**, *19*, 959. [CrossRef] [PubMed]
3. Gailey, A.; Artemiadis, P.; Santello, M. Proof of Concept of an Online EMG-Based Decoding of Hand Postures and Individual Digit Forces for Prosthetic Hand Control. *Front. Neurol.* **2017**, *8*. [CrossRef] [PubMed]
4. Al-Timemy, A.; Bugmann, G.; Outram, N.; Escudero, J.; Li, H. Finger Movements Classification for the Dexterous Control of Upper Limb Prosthesis Using EMG Signals. In *Advances in Autonomous Robotics*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 434–435. [CrossRef]
5. Kent, B.A.; Lavery, J.; Engeberg, E.D. Anthropomorphic Control of a Dexterous Artificial Hand via Task Dependent Temporally Synchronized Synergies. *J. Bionic Eng.* **2014**, *11*, 236–248. [CrossRef]
6. Daud, W.M.B.W.; Yahya, A.B.; Horng, C.S.; Sulaima, M.F.; Sudirman, R. Features Extraction of Electromyography Signals in Time Domain on Biceps Brachii Muscle. *Int. J. Model. Optim.* **2013**, 515–519. [CrossRef]
7. Phinyomark, A.; Hirunviriya, S.; Nuidod, A.; Phukpattaranont, P.; Limsakul, C. Evaluation of EMG Feature Extraction for Movement Control of Upper Limb Prostheses Based on Class Separation Index. In *IFMBE Proceedings*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 750–754. [CrossRef]
8. Jahan, M.; Manas, M.; Sharma, B.B.; Gogoi, B.B. Feature extraction and pattern recognition of EMG-based signal for hand movements. In Proceedings of the 2015 International Symposium on Advanced Computing and Communication (ISACC), Silchar, India, 14–15 September 2015. [CrossRef]
9. Merletti, R.; Parker, P. (Eds.) *Electromyography*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2004. [CrossRef]
10. Merletti, R.; Aventaggiato, M.; Botter, A.; Holobar, A.; Marateb, H.; Vieira, T.M.M. Advances in surface EMG: recent progress in detection and processing techniques. *Crit. Rev. Biomed. Eng.* **2010**, *38*, 305–345. [CrossRef] [PubMed]
11. Reaz, M.B.I.; Hussain, M.S.; Mohd-Yasin, F. Techniques of EMG signal analysis: Detection, processing, classification and applications. *Biol. Proced. Online* **2006**, *8*, 11–35. [CrossRef] [PubMed]

12. Zhang, X.; Li, X.; Samuel, O.W.; Huang, Z.; Fang, P.; Li, G. Improving the Robustness of Electromyogram-Pattern Recognition for Prosthetic Control by a Postprocessing Strategy. *Front. Neurorobot.* **2017**, *11*. [CrossRef] [PubMed]
13. Phinyomark, A.; Quaine, F.; Charbonnier, S.; Serviere, C.; Tarpin-Bernard, F.; Laurillau, Y. EMG feature evaluation for improving myoelectric pattern recognition robustness. *Expert Syst. Appl.* **2013**, *40*, 4832–4840. [CrossRef]
14. Roche, A.D.; Rehbaum, H.; Farina, D.; Aszmann, O.C. Prosthetic Myoelectric Control Strategies: A Clinical Perspective. *Curr. Surg. Rep.* **2014**, *2*. [CrossRef]
15. Luca, C.J.D.; Gilmore, L.D.; Kuznetsov, M.; Roy, S.H. Filtering the surface EMG signal: Movement artifact and baseline noise contamination. *J. Biomech.* **2010**, *43*, 1573–1579. [CrossRef] [PubMed]
16. Roland, T.; Baumgartner, W.; Amsuess, S.; Russold, M.F. Capacitively coupled EMG detection via ultra-low-power microcontroller STFT. In Proceedings of the 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Seogwipo, Korea, 11–15 July 2017. [CrossRef]
17. Meyer, M. Digitale Signale. In *Signalverarbeitung*, 7th ed.; Springer: Berlin, Germany, 2014; pp. 150–224.
18. Everett, J.E. The exponentially weighted moving average applied to the control and monitoring of varying sample sizes. In *Computational Methods and Experimental Measurements XV*; WIT Press: Southampton, UK, 2011. [CrossRef]
19. Boslaugh, S. *Statistics in a Nutshell (In a Nutshell (O'Reilly))*; O'Reilly Media: Newton, MA, USA, 2012.
20. Hastie, T.; Tibshirani, R.; Wainwright, M. *Statistical Learning with Sparsity: The Lasso and Generalizations (Chapman & Hall/CRC Monographs on Statistics and Applied Probability)*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2015.
21. Saleh, A.K.M.E.; Arashi, M.; Kibria, B.M.G. *Theory of Ridge Regression Estimation with Applications (Wiley Series in Probability and Statistics)*; Wiley: Hoboken, NJ, USA, 2019.
22. Breiman, L.; Friedman, J.; Stone, C.J.; Olshen, R. *Classification and Regression Trees (Wadsworth Statistics/Probability)*; Chapman and Hall/CRC: Boca Raton, FL, USA, 1984.
23. Isabelle Guyon, A.E. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182. [CrossRef]
24. Levenberg, K. A method for the solution of certain non-linear problems in least squares. *Q. Appl. Math.* **1944**, *2*, 164–168. [CrossRef]
25. Marquardt, D.W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *J. Soc. Ind. Appl. Math.* **1963**, *11*, 431–441. [CrossRef]
26. Jain, A.; Fandango, A.; Kapoor, A. *TensorFlow Machine Learning Projects: Build 13 Real-World Projects with Advanced Numerical Computations using the Python Ecosystem*; Packt Publishing: Birmingham, UK, 2018.
27. ARM Limited. *ARM Developer Suite AXD and Armsd Debuggers Guide*; ARM Limited: Cambridge, UK, 2001.