CHALMERS
UNIVERSITY OF TECHNOLOGY

An Accurate and Real-time Method for Resolving Superimposed Action Potentials in MultiUnit Recordings

Downloaded from: https://research.chalmers.se, 2025-12-04 09:29 UTC

Citation for the original published paper (version of record):
Shirzadi, M., Marateb, H., McGill, K. et al (2023). An Accurate and Real-time Method for Resolving Superimposed Action Potentials in MultiUnit
Recordings. IEEE Transactions on Biomedical Engineering, 70(1): 378-389.
http://dx.doi.org/10.1109/TBME.2022.3192119

N.B. When citing this work, cite the original published paper.

© 2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

(article starts on next page)

An Accurate and Real-time Method for Resolving Superimposed Action Potentials in MultiUnit Recordings

Mehdi Shirzadi, Student *Member*, *IEEE*, Hamid R. Marateb*, Senior *Member*, *IEEE*, Kevin C. McGill, Silvia Muceli, *Senior Member*, *IEEE*, Miguel A. Mañanas, *Member*, *IEEE*, and Dario Farina, *Fellow*, *IEEE*

Abstract—*Objective*: Spike sorting of muscular and neural recordings requires separating action potentials that overlap in time (superimposed action potentials (APs)). We propose a new algorithm for resolving superimposed action potentials, and we test it on intramuscular EMG (iEMG) and intracortical recordings. *Methods*: Discrete-time shifts of the involved APs are first selected based on a heuristic extension of the peel-off algorithm. Then, the time shifts that provide the minimal residual Euclidean norm are identified (Discrete Brute force Correlation (DBC)). The optimal continuous-time shifts are then estimated (High-Resolution BC (HRBC)). In Fusion HRBC (FHRBC), two other cost functions are used. A parallel implementation of the DBC and HRBC algorithms was developed. The performance of the algorithms was assessed on 11,000 simulated iEMG and 14,000 neural recording superpositions, including two to eight APs, and eight experimental iEMG signals containing four to eleven active motor units. The performance of the proposed algorithms was compared with that of the Branch-and-Bound (BB) algorithm using the Rank-Product (RP) method in terms of accuracy and efficiency. *Results*: The average accuracy of the DBC, HRBC and FHRBC methods on the entire simulated datasets was 92.16±17.70, 93.65±16.89, and 94.90±15.15 (%). The DBC algorithm outperformed the other algorithms based on the RP method. The average accuracy and running time of the DBC algorithm on 10.5 ms superimposed spikes of the experimental signals were 92.1±21.7 (%) and 2.3±15.3 (ms). *Conclusion and Significance*: The proposed algorithm is promising for real-time neural decoding, a central problem in neural and muscular decoding and interfacing.

***Index Terms*— Biomedical Signal Processing, Decomposition, Electromyography, Neural Decoding, Overlapping Spikes, Resolving Superimposition, Spike Sorting.**

I. Introduction

Neurons produce action potentials, also termed spikes in signal processing terminology, that are detected in extracellular recordings [1] [2]. These signals can be recorded with microelectrodes. With current technologies, it is now possible to measure hundreds to thousands of neurons concurrently [3-5]. The identification and classification of individual neurons' spikes from extracellular recordings is a processing task known as spike sorting [2, 6]. It has similar procedures as intramuscular electromyographic signal (iEMG)

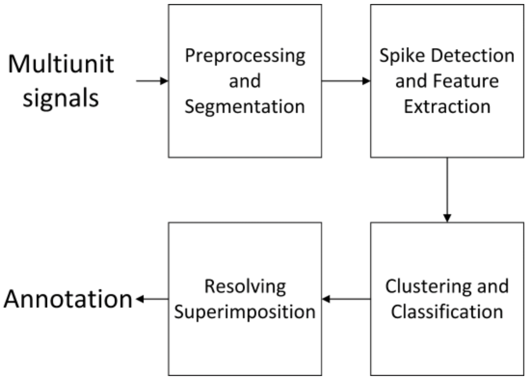
Multiunit signals
Preprocessing and Segmentation
Spike Detection and Feature Extraction
Clustering and Classification
Resolving Superimposition
Annotation

Fig. 1. Procedure of decoding of multiunit signals, which is similar in intramuscular electromyographic signal (iEMG) decomposition and spike sorting.

M. Shirzadi is with the Department of Automatic Control, Biomedical Engineering Research Center, Universitat Politècnica de Catalunya, BarcelonaTech (UPC), Barcelona, Spain (email: mehdi.shirzadi@upc.edu).

H. R. Marateb* is with the Biomedical Engineering Department, Faculty of Engineering, the University of Isfahan, Isfahan, Iran and Department of Automatic Control, Biomedical Engineering Research Center, Universitat Politècnica de Catalunya, BarcelonaTech (UPC), Barcelona, Spain (emails: h.marateb@eng.ui.ac.ir, hamid.reza.marateb@upc.edu).

K. C. McGill was with the VA Palo Alto Health Care System, Palo Alto, CA, USA (retired; email: kcmcgill43@gmail.com).

S. Muceli is with the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden (email: muceli@chalmers.se).

M. A. Mañanas is with the Department of Automatic Control, Biomedical Engineering Research Center, Universitat Politècnica de Catalunya, BarcelonaTech (UPC), Barcelona, Spain and Biomedical Research Networking Center in Bioengineering, Biomaterials, and Nanomedicine (CIBER-BBN), Barcelona, Spain (email: miguel.angel.mananas@upc.edu).

D. Farina is with the Imperial College London, London SW7 2AZ, U.K. (e-mail: d.farina@imperial.ac.uk).

Corresponding author: Hamid Reza Marateb (e-mail: h.marateb@eng.ui.ac.ir).

This work was supported by the Beatriu de Pinós post-doctoral programme from the Office of the Secretary of Universities and Research from the Ministry of Business and Knowledge of the Government of Catalonia programme (#2020 BP 00261) (HRM), the Spanish Ministry of Science and Innovation (project PID2020-117751RB-I00) (MS, MAM), Chalmers Life Science Engineering Area of Advance and the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 846679 (INFANTPATTERNS) (SM), and the European Research Council Synergy project Natural BionicS (contract #810346) (DF).

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

decomposition into the constituent trains of motor unit action potentials (MUAPs) [7]. Direct analysis of neural codes on such multiunit recordings, either by spike sorting or iEMG decomposition, is known as neural decoding [8].

Neural decoding is an essential and challenging step in processing neural and muscular signals. It has applications in neuroscience and neurotechnology, such as in subthalamic nucleus recordings in Parkinson's disease [6], brain-computer interfacing [1, 9], monitoring the activity of single neurons *in vivo* [10], and so on. The steps of multiunit signal decoding are shown in Fig. 1. A central problem in this decoding task is identifying action potentials overlapping in time (a.k.a., superimposition of different spikes) [1]. Superimposed action potentials must be identified to extract the complete neural code and identify physiological mechanisms, such as neural synchronization [1, 4, 11].

Various spike sorting algorithms have been proposed in the literature [3, 5, 11-26], and some include the analysis of overlapping spikes [5, 11, 15-17, 21-26]. Zhang et al. [24] combined different templates to minimize the residual variance using the χ^2test. They generated 14 overlapping spikes from two to three templates with background noise and resolved six of them correctly. Hulata et al. [25] used wavelet packet (WP) decomposition for spike sorting and selected the nine most discriminating WPs for the detection and classification of different spikes. Among 100 overlappings from up to four motor units (MUs), 66 overlappings were correctly resolved. Lewicki [26] proposed a spike sorting algorithm based on a Bayesian network, and among 408 overlappings, 264 cases were correctly identified. Franke et al. [15] proposed a neural decoding algorithm based on Bayes optimal template matching in which overlapping spikes were resolved. The results were promising for partially overlapped spikes with up to two templates. To the best of our knowledge, the superposition resolution algorithms proposed in the literature for spike sorting (those discussed above and others, including [5, 11, 15-17, 21-23]) are suboptimal algorithms in which a maximum of four templates have been considered. Other examples of decoding overlapped action potentials can be found in the field of iEMG decomposition, as discussed in the following.

EMG decomposition identifies the instants of activation of MUs from multiunit EMG recordings [27]. EMG decomposition can be mathematically expressed in the same way as neural spike sorting and has applications for the estimation of muscle architecture [28], the diagnosis of neuromuscular disorders [29], the study of muscle function and coordination, and the control of prosthetic devices [30]. As for spike sorting of neural signals, the most critical stage of iEMG decomposition is resolving superimposition, i.e., identifying action potentials of different MUs when they occur at similar time instants. The mathematical model of overlapped MUAPs is the same as that of overlapped spikes in neural recordings.

Various methods have been proposed to resolve superimpositions for iEMG decomposition. The fastest and most common is the Peel-Off approach [31-34]. The cross-correlation between the superimposed waveform and each template is first calculated with this technique. The most similar template is selected and then subtracted to create the residual signal. This procedure continues until the superimposition is fully analyzed [31].

The Peel-Off method is fast but has very low accuracy for constructive and destructive superpositions. Other methods proposed in the literature consider this problem as an optimization problem. Population-based optimization methods, such as particle swarm optimization (PSO) [35] and genetic algorithms (GA) [36], have been proposed in the literature to resolve MUAP superpositions. Another proposed method is based on branch and bound (BB) optimization [37], in which the entire set of feasible solutions is partitioned into smaller subsets (i.e., discrete-time shifts) to be evaluated systematically. The BB algorithm was shown to have superior performance compared with PSO, GA, and peel-off methods. It is generally an accurate and fast algorithm when up to five MUAPs. However, there are cases where the BB algorithm is too slow and thus not suitable for real-time applications [38].

This article introduces a new method for resolving superimpositions in iEMG decomposition and spike sorting. Its performance and efficiency were assessed and compared with the state-of-the-art methods for superpositions of up to 8 APs.

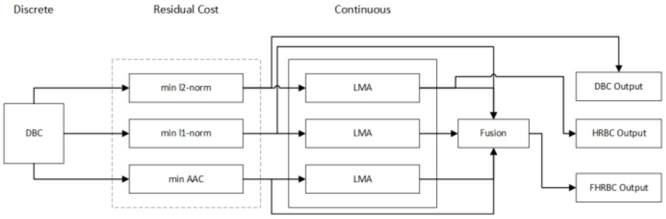

Fig. 2. Scheme of the overall algorithm. a) Discrete-time part of the algorithm. In this section, the permutation of the templates is tested and good local minima in the discrete space are found. The minimum l_2-norm of the residual signal corresponds with the Discrete Brute-force Correlation (DBC) output. b) Continuous-time part of the algorithm. It starts from the selected local minimum in the discrete-time and find the local minima in the continuous-time using the Levenberg-Marquardt algorithm (LMA). While the minimum l_2-norm of the residual signal is considered for continuous time-shift optimization in the High-Resolution BC (HRBC) algorithm, other cost functions (l_1-norm, and are average amplitude change (AAC)) are also considered as inputs to the LMA algorithm in the Fusion HRBC (FHRBC) algorithm. The fusion step considers the best solution, with the minimum l_2-norm as the FHRBC output.

II. Method And Material

The algorithm involves two consecutive procedures. First, a set of discrete-time shifts are selected based on heuristic algorithms. Then, the Levenberg-Marquardt algorithm (LMA) [39] is used to find the continuous-time shifts starting from the selected discrete-time shifts. We propose the discrete algorithm, Discrete Brute force Correlation (DBC), and its two continuous-time versions, namely the high-resolution brute force correlation (HRBC) and its extension as fusion HRBC (FHRBC) (Fig. 2). We focus on the "known constituent" problem in which the set of templates involved in the superposition is known. However, the extension of the methods to the problem of unknown constituent templates out of a template dictionary is straightforward. The inputs to the algorithm are the superimposed waveform and the AP shapes (templates). The outputs are the estimated continuous-time shifts of the templates. The input waveforms $\tilde{x}(nT)$ are first interpolated by a factor of four using the discrete Fourier transform to increase the time resolution [40], as the following:

$$x(t/F) = \frac{X_0}{N} + \frac{1}{N}\sum_{k=1}^{N/2-1}\left[X_k \times exp\left(\frac{j2\pi kt}{FNT}\right) + X_k^* \times exp\left(\frac{-j2\pi kt}{FNT}\right)\right] \quad (1)$$

where, N is the number of samples, T is the sampling interval, X_k is the discrete Fourier transform (DFT) of the sequence $\tilde{x}(nT)$, $*$ denotes complex conjugation, and F is the interpolation factor set to four in our analysis.

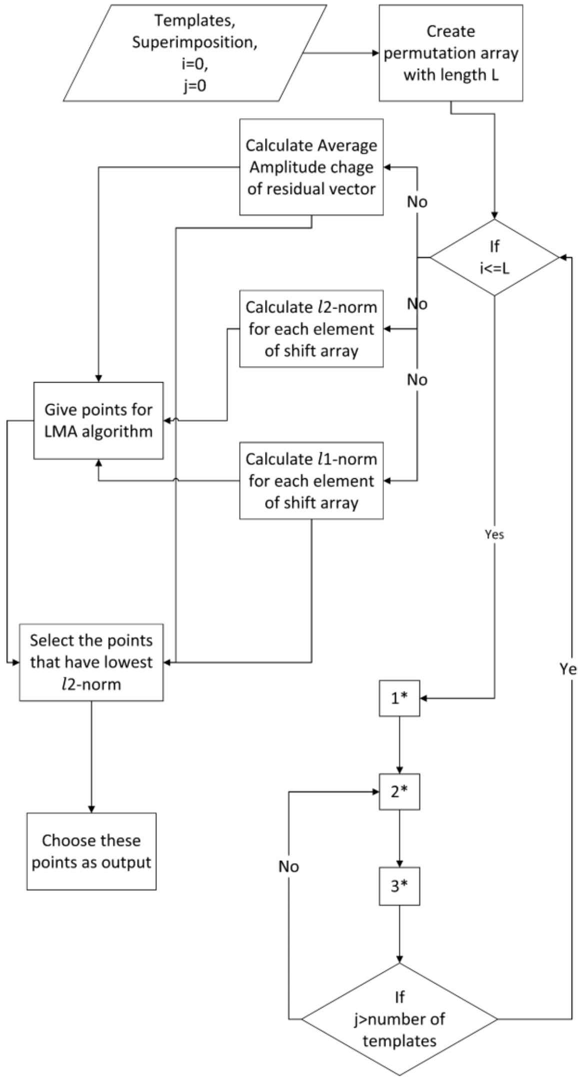
Templates,
Superimposition,
i=0,
j=0
Create permutation array with length L
Calculate Average Amplitude chage of residual vector
No
If i<=L
Calculate l2-norm for each element of shift array
No
No
Give points for LMA algorithm
Calculate l1-norm for each element of shift array
Yes
Select the points that have lowest l2-norm
Yes
1*
2*
No
3*
Choose these points as output
If j>number of templates

Fig. 3. Block diagram of the FHRBC algorithm. 1* = Select the *i*th order of permutation array and name it as current_temps and then increment the index (i++). 2* = Pick *j* template of current_temps array and calculate cross-correlation between it and superimposed waveform, then increment the index (j++). 3* = Find best match of cross- correlation result, save best match in shift array, subtract shifted template from superimposed waveform and save result in residual vector. In fact, the entire $N_t!$ different permutations are ordered in our method, where N_t is the number of templates. We, then, peel off the templates from the superposition waveform in the prescribed order for each permutation. To identify the location of a template with respect to the superposition waveform (at the first permutation order), or the residual signal (at the next orders), (circular) cross-correlation is used. A detailed practical example was provided in sub-section II.F for better clarification.

A. *Discrete Brute-Force Correlation (DBC)*

DBC finds discrete-time shifts of the templates that provide a good fit to the superimposed waveform. The concept is similar to the peel-off algorithm. We observed that peeling off the single template with the best correlation at each analysis iteration overlooks many plausible time shifts and often fails to find the best one. Thus, in DBC we try all the $N_t!$ different permutations in which the N_t templates could be ordered. For each permutation, the templates are peeled off in the prescribed order. The discrete-time shift that maximizes its (circular) cross-correlation with the residual signal is determined for each template. Then the shifted template is subtracted to give the new residual signal. The set of time shifts that gives the best fit (least l_2-norm of the final residual) is saved as the discrete brute-force correlation (DBC) result.

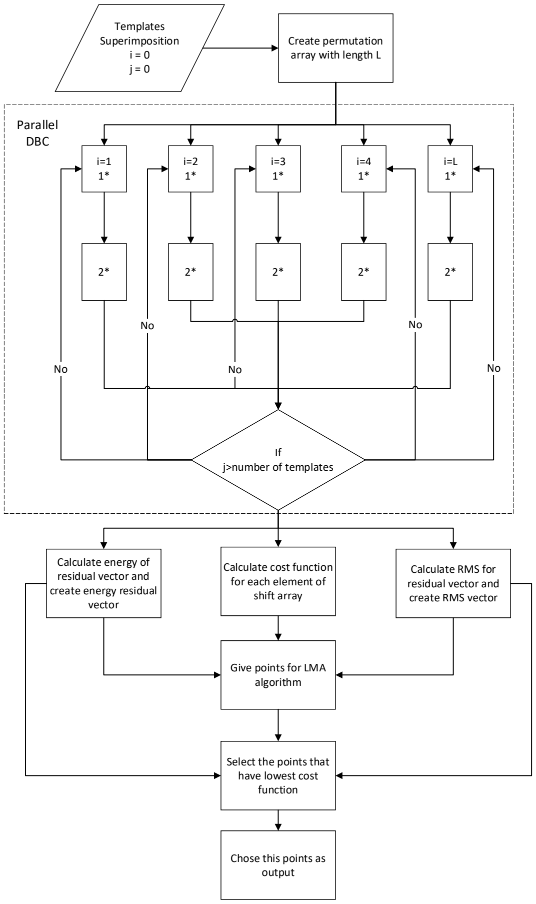
Templates
Superimposition
i = 0
j = 0
Create permutation array with length L
Parallel DBC
i=1
1*
i=2
1*
i=3
1*
i=4
1*
i=L
1*
2*
2*
2*
2*
2*
No
No
No
No
No
No
If
j>number of templates
Calculate energy of residual vector and create energy residual vector
Calculate cost function for each element of shift array
Calculate RMS for residual vector and create RMS vector
Give points for LMA algorithm
Select the points that have lowest cost function
Chose this points as output

Fig. 4. Block diagram of parallel FHRBC. 1* = Pick *j* template of current_temps array and calculate cross-correlation between it and the superimposed waveform, then increment the index (j++). 2* = Find best match of cross-corr result, save best match in shift array, substract shifted template from superimposed waveform and saveresult in residual vector. The DBC algorithm is implemented in parallel for the parallel FHRBC algorithm, as shown by “parallel DBC” in the flowchart. Each permutation set is performed as a separate process in a parallel structure. For example, if we have five input templates, 5! (=120) permutation sets must be analyzed. These sets are independent and could be implemented as separate procedures. However, the cost functions must be calculated when each of which is terminated. This implementation is efficient when the number of involved units is rather high, since the running time of the data exchanges and moderation of the parallel structures is not negligible when few units are analyzed.

B. High-Resolution Brute-Force Correlation (HRBC)

The result of the DBC algorithm is passed on to the LMA procedure. The LMA algorithm (discussed in the following) uses the initial discrete-time points as the initial points and uses continuous-time optimization to fine-tune the results (HRBC).

C. Levenberg–Marquardt algorithm

We consider the least square cost function of the superimposition problem in the frequency domain as the following [40] :

$$e^2 = \frac{1}{N}\sum_{k=0}^{N-1}|D_k|^2 \tag{2}$$

where,

$$D_k \equiv X_k - S_{1,k,\phi_1} - \cdots - S_{M,k,\phi_M} \tag{3}$$

$$S_{m,k,\phi} \ \equiv S_{m,k}\exp\left(\frac{j2\pi k\phi}{N}\right) \tag{4}$$

and where N is the length of the signal, X is the Fourier transform of the superimposition, S_m is the Fourier transform of the mth MUAP template.

The Gradient and Hessian of the cost function are calculated as follows:

$$\frac{\partial e^2}{\partial \phi_i} = \frac{4}{N}\sum_{k=1}^{\frac{N}{2}-1}\left(\frac{2\pi k}{N}\right) Im\{S_{i,k,\phi_i}D_k^*\} \tag{5}$$

$$\frac{\partial^2 e^2}{\partial\phi_i\partial\phi_j} = \begin{cases} \frac{4}{N}\sum_{k=1}^{\frac{N}{2}-1}\left(\frac{2\pi k}{N}\right)^2\left(Re\{S_{i,k,\phi_i}D_k^*\} + \left|S_{i,k,\phi_i}\right|^2\right) & if\ i = j \\ \frac{4}{N}\sum_{k=1}^{\frac{N}{2}-1}\left(\frac{2\pi k}{N}\right)^2 Re\left\{S_{i,k,\phi_i}S_{j,k,\phi_j}\right\} & if\ i \neq j \end{cases}$$

(6)

where *i* and *j* are the MUAP template indices.

Since this cost function is non-linear, the Levenberg-Marquardt algorithm (LMA) was used for the optimization [41-43]. The LMA, also known as damped least-squares, is used to solve non-linear least-squares problems. It can be formulated as:

$$\Phi(x) = arg\ min_{\emptyset}\ e^2 \tag{7}$$

When our initial guess is close to the global minimum, the LMA algorithm converges to the global minimum.

In the next iteration, the parameter $\emptyset$ is replaced with $\emptyset + \delta,$ and Equation (8) is obtained with linearization:

$$e^2(\emptyset + \delta) = \ e^2(\phi) + \ J_i\delta \tag{8}$$

where,

$$J_i = \ \frac{\partial e^2}{\partial \phi_i} \tag{9}$$

The first-order approximation of $e^2(\phi+\delta)$ in vector notation is the following:

$$e^2(\phi + \delta) \ \approx \ \left\|X - S_{k\phi} - J\delta\right\|^2 \tag{10}$$

After taking the derivative and setting it to zero, we have:

$$(J^T J)\delta = \ J^T[X - S_{k\phi}] \tag{11}$$

where J is the Jacobian Matrix. By using Levenberg's contribution, we obtain:

$$(J^T J + \lambda I)\delta = \ J^T[X - S_{k\phi}] \tag{12}$$

where *I* is the identity matrix and λ is the damping factor.

The LMA is, in fact, the combination of the Gauss-Newton and gradient descent algorithms. The method proposed by Marquardt [43] was used to tune the damping factor adaptively. It was initially set to $\lambda^{(0)}$ =0.1 in our study. Its scaling factor (ν) was set to 2. The damping factor was scaled by the scaling factor if the sum square error (e^2) is reduced at each iteration. Otherwise, the damping factor increased to reduce the sum square error.

D. Fusion HRBC (FHRBC)

The use of the l_2-norm does not always result in a discrete-time solution that converges to the globally best continuous-time solution. This is partly due to the background noise and its distribution [35]. In FHRBC, two other cost functions were used in addition to the l_2-norm (Fig. 3): the l_1-norm, which is more robust to noise, and the average amplitude change (AAC) [44]. AAC was defined as follows:

$$AAC = \frac{1}{N}\sum_{i=1}^{N-1}|x_{i+1} - x_i\ | \tag{13}$$

where, N is the length of the residual signal, and x_i is the ith element of the residual signal.

E. Parallel Algorithm

One of the benefits of the HRBC and FHRBC algorithms is that they can use a parallel implementation to improve efficiency. The block diagram of the parallel FHRBC algorithm is shown in Fig. 4. The discrete part of the algorithm (DBC) is, in fact, implemented in parallel in which different permuted time shift sets are tested as a separated process. The DBC algorithm implemented in parallel is referred to as the parallel DBC algorithm.

F. An example of the FHRBC algorithm

In this subsection, an example is provided to describe how the FHRBC algorithm works. A superposition was created by shifting templates 5, 4, and 8 of Fig. 5 by -7.0719, 2.9947, and -7.5464 samples, respectively, and adding them together along with background noise. Time shifts ($\emptyset$) that are fractions of the sampling interval were computed by the inverse Fourier transform of the rotations in the frequency domain [40]:

$$X_{k,\emptyset} = X_k exp\left(\frac{j2\pi k\emptyset}{N}\right)$$

(14)

The algorithm considered six permutations {(1,2,3), (1,3,2), (2,1,3), (3,1,2), (2,3,1), (3,2,1)}. The algorithm peeled off the templates in the prescribed order for each permutation, resulting in the six residual signals shown in Fig. S1: P1-P6. The FHRBC algorithm calculated the l_2-norm, l_1-norm, and AAC of each residual signal and chose the minimum for each norm. In this case, permutation P4 had the minimum l_2-norm, and AAC and P5 had the minimum l_1-norm. The corresponding sets of discrete-time shifts were sent to the LMA procedure, which used them as starting points to find the optimal continuous-time shifts. In our example, both P4 and P5 converged to the same set of continuous-time shifts: -6.9689, 2.8945, -7.5772 samples, with an l_2-norm of 0.54. The final estimation errors of the time shifts were 0.10, 0.10, and 0.03 samples. Table S1 shows the estimated time shifts for each step of the algorithm.

G. iEMG Datasets

Two sets of MUAP templates, taken from iEMG signals from the public domain database at www.emglab.net, were used in our study. They were sampled at 10 kHz and high-pass filtered at 1 kHz [38]. The first set had ten MUAPs with a variety of energies (5000-130,000)$(\mu V)^2$ (Fig. 5), while the second set had six MUAPs with a high degree of similarity (average bivariate correlation coefficients of 0.59 ± 0.28) (Fig. 6). In the first set, we simulated 7000 superimpositions from 2 to 8 MUAPs, while in the second set, we simulated 4000 superimpositions from 2 to 5 MUAPs. The time shifts of the MUAPs were randomly distributed between -1ms to +1ms, which produced more constructive and destructive rather than partial superimpositions.

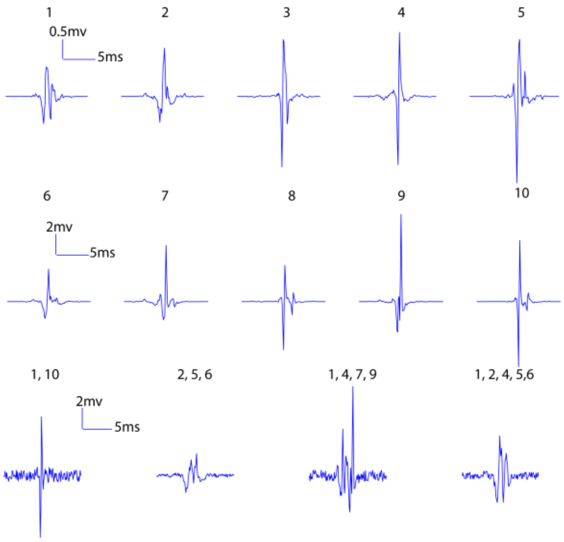

Fig. 5. Ten MUAP templates in the set 1 of iEMG decomposition dataset and some examples of the noisy superimpositions along with the constituent MUAP templates.

For simulating practical situations, we added uniform random noise that varied from -5% to 5% of the difference between maximum and minimum of the signal to each of the samples of the resulting superimposition. Also, for simulating the effect of movement of electrodes, we randomly changed the MUAP amplitudes involved in the superimposition to improve the generalization of the results [38]. The amplitude gain was randomly distributed between 0.7 and 1.3, and was randomly applied to the involved MUAPs.

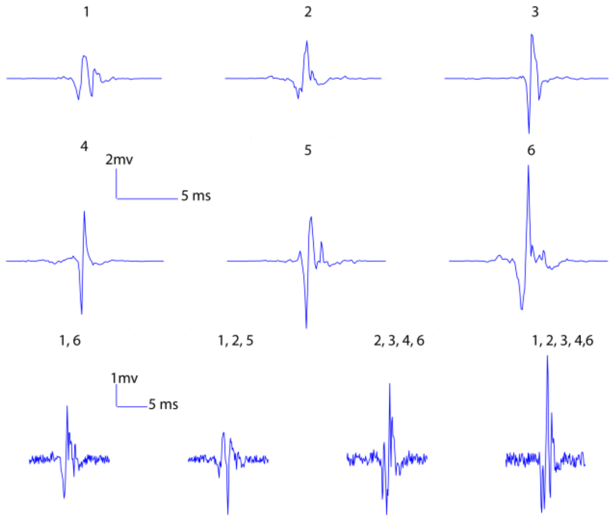

Fig. 6. Six MUAP templates of the set 2 of iEMG decomposition dataset and some examples of noisy superimpositions along with their constituent MUAP templates.

H. Neural Recording Dataset

The benchmark dataset proposed by Pedreira et al. [45] was used for spike sorting. Spike-triggered averaging was used to extract different spikes [46]. Two spike sets were then generated. The first set had 10 spikes with a wide variety of energies (0.75 – 9.79) $(\mu V)^2$ (Fig. 7) while the second set had 11 spikes with a high degree of similarity (correlation 0.58 ± 0.24) (Fig. 8). This dataset was recorded with a 24 kHz sampling frequency.

For each spike set, 1000 different superimpositions of combinations of 2 to 8 spikes (a total of 14000 for the entire set) were simulated. Electrode movement and background noise were simulated as with iEMG.

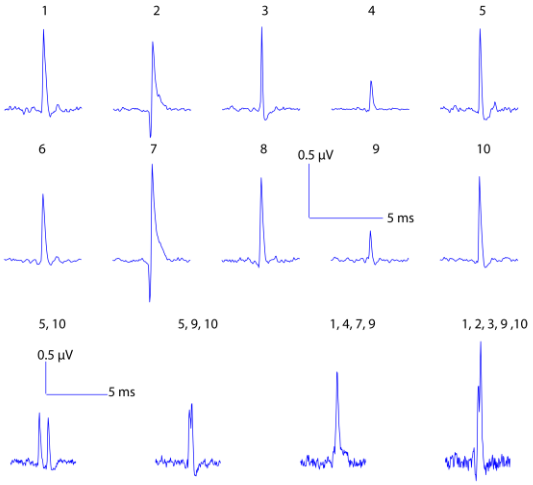

Fig. 7. Different Spikes in the set 1 of neural recording dataset and some examples of noisy superimpositions (overlapping Spikes) (bottom).

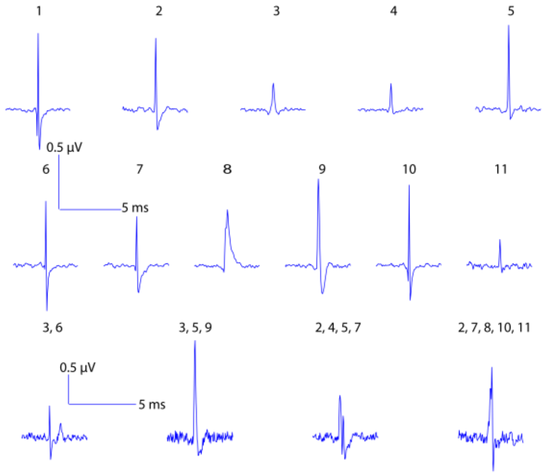

Fig. 8. Different Spikes in the set 2 of neural recording dataset and some examples of noisy superimpositions (overlapping Spikes) (bottom).

I. *Real-time decomposition of experimental signals*

The performance of the proposed DBC algorithm was further assessed in conjunction with a real-time iEMG decomposition algorithm on experimental signals. The iEMG decomposition algorithm proposed by Karimimehr et al. [47] was used in our study. In this algorithm, superimposed MUAPs were not resolved. The active segments with superimposed MUAPs were thus resolved by the DBC algorithm. The identities of the MUs involved in the superposition were estimated from the expected firing times of the MUs based on the firing times of their surrounding discharges [46, 48].

Experimental signals recorded from biceps femoris long-head (BFlh), tibialis anterior (TA), and Vastus Medialis obliquus (VM) muscles were used in our study. The experimental protocol was described elsewhere [47]. Briefly, the constant force plateau regions of isometric contractions were analyzed. The levels of contractions were 5% and 10% MVC for BFlh (five subjects, all men) and 10% and 30% MVC for VM (five subjects, all men). The subjects were asked to increase the strength of the contraction of the TA muscle until the iEMG signal contained from 6 to 12 active MU trains, as judged by the investigators. All subjects were informed about the study protocol in accordance with the Declaration of Helsinki and approved by the local ethics committee. To check the results of the DBC algorithm, the single-channel signals were decomposed by EMGLAB [46] and then manually corrected by an experienced operator. The decomposition results were then corrected using a rigorous a posteriori statistical analysis [49]. These results were considered the gold standard [50].

J. *Validation*

The results of each algorithm were compared with the actual or gold-standard time shifts of the MUAPs or spikes in the superimpositions. Each time-shift error was classified as being in one of these three classes: $< \pm 0.1\ ms$ (essentially correct), $\leq \pm 0.5\ ms$ (close), and $> \pm 0.5\ ms$ (incorrect) [37, 49]. The Identification rate (Id) was used [35, 38] to assess the accuracy of the algorithms:

$$Id = \frac{n_c}{n_i + N_t} \qquad (15)$$

where n_c is the number of essentially correct time shifts, n_i is the number of incorrect time shifts, and N_t is the total number of ground truth MUAPs (or spikes) involved in this superimposition. The analyzed algorithms were ranked overall on the entire datasets using the Rank Product (RP) [38, 51]. Moreover, the complexity of the experimental iEMG signals was assessed by the superposition percentage (% Sup), defined as the number of superimposed spikes divided by the total number of spikes detected in the iEMG signal. A margin of 3 ms was used to define superimposed spikes [7, 52].

The running time of the superposition resolution algorithms was also reported as a measure of computational complexity. The running time of the algorithms was assessed on an Intel(R) Core (TM) i7-8750H 2.20GHz CPU with 16 GB of RAM.

K. *Statistical Methods*

Results are reported as mean ± standard deviation. We used the Generalized Estimating Equation (GEE) method to model factors associated with repeated responses [53]. GEE was used to find significant factors (e.g., set number and number of APs) affecting the performance of the superposition resolution algorithms. The Friedman test was further used to compare the performance of the DBC, HRBC, and FHRBC methods. The Wilcoxon signed-rank test with the Bonferroni correction was used for pairwise comparison, and also to assess whether initial interpolation was required for the DBC, HRBC, and FHRBC methods. Adjusted P-values less than 0.05 were considered significant. The statistical analysis and calculations were performed using the IBM SPSS Statistics for Windows, version 24.0, Released 2016 (IBM Corp., Armonk, NY).

III. Results

A. *Simulated Signals*

The results of the different algorithms on the iEMG sets 1 and 2 for superpositions involving 2 to 5 MUAPs are provided in Tables I and II. The results for the spike sorting sets 1 and 2 for

TABLE I
Results of different algorithms on the iEMG set 1 for combination of 2 to 5 MUAPs

Combination	Algorithm	Id (%)	Time (s)
2	DBC	99.45±6.10	0.001±0.0001
	HRBC	99.70±04.99	0.006±0.0009
	FHRBC	99.70±4.99	0.007±0.0020
	BB	99.90±2.23	0.004±0.0008
3	DBC	97.47±11.34	0.001±0.0002
	HRBC	98.33±9.83	0.008±0.0028
	FHRBC	98.88±8.38	0.012±0.0065
	BB	99.27±6.06	0.012±0.0055
4	DBC	93.51±16.34	0.006±0.0003
	HRBC	94.93±15.17	0.015±0.0033
	FHRBC	96.83±12.07	0.027±0.0104
	BB	96.33±12.78	0.042±0.0356
5	FHRBC	90.86±18.87	0.089±0.0195
	BB	92.38±17.30	0.231±0.2957
	DBC	85.52±22.20	0.033±0.0013
	HRBC	86.94±22.24	0.045±0.0073

superpositions involving 2 to 5 spikes are provided in Tables III and IV.

TABLE II
RESULTS OF DIFFERENT ALGORITHMS ON THE IEMG SET 2 FOR COMBINATION OF 2 TO 5 MUAPS

Combination	Algorithm	Id (%)	Time (s)
2	DBC	99.15±8.47	0.001±0.0001
	HRBC	99.50±7.05	0.005±0.0010
	FHRBC	99.60±6.31	0.007±0.0021
	BB	100.00±0.00	0.005±0.0026
3	DBC	97.89±10.85	0.001±0.0001
	HRBC	98.45±10.04	0.008±0.0021
	FHRBC	99.13±7.80	0.012±0.0050
	BB	100.00±0.00	0.019±0.0072
4	DBC	96.10±13.06	0.006±0.0002
	HRBC	98.01±10.33	0.015±0.0032
	FHRBC	98.70±8.61	0.026±0.0116
	BB	99.92±1.76	0.114±0.0756
5	DBC	92.62±18.03	0.033±0.0014
	HRBC	94.34±17.19	0.044±0.0055
	FHRBC	96.67±13.42	0.097±0.0238
	BB	99.48±5.02	1.026±0.9991

TABLE III
RESULTS OF DIFFERENT ALGORITHM IN THE SPIKE SORTING SET 1 FOR COMBINATIONS OF 2 TO 5 DIFFERENT SPIKES

Combination	Algorithm	Id (%)	Time (s)
2	DBC	96.60±16.84	0.001±0.0002
	HRBC	97.90±14.34	0.008±0.0014
	FHRBC	97.90±14.34	0.012±0.0045
	BB	100.00±0.00	0.012±0.0037
3	DBC	94.98±17.40	0.004±0.0002
	HRBC	96.10±16.25	0.013±0.0042
	FHRBC	96.20±16.48	0.023±0.0102
	BB	100.00±0.00	0.065±0.0357
4	DBC	93.85±15.23	0.017±0.0007
	HRBC	96.20±13.46	0.030±0.0052
	FHRBC	95.97±14.40	0.060±0.0196
	BB	99.92±1.76	0.560±0.4733
5	DBC	93.34±14.72	0.103±0.0037
	HRBC	96.28±12.10	0.119±0.0077
	FHRBC	96.04±13.56	0.227±0.0372
	BB	99.76±2.42	6.291±7.2151

TABLE IV
RESULTS OF DIFFERENT ALGORITHM IN THE SPIKE SORTING SET 2 FOR COMBINATIONS OF 2 TO 5 DIFFERENT SPIKES

Combination	Algorithm	Id (%)	Time (s)
2	DBC	99.60±4.45	0.001±0.0002
	HRBC	99.95±1.58	0.007±0.0009
	FHRBC	99.95±1.58	0.011±0.0032
	BB	100.00±0.00	0.011±0.0040
3	DBC	99.03±6.51	0.004±0.0002
	HRBC	99.66±4.93	0.012±0.0016
	FHRBC	99.66±4.93	0.020±0.0063
	BB	100.00±0.00	0.037±0.0201
4	DBC	98.91±5.97	0.017±0.0006
	HRBC	99.58±4.28	0.028±0.0048
	FHRBC	99.53±4.56	0.049±0.0131
	BB	99.97±0.79	0.164±0.1368
5	DBC	98.26±7.26	0.101±0.0028
	HRBC	98.91±6.41	0.117±0.0050
	FHRBC	99.18±5.32	0.189±0.0305
	BB	99.83±2.23	0.834±0.8904

The GEE test showed that the ranking of the methods in terms of I_d variable was BB, FHRBC, HRBC, and DBC (P-value < 0.05) on the iEMG and neural recording datasets. The ranking for running time was DBC, HRBC, FHRBC, and BB (P-value < 0.05). Thus, BB had the best accuracy on both datasets and DBC had the best running time. The dataset and number of APs significantly influenced the accuracy (P-value < 0.05).

TABLE V
RESULTS OF DIFFERENT ALGORITHMS ON IEMG SET 1 FOR COMBINATIONS 6 TO 8 DIFFERENT MUAPS

Combination	Algorithm	Id (%)	Time (s)
6	DBC	77.11±24.91	0.240±0.0157
	HRBC	79.29±25.37	0.261±0.0173
	FHRBC	85.26±22.66	0.467±0.0531
7	DBC	69.43±24.98	1.976±0.1197
	HRBC	71.10±25.23	2.032±0.1609
	FHRBC	77.74±23.97	3.476±0.5272
8	DBC	62.06±24.73	18.369±1.038
	HRBC	63.29±25.49	18.037±0.7383
	FHRBC	69.34±25.27	26.839±2.6023

TABLE VI
RESULTS OF DIFFERENT ALGORITHMS ON SPIKE SORTING SET 1 FOR COMBINATIONS 6 TO 8 DIFFERENT SPIKES

Combination	Algorithm	Id (%)	Time (s)
6	DBC	91.78±15.15	0.441±0.0202
	HRBC	95.15±13.16	0.512±0.0626
	FHRBC	94.91±14.49	1.186±0.1233
7	DBC	90.04±15.67	3.473±0.2346
	HRBC	93.19±15.18	3.598±0.2848
	FHRBC	93.71±14.78	10.033±2.6471
8	DBC	87.61±16.57	37.842±3.6410
	HRBC	91.03±16.37	37.927±3.1644
	FHRBC	92.28±15.30	78.861±9.1368

TABLE VII
RESULTS OF DIFFERENT ALGORITHMS ON SPIKE SORTING SET 2 FOR COMBINATIONS 6 TO 8 DIFFERENT SPIKES

Combination	Algorithm	Id	Time
6	DBC	97.29±8.88	0.457±0.0462
	HRBC	98.24±7.66	0.511±0.0663
	FHRBC	98.60±6.86	0.825±0.1176
7	DBC	95.91±10.04	3.580±0.1060
	HRBC	97.34±9.13	3.806±0.4226
	FHRBC	97.91±7.67	6.261±0.9126
8	DBC	96.31±8.72	37.945±3.8665
	HRBC	97.73±7.43	38.791±4.6103
	FHRBC	97.98±7.07	61.685±8.6438

Combining the accuracy and running time results, the RP method provided the following overall ranking for combinations of 2-5 APs: DBC, BB, FHRBC, and HRBC.

Although the resolution of superpositions of two to five APs is usually considered in the literature [35-38, 54], it is also necessary to assess the performance of such algorithms when more APs overlap, as it may happen in practice [48]. Thus, we also considered cases with six to eight APs. The performance of the proposed algorithms was assessed on the iEMG set 1 (Table V), spike sorting set 1 (Table VI), and set 2 (Table VII). Since the BB algorithm is relatively slow (e.g., sometimes taking more than days in complex cases involving more than 5 APs), it was not included in the extended analysis. The extended analysis provided the following ranking in terms of accuracy: FHRBC, HRBC, and DBC (P-value <0.05), and running time: DBC, FHRBC, and HRBC (P-value <0.05). The number of APs significantly affected the accuracy of the analyzed methods for the extended analysis (P-value < 0.05).

Combining the accuracy and running time results, the RP method provided the following overall ranking for combinations 6-8: DBC, FHRBC, and HRBC.

The initial interpolation significantly improved the performance of the DBC and HRBC algorithms (adj. P-value <0.05) but not the FHRBC algorithm. Also, the FHRBC and HRBC algorithms significantly outperformed the DBC algorithm in terms of the Id values (adj. P-value <0.05), showing that it is possible to benefit from the optimization block of the HRBC and FHRBC algorithms.

B. *Experimental signals*

The performance of the DBC algorithm on the experimental iEMG signals is shown in Table VIII. The number of MUs ranged from four to 11, while the superposition percentage ranged from 14.14% to 26.96%. The criterion for an essentially correct identification used in the Id formula (Eq. 13) is 0.1 ms (i.e., one sampling interval with the sampling frequency of 10 kHz). Thus, we performed a more rigorous performance assessment than other state-of-the-art studies [7, 52], which used a 1 ms criterion. The superposition percentage and the number of MUs in our study are similar to that of Yu et al. [7].

TABLE VIII
RESULTS OF THE DBC ALGORITHMS ON THE REAL IEMG SIGNALS

Muscle	% MVC	DUR (s)	#MU	% Sup	CORR	Time (ms)	Id (%)
BFlh	5	10	7	19.19	0.13-0.98	1.29±1.02	96.38±14.19
			4	12.86	0.48-0.90	1.24±0.97	89.62±22.04
			5	14.14	0.35-0.97	1.23±0.96	95.30±16.32
	10		6	17.60	0.17-0.97	1.30±1.30	92.26±23.09
			7	20.27	0.54-0.97	1.21±0.97	88.33±28.52
VM	10		7	19.13	-0.30-0.92	1.29±1.15	92.03±19.09
	30		11	24.07	-0.76-0.81	6.06±34.00	95.61±13.77
TA	-	20	7	26.96	0.56-0.96	4.70±26.78	86.87±30.21

BFlh: Biceps Femoris long-head; VM: Vastus Medialis; TA: Tibialis Anterior; DUR: duration of the signal; #MU: the number of active MUs in the gold standard; % Sup: the superposition percentage; CORR: the range of the correlation between MUAPs; Time: the average analysis time 10.5 ms spike epochs; Id: the accuracy of superposition resolution.

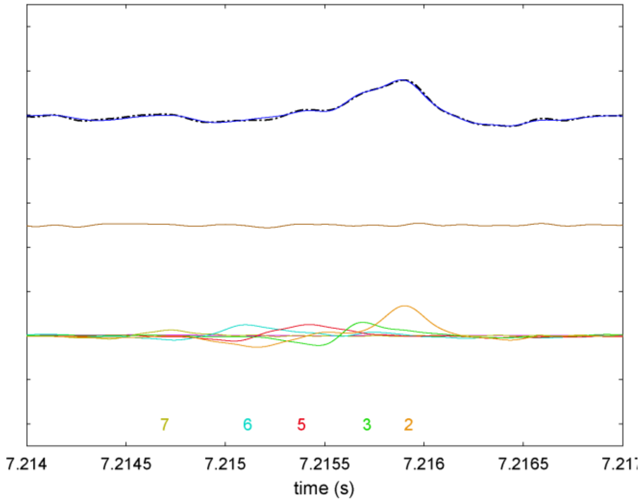

Fig. 9. A sample superimposed MUAP signal from the TA muscle (top, dashed-dotted black), and the reconstructed signal (top, solid blue). The residual signal is shown in the middle. The inclusion of five MUAP templates were identified by the MU discharge estimation algorithm, while their firing times were estimated by the DBC algorithm.

The accuracy values provided in Table VIII reflect the DBC algorithm's performance and the MU discharge estimation algorithm. A correctly resolved superposition from the TA muscle is shown in Fig. 9. Five MUAPs were involved, and the running time of the DBC algorithm was 3.24 ms.

TABLE IX
THE PERFORMANCE (ID) OF THE DBC ON DATASETS WITHOUT NOISE AND THE RUNNING TIME OF THE PARALLEL DBC ON THE NOISY DATASETS

Dataset	Set	Combinations	Id (%)	Time (s)
iEMG	1	2	99.75±4.18	0.018±0.0265
		3	99.02±8.08	0.017±0.0037
		4	98.81±6.38	0.029±0.0063
		5	97.97±6.89	0.057±0.0096
		6	98.29±6.18	0.181±0.0262
		7	98.55±4.97	1.126±0.0765
		8	98.25±4.92	10.460±0.9795
	2	2	99.70±3.86	0.018±0.0352
		3	98.83±7.29	0.019±0.0228
		4	98.88±6.11	0.027±0.0118
		5	98.35±5.67	0.056±0.0208
Spike Sorting	1	2	96.65±16.78	0.013±0.0063
		3	95.37±15.57	0.018±0.0096
		4	94.87±13.36	0.036±0.0089
		5	94.36±12.97	0.104±0.0241
		6	94.13±11.50	0.520±0.0399
		7	94.45±10.16	3.778±0.2342
		8	93.84±9.70	42.946±1.8461
	2	2	99.50±5.90	0.015±0.0040
		3	99.32±5.26	0.021±0.0078
		4	99.36±4.75	0.038±0.0087
		5	99.17±4.60	0.104±0.0062
		6	98.94±4.75	0.503±0.0495
		7	98.98±4.27	3.864±0.1904
		8	98.84±4.27	37.175±2.4920

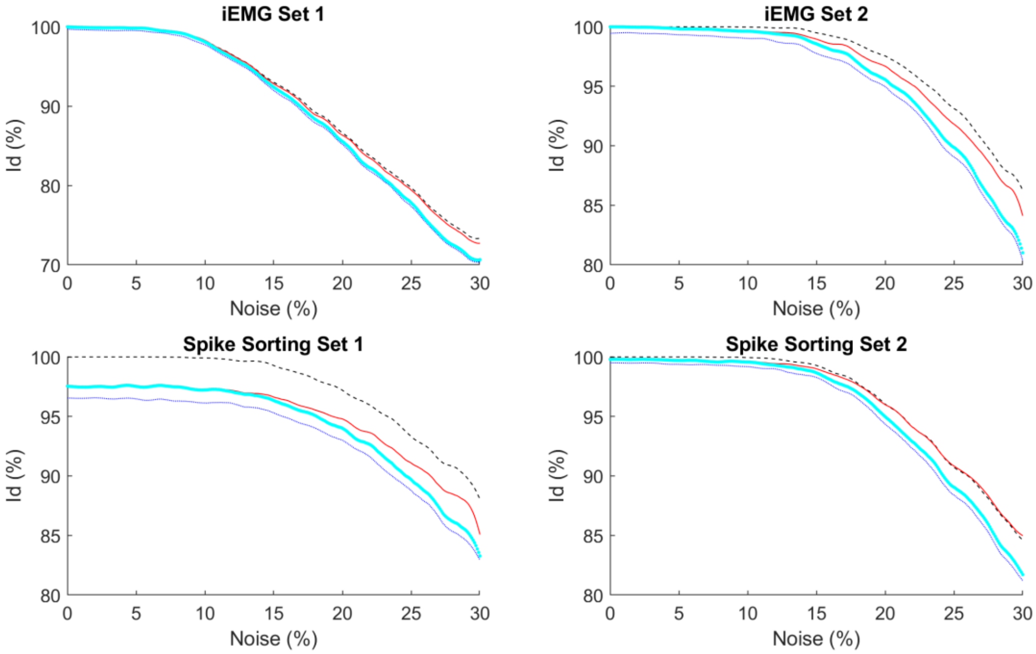

Fig. 10. The accuracy of different algorithms to resolve superimpositions of two MUs for different noise levels. The BB, FHRBC, HRBC, and DBC algorithms were shown in dashed black, solid red, dotted cyan and dashed dotted blue.

C. *Influence of the Background Noise*

Resolving superpositions is an NP-hard problem in optimization [36]. Moreover, it is not a convex optimization problem. The background noise in the superimposed waveform further increases the complexity of the problem by changing the location of the global optimum so that the global minimum of the cost function does not always correspond to the actual time shifts of the involved templates [36]. Our preliminary studies suggested that using a different norm besides the l_2-norm in the DBC algorithm might improve its accuracy. In our results, however, the improvement using the FHRBC algorithm was relatively small. Therefore, for practical purposes, the DBC algorithm using the l_2-norm is acceptable.

TABLE X
RESULTS OF THE PEEL-OFF ALGORITHM ON DIFFERENT COMBINATIONS OF THE SET 1 AND 2 OF THE IEMG AND NEURAL RECORDING DATASETS

Dataset	Set	Combinations	Id (%)	Time (s)
iEMG	1	2	49.98±32.88	0.001±0.0001
		3	31.92±21.13	0.001±0.0001
		4	25.74±16.89	0.001±0.0001
		5	21.49±13.84	0.001±0.0001
		6	19.06±11.96	0.001±0.0057
		7	18.10±10.73	0.001±0.0009
		8	17.01±9.79	0.001±0.0007
	2	2	49.77±37.33	0.001±0.0001
		3	30.21±23.06	0.001±0.0001
		4	24.82±17.93	0.001±0.0001
		5	20.48±14.15	0.001±0.0001
Spike Sorting	1	2	57.95±35.62	0.001±0.0001
		3	28.79±18.77	0.001±0.0001
		4	21.21±14.43	0.001±0.0001
		5	17.50±10.26	0.001±0.0001
		6	15.37±9.36	0.001±0.0001
		7	13.50±7.43	0.002±0.0014
		8	12.30±7.22	0.002±0.0008
	2	2	53.38±35.12	0.001±0.0001
		3	27.92±19.05	0.001±0.0001
		4	21.07±14.34	0.001±0.0001
		5	16.66±10.82	0.001±0.0001
		6	15.07±9.55	0.001±0.0008
		7	13.51±8.18	0.002±0.0008
		8	12.01±7.47	0.002±0.0008

We investigated the performance of the analyzed algorithms for superpositions of two APs with different noise levels (Fig. 10). With the noise level up to 8%, none of the analyzed methods showed decreased accuracy. It is thus essential to provide high-quality recorded signals to increase the performance of the superposition resolution algorithms. Moreover, the performance of the proposed DBC algorithm was assessed without adding the noise to the superimposed waveforms (Table IX, Id column). The average accuracy of the DBC algorithm was above 97% for the iEMG dataset (combination 2-8). Such values were higher than 93% on the neural recording dataset. We further implemented the parallel DBC algorithm, whose running time analysis is provided in Table IX. The average running time of the parallel implementation was less than 500 ms for the combinations 2-6, where at most 720 permutations must be analyzed. However, when resolving 7-8 MUs, 5040 to 40320 permutations must be analyzed, thus significantly increasing the running time. The running time of the parallel DBC algorithm was further assessed on a server (Table S2).

D. *The complexity of the superpositions of the analyzed datasets*

The superpositions considered in our study were not easy to resolve. The result of the peel-off algorithm is provided in Table X as a measure of the complexity of the superpositions. The accuracy of the peel-off algorithm ranged from 57.95 % for superpositions involving 2 APs in spike sorting set 1 to 12.01 % for superpositions involving 8 MUs in spike sorting set 2. Overall, its average accuracy was 25.39±23.16 (%). The peel-off algorithm peels off the templates in one specific order, and, as our results show, this does not always lead to the best solution. However, versions of it are still being used in the literature [55]. This might be because the algorithm is fast and works fairly well for partial superpositions [38].

IV. Discussion

A. *Evaluating the Results*

Resolving APs is a critical final step when complete decomposition is envisioned. It is time-consuming, especially when interactive decomposition programs are used [46]. Various algorithms have been proposed in the literature to resolve superimposed APs [38, 54]. However, in principle, they are not suitable for online neural decoding. Moreover, their performance significantly drops when several APs overlap. Some attempts were made to bypass this step in online decoding, including applying Cumulative Discharge Rate (CDR) and estimating approximate firing times of MUs to use incomplete decomposition results [47]. However, such an approximation is not suitable when the precise timing of each MU is required [48]. The literature also shows that resolving superimposed MUAPs can significantly increase decomposition accuracy [7, 47, 52, 56]. Thus, this study aimed to propose an accurate and efficient method to resolve superimposed MUAPs that could also be implemented in parallel (Fig. 4).

The DBC algorithm was superior to the other methods analyzed in terms of accuracy and running time. Its average running time was 33 ms and 102 ms for the combinations of five MUs in the iEMG (Tables I, II) and spike sorting (Tables III, IV) datasets. Its parallel implementation could be used, in principle, for online decoding. It is too slow, however, to be used online for superpositions involving more than 5 APs, but it may still be helpful offline, showing an average accuracy of 87.61% and 96.31% for 8 MUs (Tables VI, VII). An exhaustive-search algorithm must calculate the entire N^{N_t} residual signals, while in the DBC method $N \times (N_t)!$ residual signals must be evaluated. For example, if we limit the number of discrete-time shifts to (N=42), the number of residual signals for N_t=8 templates are 9.68E+12, and 1.69E+6 for the exhaustive search and the DBC algorithms, respectively.

For signals with sharp spikes sampled near the Nyquist rate, using interpolation to increase the time resolution is essential for increasing the accuracy of algorithms based on discrete-time shifts. We found that interpolation by a factor of four provided sufficient time resolution so that the discrete-time DBC algorithm showed high accuracy. Fine-tuning the results using the continuous-time LMA algorithm provided only a slight improvement. However, sub-sampling superposition resolution is a critical step in decreasing the clusters' dispersion, MUAP template variability for spike-triggered averaging. Such sub-sampling time shifts are important when investigating MUAP jitter, and such values affect the accuracy of neural decoding algorithms in the validation steps [36, 48]. It must also be noted that the formulation of the Gradient vector and Hessian matrix was originally proposed by McGill and Dorfman [40], and our main contribution is the development of the heuristic DBC algorithm and using three cost functions to find the best discrete-time solution. Our main message is that, unlike the peel-off algorithm, we must consider all the permutations of the templates that could provide suitable candidates for time shift estimation. Such a procedure is accurate and efficient (Tables III-VIII), and is promising for real-time neural decoding.

B. *Resolving destructive superpositions*

Destructive superimpositions are usually created when a MUAP is out of phase with another MUAP, or in a more general case, some MUAPs cancel out the others. When destructive superimpositions occur, the superimposed waveform is not similar to any of the MUAPs involved in the superimposition. The correlation of the involved MUAPs and the superimposed waveform is not thus a suitable method to resolve destructive superpositions. Resolving superimposed MUAPs can be formulated as the following, in a more general case:

$$-x_i(t+\varphi_i) \approx -s(t) + \sum\nolimits_{j=1,\ldots,N_t;\, j\neq i} x_j\left(t+\varphi_j\right) \quad (16)$$

where x_i is the ith MUAP, s is the superimposed waveform, and φ_i is the time shift of the x_i. In fact, we try to fit the right-hand side of the eq. (14) to each MUAP (i.e., i=1,..,N_t). The case with the best residual norm is considered the solution. The DBC algorithm was thus used for the implementation. Over the entire multiunit sets and the combinations of two to five MUs, this extension significantly improved the average accuracy of the superposition resolution by 2% (P-value<0.001). Since the DBC method must run N_t times, this extension is suitable for offline algorithms.

C. *Applications in Other Fields*

The problem of resolving superimpositions (overlapping waveforms) also arises in other areas, such as nuclear magnetic resonance spectroscopy [57], the overlapping of asphaltenes in the oil industry [58], and the spectral overlapping of phosphors in light-emitting diodes [59]. The algorithms proposed here could be used in these applications.

D. *Limitation and Future Works*

The present study has some limitations. The cost functions used in our study are not entirely robust to the background noise (Fig. 9). The background noise could resemble the background activity, and the performance of the proposed algorithms drops in signals with high background activities. Other cost functions should be examined in the future. The DBC algorithm is a suboptimal heuristic method. Although it was converged to the global minimum in 79% of the entire cases (i.e., the percentage of the cases with the I_d of 100%), the global convergence is not guaranteed. Using hidden convexity in nonconvex quadratic optimization is thus a focus of our future activities.

The firing time estimation method used in our study and the a-posterior accuracy assessment algorithm work on isometric constant-force contractions. Signals recorded during ramp contractions must also be analyzed, and other firing time estimation methods with more flexibility [60, 61] must be used. Also, low-to-moderate contraction levels were used in our study. Various experimental signals and higher contraction levels must be used for extensive validation, which will be a focus of future work.

Having compared the running time of the sequential DBC algorithm and its parallel implementation (Table IX), the parallel implementation is more efficient when more than five MUAPs are involved in superpositions. Due to the management between CPU cores, such an implementation is inefficient with fewer MUAPs.

V. Conclusion

In conclusion, new algorithms were proposed to resolve superimposed MUAPs for iEMG decomposition and neural spike sorting. Various simulated and synthetic signals were used to assess the accuracy and efficiency of the algorithms. Overall, the parallel DBC algorithm was superior to the other proposed methods (i.e., HRBC and FHRBC). It had comparable accuracy in comparison with the state-of-the-art (BB, [38]). However, its running time was significantly better than that of the BB method. The DBC algorithm is a simple and accurate method and could be used, in principle, for real-time neural decoding.

References

[1] D. Ge, and D. Farina, "Spike Sorting," *Introduction to Neural Engineering for Motor Rehabilitation*, pp. 155-172, 2013.

[2] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network,* vol. 9, no. 4, pp. R53-78, Nov, 1998.

[3] J. E. Chung et al. "A Fully Automated Approach to Spike Sorting," *Neuron,* vol. 95, no. 6, pp. 1381-1394.e6, 2017/09/13/, 2017.

[4] D. Carlson, and L. Carin, "Continuing progress of spike sorting in the era of big data," *Current Opinion in Neurobiology,* vol. 55, pp. 90-96, 2019/04/01/, 2019.

[5] C. Rossant et al. "Spike sorting for large, dense electrode arrays," *Nature Neuroscience,* vol. 19, no. 4, pp. 634-641, 2016/04/01, 2016.

[6] J. Sukiban et al. "Evaluation of Spike Sorting Algorithms: Application to Human Subthalamic Nucleus Recordings and Simulations," *Neuroscience,* vol. 414, pp. 168-185, 2019/08/21/, 2019.

[7] T. Yu et al. "Online Recursive Decomposition of Intramuscular EMG Signals Using GPU-Implemented Bayesian Filtering," *IEEE Transactions on Biomedical Engineering,* vol. 67, no. 6, pp. 1806-1818, 2020.

[8] M. R. Mohebian et al. "Non-invasive Decoding of the Motoneurons: A Guided Source Separation Method Based on Convolution Kernel Compensation With Clustered Initial Points," *Frontiers in Computational Neuroscience,* vol. 13, no. 14, 2019-April-02, 2019.

[9] M. Zamani et al."An Adaptive Neural Spike Processor With Embedded Active Learning for Improved Unsupervised Sorting Accuracy," *IEEE Transactions on Biomedical Circuits and Systems,* vol. 12, no. 3, pp. 665-676, 2018.

[10] H. G. Rey et al. "Past, present and future of spike sorting techniques," *Brain Research Bulletin,* vol. 119, pp. 106-117, 2015/10/01/, 2015.

[11] S. Takahashi et al."Automatic Sorting for Multi-Neuronal Activity Recorded With Tetrodes in the Presence of Overlapping Spikes," *Journal of Neurophysiology,* vol. 89, no. 4, pp. 2245-2258, 2003.

[12] P. Yger et al."A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo," *eLife,* vol. 7, pp. e34518, 2018/03/20, 2018.

[13] M. Pachitariu et al. "Fast and accurate spike sorting of high-channel count probes with KiloSort."

[14] M. Pachitariu et al. "Kilosort: real-time spike-sorting for extracellular electrophysiology with hundreds of channels," *bioRxiv,* pp. 061481, 2016.

[15] F. Franke et al. "Bayes optimal template matching for spike sorting – combining fisher discriminant analysis with optimal filtering," *Journal of Computational Neuroscience,* vol. 38, no. 3, pp. 439-459, 2015/06/01, 2015.

[16] C. Ekanadham et al ."A unified framework and method for automatic neural spike identification," *Journal of neuroscience methods,* vol. 222, pp. 47-55, 2014.

[17] J. W. Pillow et al. "A Model-Based Spike Sorting Algorithm for Removing Correlation Artifacts in Multi-Neuron Recordings," *PLOS ONE,* vol. 8, no. 5, pp. e62123, 2013.

[18] V. Ventura, and R. C. Gerkin, "Accurately estimating neuronal correlation requires a new spike-sorting paradigm," *Proceedings of the National Academy of Sciences,* vol. 109, no. 19, pp. 7230-7235, 2012.

[19] A. Oliynyk et al. "Automatic online spike sorting with singular value decomposition and fuzzy C-mean clustering," *BMC neuroscience,* vol. 13, pp. 96-96, 2012.

[20] J. S. Prentice et al. "Fast, Scalable, Bayesian Spike Identification for Multi-Electrode Arrays," *PLOS ONE,* vol. 6, no. 7, pp. e19884, 2011.

[21] F. Franke et al. "An online spike detection and spike classification algorithm capable of instantaneous resolution of overlapping spikes," *Journal of Computational Neuroscience,* vol. 29, no. 1, pp. 127-148, 2010/08/01, 2010.

[22] D. A. Adamos et al. "NASS: An empirical approach to spike sorting with overlap resolution based on a hybrid noise-assisted methodology," *Journal of Neuroscience Methods,* vol. 190, no. 1, pp. 129-142, 2010/06/30/, 2010.

[23] W. Guang-Li et al. "A robust method for spike sorting with automatic overlap decomposition," *IEEE Transactions on Biomedical Engineering,* vol. 53, no. 6, pp. 1195-1198, 2006.

[24] P.-M. Zhang et al. "Spike sorting based on automatic template reconstruction with a partial solution to the overlapping problem," *Journal of neuroscience methods,* vol. 135, no. 1-2, pp. 55-65, 2004/05//, 2004.

[25] E. Hulata et al. "A method for spike sorting and detection based on wavelet packets and Shannon's mutual information," *Journal of Neuroscience Methods,* vol. 117, no. 1, pp. 1-12, 2002/05/30/, 2002.

[26] M. S. Lewicki, "Bayesian Modeling and Classification of Neural Signals," *Neural Computation,* vol. 6, no. 5, pp. 1005-1030, 1994.

[27] D. Stashuk, "EMG signal decomposition: how can it be accomplished and used?," *Journal of Electromyography and Kinesiology,* vol. 11, no. 3, pp. 151-173, 2001/06/01/, 2001.

[28] Z. C. Lateva et al. "The innervation and organization of motor units in a series-fibered human muscle: the brachioradialis," *Journal of Applied Physiology,* vol. 108, no. 6, pp. 1530-1541, 2010.

[29] R. Merletti, and D. Farina, "Analysis of intramuscular electromyogram signals," *Philos Trans A Math Phys Eng Sci,* vol. 367, no. 1887, pp. 357-68, Jan 28, 2009.

[30] D. Farina, and A. Holobar, "Human-Machine Interfacing by Decoding the Surface Electromyogram [Life Sciences]," *IEEE Signal Processing Magazine,* vol. 32, no. 1, pp. 115-120, 2015.

[31] H. Etawil, and D. Stashuk, "Resolving superimposed motor unit action potentials," *Med Biol Eng Comput,* vol. 34, no. 1, pp. 33-40, Jan, 1996.

[32] C. I. Christodoulou, and C. S. Pattichis, "Unsupervised pattern recognition for the classification of EMG signals," *IEEE Transactions on Biomedical Engineering,* vol. 46, no. 2, pp. 169-178, 1999.

[33] J. Fang et al. “Decomposition of multiunit electromyographic signals,” *IEEE Transactions on Biomedical Engineering,* vol. 46, no. 6, pp. 685-697, 1999.

[34] C. J. De Luca et al. “Control scheme governing concurrently active human motor units during voluntary contractions,” *J Physiol,* vol. 329, pp. 129-42, Aug, 1982.

[35] H. R. Marateb, and K. C. McGill, “Resolving superimposed MUAPs using particle swarm optimization,” *IEEE Trans Biomed Eng,* vol. 56, no. 3, pp. 916-9, Mar, 2009.

[36] J. R. Florestal et al. “A genetic algorithm for the resolution of superimposed motor unit action potentials,” *IEEE Trans Biomed Eng,* vol. 54, no. 12, pp. 2163-71, Dec, 2007.

[37] K. C. McGill, “Optimal resolution of superimposed action potentials,” *IEEE Trans Biomed Eng,* vol. 49, no. 7, pp. 640-50, Jul, 2002.

[38] M. Shirzadi et al. “Rigorous performance assessment of the algorithms for resolving motor unit action potential superpositions,” *Journal of Electromyography and Kinesiology,* vol. 56, pp. 102510, 2021/02/01/, 2021.

[39] J. J. Moré, “The Levenberg-Marquardt algorithm: Implementation and theory.” pp. 105-116.

[40] K. C. McGill, and L. J. Dorfman, “High-resolution alignment of sampled waveforms,” *IEEE Trans Biomed Eng,* vol. 31, no. 6, pp. 462-8, Jun, 1984.

[41] J. J. Moré, “The Levenberg-Marquardt algorithm: Implementation and theory,” *Numerical Analysis.* pp. 105-116.

[42] K. Levenberg, “A method for the solution of certain non-linear problems in least squares,” *Quarterly of Applied Mathematics,* vol. 2, no. 2, pp. 164-168, 1944.

[43] D. W. Marquardt, “An Algorithm for Least-Squares Estimation of Nonlinear Parameters,” *Journal of the Society for Industrial and Applied Mathematics,* vol. 11, no. 2, pp. 431-441, 1963.

[44] A. Phinyomark et al. “Feature reduction and selection for EMG signal classification,” *Expert Systems with Applications,* vol. 39, no. 8, pp. 7420-7431, 2012/06/15/, 2012.

[45] C. Pedreira et al. “How many neurons can we see with current spike sorting algorithms?,” *Journal of neuroscience methods,* vol. 211, no. 1, pp. 58-65, 2012.

[46] K. C. McGill et al. “EMGLAB: An interactive EMG decomposition program,” *Journal of Neuroscience Methods,* vol. 149, no. 2, pp. 121-133, 2005/12/15/, 2005.

[47] S. Karimimehr et al. “A Real-Time Method for Decoding the Neural Drive to Muscles Using Single-Channel Intra-Muscular EMG Recordings,” *International Journal of Neural Systems,* vol. 27, no. 06, pp. 1750025, 2017.

[48] H. R. Marateb, and K. C. McGill, “Electromyographic (EMG) Decomposition,” *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 1-15, 2016.

[49] K. C. McGill, and H. R. Marateb, “Rigorous a posteriori assessment of accuracy in EMG decomposition,” *IEEE Trans Neural Syst Rehabil Eng,* vol. 19, no. 1, pp. 54-63, Feb, 2011.

[50] J. C. Kline, and C. J. D. Luca, “Error reduction in EMG signal decomposition,” *Journal of Neurophysiology,* vol. 112, no. 11, pp. 2718-2728, 2014.

[51] R. Breitling et al. “Rank products: a simple, yet powerful, new method to detect differentially regulated genes in replicated microarray experiments,” *FEBS Letters,* vol. 573, no. 1, pp. 83-92, 2004/08/27/, 2004.

[52] T. Yu et al. “Recursive Decomposition of Electromyographic Signals With a Varying Number of Active Sources: Bayesian Modeling and Filtering,” *IEEE Transactions on Biomedical Engineering,* vol. 67, no. 2, pp. 428-440, 2020.

[53] J. W. Hardin, and J. M. Hilbe, *Generalized estimating equations*: Chapman & Hall/CRC Boca Raton, FL, 2003.

[54] D. Ge et al. “Spike sorting by stochastic simulation,” *IEEE transactions on neural systems and rehabilitation engineering,* vol. 19, no. 3, pp. 249-259, 2011.

[55] X. Ren et al. “Intramuscular EMG Decomposition Basing on Motor Unit Action Potentials Detection and Superposition Resolution,” *Frontiers in Neurology,* vol. 9, no. 2, 2018-January-23, 2018.

[56] J. Monsifrot et al. “Sequential Decoding of Intramuscular EMG Signals via Estimation of a Markov Model,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 22, no. 5, pp. 1030-1040, 2014.

[57] S. Ye et al. “Localised high resolution spectral estimator for resolving superimposed peaks in NMR signals,” *Signal Processing,* vol. 130, pp. 343-354, 2017/01/01/, 2017.

[58] M. Asemani, and A. R. Rabbani, “Detailed FTIR spectroscopy characterization of crude oil extracted asphaltenes: Curve resolve of overlapping bands,” *Journal of Petroleum Science and Engineering,* vol. 185, pp. 106618, 2020/02/01/, 2020.

[59] J. S. Lee et al. “Smart design to resolve spectral overlapping of phosphor-in-glass for high-powered remote-type white light-emitting devices,” *Optics Letters,* vol. 39, no. 4, pp. 762-765, 2014/02/15, 2014.

[60] J. Navallas et al. “Exact inter-discharge interval distribution of motor unit firing patterns with gamma model,” *Medical & biological engineering & computing,* vol. 57, no. 5, pp. 1159-1171, 2019.

[61] J. Navallas et al. “Maximum likelihood estimation of motor unit firing pattern statistics,” *IEEE Trans Neural Syst Rehabil Eng,* vol. 22, no. 3, pp. 460-9, May, 2014.