A Zynq-based dynamically reconfigurable high density myoelectric prosthesis controller

Alexander Boschmann, Georg Thombansen, Linus Witschen, Alex Wiens, Marco Platzner
Department of Computer Science, Paderborn University, 33098 Paderborn, Germany
Email: { alexander.boschmann — thombang— witschen — awiens — platzner }@uni-paderborn.de

***Abstract*—The combination of high-density electromyographic (HD EMG) sensor technology and modern machine learning algorithms allows for intuitive and robust prosthesis control of multiple degrees of freedom. However, HD EMG real-time processing poses a challenge for common microprocessors in an embedded system. With the goal set on an autonomous prosthesis capable of performing training and classification of an amputee's HD EMG signals, the focus of this paper lies in the acceleration of the computationally expensive parts of the embedded signal processing chain: the feature extraction and classification. Using the Xilinx Zynq as a low-cost off-the-shelf system, we present a solution capable of processing 192 HD EMG channels with controller delays below 120 milliseconds, suitable for highly responsive real-world prosthesis control, achieving speed-ups up to 2.8 as compared to a software-only solution. Using dynamic FPGA reconfiguration, the system is able to trade off increased controller delay against improved classification accuracy when signal quality is decreased due to noisy channels. Offloading feature extraction and classification to the FPGA also reduced the system's power consumption, making it more suitable to be used in a battery-powered setup. The system was validated using real-time experiments with online HD EMG data from an amputee to control a state-of-the-art prosthesis.**

I. Introduction

Electromyography (EMG) has been investigated as a control source for modern electronic prosthesis for decades. EMG signals are physiological signals generated during muscular contraction and can be measured as electrical potentials on the skin surface above the remaining muscles. In combination with machine learning algorithms, patients can potentially perform multiple degrees of freedom of their prosthesis intuitively, which enables the user to perform activities of daily living and increases quality of life. Research on myoelectric control has focused on a small number of EMG channels for low computational delay and better clinical applicability. Due to advances in EMG sensor technology, high density EMG (HD EMG) sensor arrays are available today, offering hundreds of signal sources. It has been shown that increasing the number of input channels can improve robustness in non-stationary conditions for traditional EMG signal classification or combinations of HD EMG-based features and classifiers like experimental variogram in combination with Support Vector Machines (SVM) [1] or common spatial patterns classified by Linear Discriminant Analysis (LDA) [2]. However, processing hundreds of EMG channels is challenging for an embedded system and microprocessors used in current prostheses are not suitable for HD EMG data processing in real-time.

The main objective of our work is implementing a controller for autonomous multi-movement prostheses capable of robust HD EMG signal classification. Our application scenario dictates that a prosthesis controller must be an embedded system able to perform the computation autonomously inside the prosthesis. Since classification performance strongly depends on a small controller delay, off-loading the computation to external devices like smartphones over wireless communication channels is unsuitable. Furthermore, embedded prosthesis controllers must also be small in size and show a moderate energy consumption because they are operated by batteries which are typically recharged once a day. These characteristics make HD EMG prosthesis controllers an appealing application domain for FPGAs.

The classification algorithm's runtime is mainly dominated by the feature extraction process applied to independent signal sources, suitable for parallel computation. In our previous work [3], we have presented a Zynq-based HD EMG prosthesis controller, where the widely-used time domain feature extraction process was accelerated on the FPGA, resulting in fast classification speed of one movement decision per millisecond. However, more advanced control schemes have been recently investigated, potentially allowing for higher robustness against noisy input channels which is one of the main obstacles for using HD EMG as a control source for prostheses. In [4], we present a more advanced Zynq-based prosthesis controller using a computationally more challenging signal processing chain, making it more robust against noisy input channels. This robustness comes at the cost of slower prosthesis control and decreased usability.

The novel contribution of this work is an architecture where in addition to hardware-accelerated feature extraction, we also accelerate the classification function on the FPGA, resulting in better controllability of the prosthesis. Furthermore, we introduce the concept of dynamic reconfiguration into the prosthesis controller: Using a periodic heuristic to detect noisy channels in the input data, we are able to reconfigure the FPGA with pre-computed bitstreams, offering a better robustness against noise at the cost of slower processing speed when the signal quality demands it. Finally, we validate our system in real-time experiments using online HD EMG data from an amputee subject to control a real state-of-the-art prosthesis.

The remainder of this work is structured as follows. In Section II we give background information about EMG signal processing and pattern recognition in EMG, ReconOS and discuss related work. In Section III we describe the architecture of our proposed system. In Section IV we validate our system in a real world online experiment with an amputee. Section V contains experimental results and finally we draw a conclusion in Section VI.

978-3-9815370-8-6/17/$31.00

II. Background and Related Work

A. Real-time EMG signal processing

For real-time EMG signal processing, an overlapping window approach is often implemented as illustrated in Figure 1. Features are extracted from the analysis window T_a. Subsequently, the system uses the signal processing time T_d to perform the pattern recognition algorithm. In order to remove misclassifications in the decision stream, the classifier decision is added to a decision queue with the last n decisions. The majority vote winner is then passed to the movement controller. To make the decision stream as dense as possible, a new analysis window is calculated each T_{inc} period. Evidently, T_d has to be smaller or equal to T_{inc}.

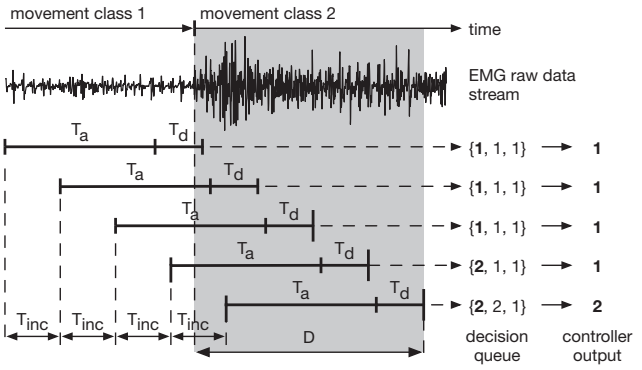

Fig. 1. Windowed approach to EMG classification. The controller delay D extends between the user's change in EMG signals and the first correct controller output, depicted as gray area.

Farrell and Weir [5] define the delay between the user's change of movement and the first correct controller output as controller delay D, as shown in Figure 1 (gray area). D is influenced by T_a, T_{inc}, T_d and n:

$$D = \frac{1}{2} \cdot T_a + \left(\frac{n+1}{2}\right) \cdot T_{inc} + T_d \tag{1}$$

In a relatively extensive study including 20 subjects, Farrell and Weir [6] investigated the trade-off between classification accuracy and responsiveness of prosthesis control. To this extent, they investigated different artificial controller delays for varying tasks. The results show a linear decrease in prosthesis performance with increasing controller delay. The authors therefore suggest keeping the controller delay between 100 and 125 ms or lower as a general system design goal. In this work, one of our main goals is to minimize the controller delay by accelerating the most time consuming parts of the signal processing chain, the feature extraction and classification, in hardware.

B. HD EMG pattern recognition

We extract DSIFT-HOW features from the HD EMG signals and classify them using a linear SVM for prosthesis control. DSIFT (Dense SIFT) features are SIFT [7] features with a fixed orientation extracted in a dense grid of fixed keypoints from an image. When extracted at multiple scales and combined with a histogram-of-visual-words (HOW) approach they form a powerful, rotation and scale invariant feature frequently used in image categorization tasks. In this work, we extract DSIFT-HOW features from 2D image representations (maps) of HD EMG signals and use them for movement classification analog to their original application, image categorization. Because HD EMG maps do not change in scale, we extract the features in a single fixed scale.

Training of a linear SVM using a HOW approach is shown in Algorithm 1. First, HD EMG data are acquired over a sampling window. The individual samples in each channel are then averaged using the root mean square (RMS) method. The RMS values of all channels can be regarded as a heat map of electromyographic activity in a given time interval. This heat map is then scaled up to a higher resolution through linear interpolation and results in a sample image I. This step is necessary because the HD EMG maps' original resolution of 8x24 pixel (each pixel corresponds to the RMS value of one HD EMG electrode) is not large enough to allow meaningful computer vision-based feature extraction.

Algorithm 1 SVM classifier training

procedure SVM_TRAINING($samples, labels$)
  for all training samples **do**
    $I \leftarrow interpolate(rms(samples))$
    $D \leftarrow DSIFT\_extraction(I)$
  end for
  $V \leftarrow train\_vocab(rand\_subset(V, D))$
  $WORD \leftarrow kdtree\_query(D)$
  $HOW \leftarrow create\_HOW(WORD)$
  $MAP \leftarrow homkermap(HOW)$
  $(W, b) \leftarrow train\_linear\_SVM(MAP, labels)$
end procedure

DSIFT features D are calculated for each sample image of a training session. A small random subset of DSIFT features is selected to train a vocabulary V of visual words using K-means clustering. In order to achieve high performance in lookups, the vocabulary is stored in a Kd-tree data structure. Querying the Kd-tree with DSIFT features of remaining images assigns each frame to a corresponding visual word $WORD$. For each image, histograms of visual words (HOW) are constructed.

In order to classify unknown sample data with SVM prediction, HOWs need to be constructed. This processes is summarized in Algorithm 2. Analog to the training phase, first, RMS values of sample windows are calculated and interpolated to form an image I. Then, DSIFT features D are extracted on this image and mapped to visual words $WORD$ by querying the Kd-tree. Depending on image partitioning, histograms of visual words HOW are constructed and encoded with the homogeneous kernel map. In a last step, the linear SVM predicts a movement class later used as controller output.

The image can be partitioned in x and y direction. Then, multiple histograms are obtained, one for each partition and later concatenated. Increasing the number of partitions results in more complex features leading to increased classification accuracy and robustness, but also to larger computational cost. A homogeneous kernel map of Chi2 kernel (HKM) is used to calculate the feature map (MAP) for the concatenated histogram vectors of visual words. This feature transformation step makes it possible to use a fast linear SVM instead of a

more complex kernel function. Finally, a linear SVM is trained using the feature map and the known labels as input.

Algorithm 2 SVM classification

1: **procedure** SVM_PREDICT($samples$)
2: $I \leftarrow interpolate(rms(samples))$
3: $D \leftarrow DSIFT\_extraction(I)$
4: $WORD \leftarrow kdtree\_query(D)$
5: $HOW \leftarrow create\_HOW(WORD)$
6: $MAP \leftarrow homkermap(HOW)$
7: $Y \leftarrow argmax(W \cdot X + b)$
8: **end procedure**

C. Acceleration of EMG processing

Borbely et al. [8] introduced an LDA-based classification design for a Zynq-7000 platform. In their work, a developed system is presented that process EMG data on a PC as well as on the ARM cores of a Zynq-7020 and a hardware implementation for an Zynq's FPGA. While the proposed software system was tested processing up to 8 channels, the performance of their hardware design, considered only the synthesis results but HD EMG processing was not implemented on a real system.

D. ReconOS

The proposed architecture follows the ReconOS multi-threaded approach to hardware acceleration, as detailed in [9]. In this approach, ReconOS acts as a run-time environment on top of an existing POSIX compliant operating system. By utilizing well known operating system abstractions, the application is first partitioned into multiple software threads that communicate and synchronize using standard methods like mutexes and message boxes. Performance critical threads are then moved to hardware where the ReconOS run-time environment provides a seamless integration into the existing multi-threaded application. A key aspect of this integration is handled by ReconOS by providing the same programming model abstractions to both, hardware and software threads.

III. Architecture

Figure 2 presents the top level view of the embedded prosthesis controller. Raw HD EMG data, sampled at 1 kHz, is periodically acquired for a window of 150 samples. Sampled data of one window is averaged with RMS calculation. The result can be interpreted as a heat map of muscular activity. Histograms of DSIFT features of this heat map serve as training and testing features for SVM-based classification of prosthesis movement classes.

We use a ReconOS based architecture under the Linux operating system on the Xilinx Zynq [10] processing platform. The application is divided into hardware and software threads, where the software threads execute on the ARM CPUs and the hardware threads run on the FPGA part of the Zynq. Threads communicate using shared memory and messages boxes - services provided by the ReconOS runtime environment. Message boxes can be seen as 32 bit wide FIFO buffers with synchronous (blocking) access semantics, managed by the operating system. Threads interact with message boxes by posting or receiving single word messages. The partitioning

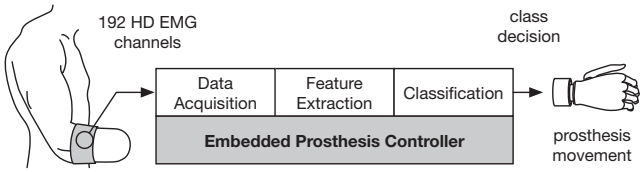

Fig. 2. Top level view of the embedded prosthesis controller. HD EMG data from up to 192 channels are acquired periodically each millisecond. The current samples of all channels are fed into the signal processing chain. The output of the classification is used for controlling the prosthesis.

into threads follows the EMG processing chain with data acquisition, and classification running as software threads, feature extraction and classification as hardware threads. During training, additional software threads run on the CPU. Data flow and synchronization between threads is facilitated by shared memory buffers. Pointers to these buffers are exchanged via message boxes.

Figure 3 shows a more detailed view of the architecture. The figure displays the threads of the application and their interfaces to ReconOS, in particular for the two hardware threads "DSIFT Extraction" and "SVM Prediction". Main components of both hardware threads are state machines for ReconOS control ("FSM"), local block RAMs for input data ("Image RAM", "X RAM") and output data ("Feature RAM", "Y_C RAM") and calculation cores generated with Vivado HLS. The block RAMs are connected to the ReconOS MEMIF on one side and to the HLS cores on the other side. This way ReconOS can transfer data to/from main memory to the local block RAMs given a pointer address via a message box, and the HLS cores can manage local block RAM accesses directly.

For the matrix multiplication part of the SVM classification thread, a newly acquired feature vector X is written to local block RAM and multiplied with the first row of the SVM training matrix W, also stored in local block RAM. As there is only space for one matrix row in block RAM, additional rows of W are stored subsequently and multiplied with X until the entire matrix is processed and the resulting scores vector Y_c can be written back to main memory. Values of W_c are stored in 16-bit fixed point format, two values in each block RAM word. Values of X and Y_c are in single precision floating point format. Experiments showed that approximating the data types has no negative effect on classification accuracy, while it saves FPGA resources and calculation time.

The system is capable of reconfiguration in case the signal quality decreases due to noisy HD EMG channels. In such a case a larger partitioning is required for more accurate classification results. We implemented a simple heuristic based on the Hampel identifier as described in [11] for outlier detection in the HD EMG data. This software thread is called periodically to estimate the number of noisy HD EMG channels and has a runtime of about 1 s using a window of 150 samples as input. In the current state of development, our system uses a lookup table to decide whether the FPGA should be reconfigured and, if so, which bitstream representing different partitioning levels should be used for reconfiguration. As the sizes of W and X increase with larger partitionings, more block RAM cells are needed to ensure low classification time for acceptable controller delay. In this case, DSIFT feature extraction needs to be scaled down to free up block RAM resources. During

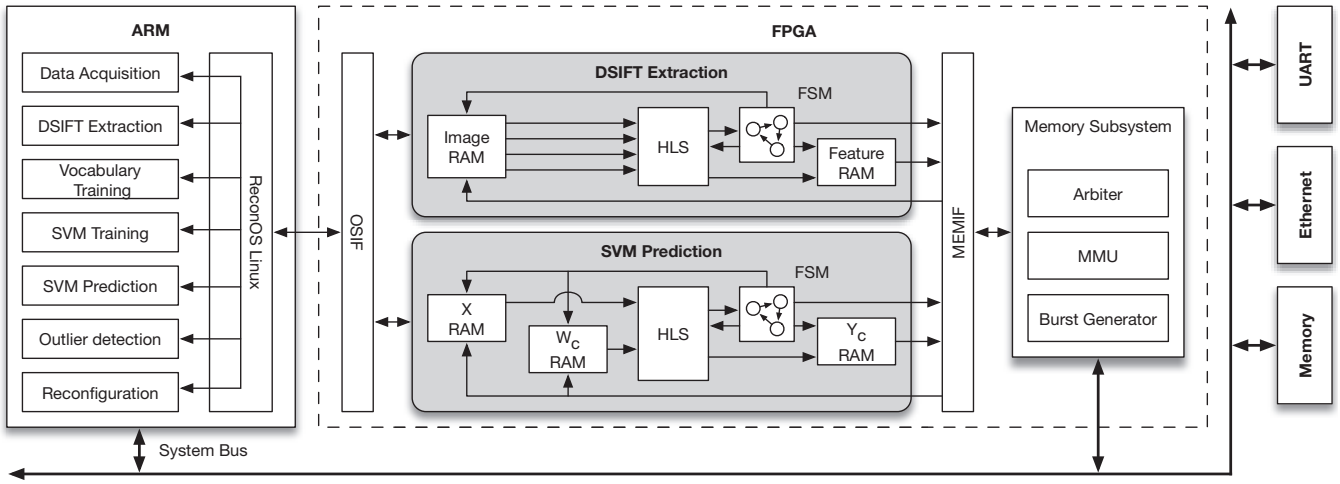

Fig. 3. Application architecture with focus on the feature extraction and SVM classification hardware threads. The ReconOS runtime system enables communication between hardware and software through OS (OSIF) and memory interfaces (MEMIF). The feature extraction and classification units are streaming processors (HLS) internal to the hardware threads.

reconfiguration, ReconOS and all its threads are suspended and prosthesis control is halted until reconfiguration is finished.

IV. Functional Validation

We conducted real-time experiments with an 18 years old male trans-radial amputated test subject to validate our embedded controller architecture under real-world conditions. To this extent, we acquired data from 192 HD EMG sensors to control a modern hand prosthesis in real-time. For data acquisition, we used an adhesive electrode grid (8 rows $\times$ 24 columns) on the amputee's residual limb. The grid was attached to an OT Bioelettronica EMG-USB2 HD EMG DAQ [12]. The HD EMG DAQ was connected to a PC used to send the sensor data to the ZedBoard via TCP/IP. The classifier on the embedded prosthesis controller was trained with EMG data of 6 movements (hand open, lateral grip, pronation, supination, wrist extension and flexion) to match the real prosthesis' available degrees of freedom. After processing the EMG data, the control signals were sent back from the ZedBoard to the PC. A state-of-the-art Otto Bock Michelangelo hand prosthesis was connected to the PC via Bluetooth to perform the classified movements. The experimental setup is depicted in Figure 4. The experiments showed that our system performs well under real-time conditions using a HD EMG system to control a modern commercial multi-movement prosthesis. The HD EMG data acquired during this experiment was used for further offline evaluation in the reconfiguration experiment described in the following section.

V. Experimental Results

We implemented the architecture described in Section III on the ZedBoard development platform. The ZedBoard features a Xilinx Zynq XC7Z020 SoC, 512 MB DDR system memory and various I/O interfaces, such as USB (for UART and JTAG) and Ethernet. While our setup supports both, training and classification, the classification of the EMG signal is more time critical because data must be processed in real-time and large controller delays (above 125 ms) are

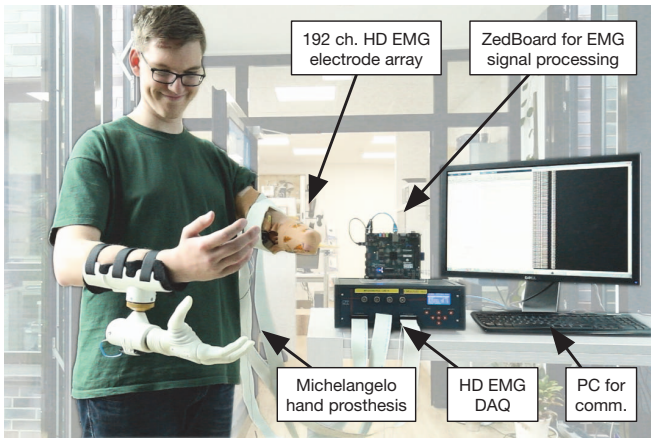

Fig. 4. Experimental setup used for the real-time experiment.

deemed inadequate for sustained prosthesis use. For the evaluation of the application recorded real-world HD EMG data was used for tests using 192 channels.

Figure 6 shows the classification times for three different architectures and five different partitioning configurations from (1,1) to (5,5). Four individual parts during classification are shown, DSIFT-HOW feature extraction, Kd-tree query, SVM prediction and miscellaneous, containing histogram binning, homogeneous kernel mapping and communication overhead. The first group shows a pure software solution (SW-SW), where the DSIFT feature extraction takes the largest portion of calculation time, however, with increasing numbers of partitions SVM prediction becomes more and more influential. In the second group (HW-SW) significant speed-ups have been achieved by extracting the DSIFT features in hardware. Performing the matrix multiplication of SVM prediction in hardware result in additional speed-ups (HW-HW). In this mode the effect of reconfiguration can be observed: For partitioning configurations (1,1) to (4,4) DSIFT extraction

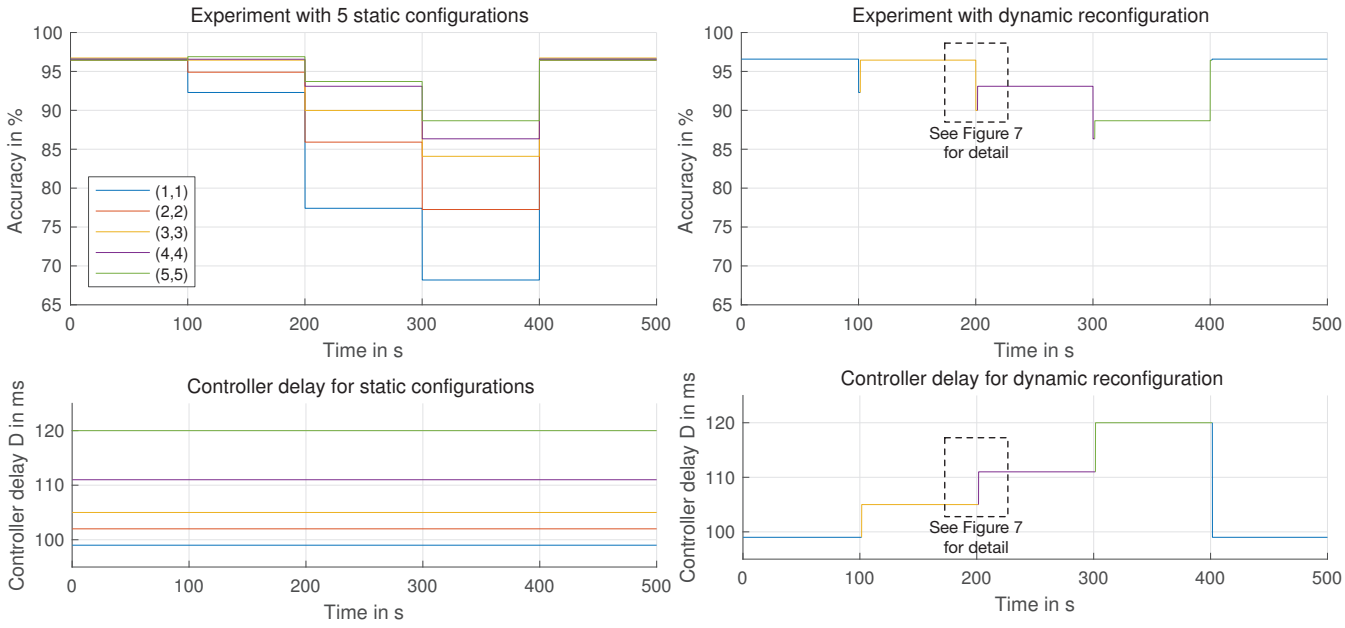

Fig. 5. Accuracy (top) and controller delay (bottom) of a static controller (left) and a dynamically reconfigurable controller (right).

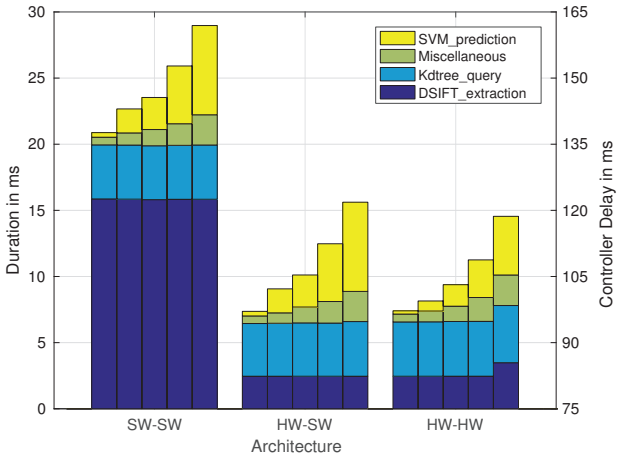

Fig. 6. Classification times for three different architectures and partitioning configurations from (1,1) to (5,5). Vocabulary size: 600 and 8 movement classes.

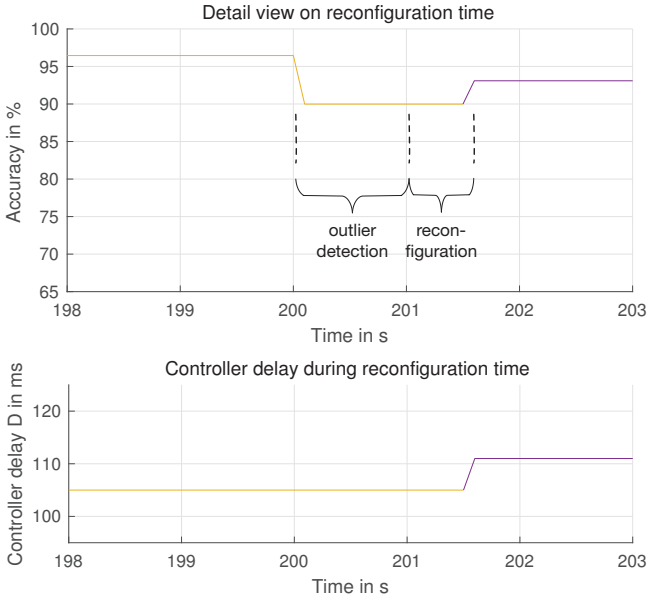

Fig. 7. Accuracy (top) and controller delay (bottom) during reconfiguration process.

operates in its fastest setup, i.e., exploiting parallelism by using 10 ports to access the image RAM. For configuration (5,5) FPGA resources previously denoted to feature extraction, i.e., BRAMs, are shifted towards SVM prediction, making matrix multiplication with larger dimensions possible in hardware. The right axis of Figure 6 shows controller delays for corresponding calculation times. Both HW-SW and HW-HW can perform within the bounds of acceptable controller delay of up to 125 ms. Further benefits of the HW-HW mode regarding energy consumption are discussed later. Figure 8 shows the FPGA resource utilization of FFs, LUTs, DSPs and BRAMs for the architecture with the partitioning configurations (1,1) to (4,4) (left) and configuration (5,5) (right). While FFs, LUTs and DSPs are nearly constant between configurations, BRAM utilization is almost inversed. Note that in both configurations BRAM utilization is almost 100 %, maximizing memory bandwidth.

Figure 5 shows accuracy (top) and controller delay (bottom) for a static (left) and dynamic (right) configuration. In the static case, one partitioning configuration between (1,1) and (5,5) is preloaded and used during the entire experiment. In the dynamic case different configurations are selected optimizing the classification accuracy based on outlier detection of raw EMG data. To evaluate our system, we tested the classification performance for noisy EMG channels. To simulate noisy

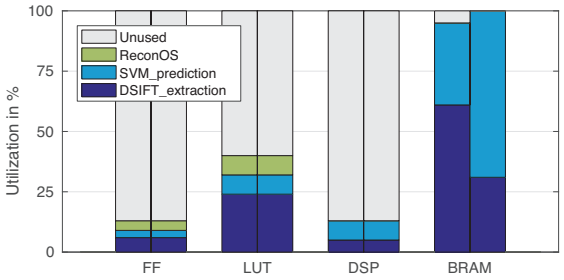

Fig. 8. Utilization of the FPGA's resources.

channels, we added Gaussian distributed noise with the same standard deviation of the selected EMG data to randomly chosen channels during the test phase. In three discrete steps at $t = 100$ s, $t = 200$ s and $t = 300$ s noise is applied to 16, 36 and 46 channels, simulating a real-world scenario during prosthesis use. For noiseless signals ($t = 0 \ldots 100$ s, $t = 400 \ldots 500$ s) all configurations achieve classification accuracies of ≈ 97 %. If more noise is applied, configurations with higher partitioning increase classification accuracy compared to configurations with lower partitioning. On the other hand controller delay is best for low partitioning compared to higher partitioning. The adaptive controller shown on the right side of Figure 5 dynamically reconfigures the configuration that is best suited for the current noise level and, thus, tries to solve this trade-off. Noisy channels are detected by an outlier detection heuristic and the appropriate configuration is selected to maximize classification accuracy while minimizing controller delay. Figure 7 illustrates the process of reconfiguration and its effect on accuracy and controller delay. When more noise is applied at 200 s, it takes ≈ 1 s until outliers are detected, a suitable configuration is selected and another ≈ 0.5 s until the FPGA is reconfigured. After reconfiguration is completed, a better classification accuracy is achieved at the cost of a larger controller delay. On the ZedBoard, a 10 mΩ

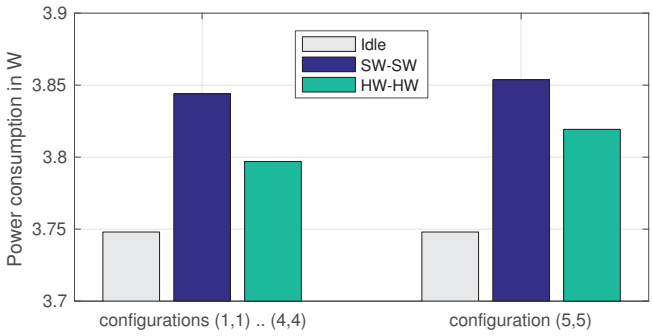

Fig. 9. Power consumption during idle, SW-SW and HW-HW mode for two FPGA configurations. Left: Partitioning (1,1) to (4,4). Right: Partitioning (5,5)

shunt resistor is available to measure the consumed power proportional to the resistor's measured voltage. To measure the voltage, a voltage monitor (LTC4151) was connected to the pin header that exposes the resistor which is positioned in series with the 12 V input power supply. The voltage monitor was read out using the I2C bus connected to the ZedBoard. The power was recorded for two different processing modes: Feature extraction and classification performed in software (SW-SW) or in hardware (HW-HW). The results are depicted in Figure 9. Our findings indicate that performing the extraction and classification in hardware not only decreases processing time but also reduces the power consumption in comparison to processing in software. Although the relative difference for power consumption between operation modes does not seem significant, the cumulative effect during prosthesis control over longer periods of time makes this observation relevant.

VI. Conclusion

We have presented a dynamically reconfigurable myoelectric prosthesis controller implemented on a Zynq platform. The system is capable of processing 192 channel EMG signals and operates with controller delays below 120 ms, suitable for highly responsive real-world prosthesis control. This was achieved by performing computationally intensive steps of the processing chain, namely feature extraction and parts of the classification, on the FPGA. The system is able to utilize dynamic FPGA reconfiguration to adapt to noisy input data, trading off increased controller delay against improved movement recognition accuracy. Offloading feature extraction and classification to the FPGA also reduced the system's power consumption. We validated our system using real-time experiments with online HD EMG data from an amputee to control a state-of-the-art prosthesis.

References

[1] A. Stango, F. Negro, and D. Farina, "Spatial Correlation of High Density EMG Signals Provides Features Robust to Electrode Number and Shift in Pattern Recognition for Myocontrol," *IEEE Trans. Neural. Syst Rehabil. Eng.*, vol. 23, no. 2, pp. 189–198, Mar. 2015.

[2] L. Pan, D. Zhang, N. Jiang, X. Sheng, and X. Zhu, "Improving robustness against electrode shift of high density EMG for myoelectric control through common spatial patterns," *J. Neuroeng. Rehabil.*, vol. 12, no. 1, pp. 1–16, 2015.

[3] A. Boschmann, A. Agne, L. Witschen, G. Thombansen, F. Kraus, and M. Platzner, "FPGA-based acceleration of high density myoelectric signal processing," in *Proc. IEEE Int. Conf. on ReConFigurable Computing and FPGAs ReConFig*, 2015.

[4] A. Boschmann, A. Agne, G. Thombansen, L. Witschen, F. Kraus, and M. Platzner, "Zynq-based Acceleration of Robust High Density Myoelectric Signal Processing," *J. Parallel Distr. Com., Submitted.*

[5] T. R. Farrell and R. F. Weir, "Analysis Window Induced Controller Delay for Multifunctional Prostheses," in *Proc. MyoElectric Controls Symposium (MEC)*, 2008.

[6] ——, "The Optimal Controller Delay for Myoelectric Prostheses," *Trans. Neural Syst. Rehabil. Eng.*, vol. 15, no. 1, pp. 111–118, Mar. 2007.

[7] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. Int. Conf. on Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[8] B. J. Borbely, Z. Kineses, Z. Vorohazi, Z. Nagy, and P. Szolgay, "Analysis of myoelectric signals using a Field Programmable SoC," in *Eur. Conf. Circuit Theory and Design (ECCTD)*, 2013.

[9] A. Agne, M. Happe, A. Keller, E. Lubbers, B. Plattner, M. Platzner, and C. Plessl, "ReconOS: An Operating System Approach for Reconfigurable Computing," *IEEE Micro*, vol. 34, no. 1, pp. 60–71, 2014.

[10] *Zynq-7000 Technical Reference Manual*, Xilinx, February 2015. [Online]. Available: http://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf

[11] H. R. Marateb, M. Rojas-Martínez, M. Mansourian, R. Merletti, and M. A. Mañanas Villanueva, "Outlier detection in high-density surface electromyographic signals," *Medical & Biological Engineering & Computing*, vol. 50, no. 1, pp. 79–89, Jun. 2011.

[12] *Quattrocento Multichannel Data Sheet*, OT Bioelettronica, 2015. [Online]. Available: http://www.otbioelettronica.it/attachments/article/72/Quattrocento.pdf