



OPEN

A novel method for in-situ extracting bio-impedance model parameters optimized for embedded hardware

Mitar Simić¹✉, Todd J. Freeborn², Tomislav B. Šekara³, Adrian K. Stavrakis¹, Varun Jeoti¹ & Goran M. Stojanović¹

A novel method for embedded hardware-based parameter estimation of the Cole model of bioimpedance is developed and presented. The model parameters R_{∞} , R_1 and C are estimated using the derived set of equations based on measured values of real (R) and imaginary part (X) of bioimpedance, as well as the numerical approximation of the first derivative of quotient R/X with respect to angular frequency. The optimal value for parameter α is estimated using a brute force method. The estimation accuracy of the proposed method is very similar with the relevant work from the existing literature. Moreover, performance evaluation was performed using the MATLAB software installed on a laptop, as well as on the three embedded-hardware platforms (Arduino Mega2560, Raspberry Pi Pico and XIAO SAMD21). Obtained results showed that the used platforms can perform reliable bioimpedance processing with the same accuracy, while Raspberry Pi Pico is the fastest solution with the smallest energy consumption.

Impedance spectroscopy is a non-invasive method used in various fields, such as biology, medicine and material science¹ to characterize the passive electrical impedance of a material or sample. It is widely applied to characterize biological tissues, where the measured impedance is often referred to as tissue bioimpedance². Recently tissue bioimpedance has been utilized for respiratory monitoring during inspiratory loading, which enabled development a wearable device for detection of normal and restrictive breathing³. Further, combining bioimpedance with electromyography measurements have been investigated to increase the reliability of detecting muscle contraction with potential to support prosthetics and human–computer interfaces⁴. Localized bioimpedance measurements have also been utilized to monitor exercise induced changes of bicep tissues⁵. In addition to the above-mentioned applications, there are reported studies using tissue bioimpedance to non-invasively assess cardiac and respiratory activity⁶. Bioimpedance techniques have also been reported to detect of cell morphological changes in studies of ischemic rabbit liver tissue⁷ and blood glucose monitoring⁸. In recent years, significant efforts have been made to promote a personalized and patient-centric healthcare paradigm. The future healthcare will include wearable devices that will enable continuous monitoring of various parameters outside the clinics, including bioimpedance with real-time online data access⁹. Recent examples highlighting this type of system the wearable devices for knee joint health monitoring reported by Hersk et al.¹⁰ and Critcher and Freeborn¹¹. A body composition analysis based on bioimpedance processing is important for monitoring of various chronic diseases¹². Moreover, portable human skin surface stimulators are found to be safer if bioimpedance feedback is included to reduce the possibility of burns and injuries¹³. Further, bioimpedance based techniques are also being investigated as a sensing modality for implantable devices¹⁴, which opens up new health-focused applications that may not be possible using portable or wearable devices.

The traditional approaches for processing bioimpedance data can be broadly sorted into two categories: discrete processing and equivalent circuit modeling. For discrete processing, bioimpedance data (e.g. modulus/phase or real/imaginary data) collected at one or more discrete frequencies are investigated for their relationship to various health statuses of the analyzed tissue. Such an approach was used for monitoring of dialysis patients during ultrafiltration¹⁵, or for detection of intraneural needle placement¹⁶. The second approach uses equivalent

¹Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia. ²Department of Electrical and Computer Engineering, The University of Alabama, Box 870286, Tuscaloosa, AL 35487, USA. ³School of Electrical Engineering, University of Belgrade, Bulevar Kralja Aleksandra 73, 11120 Belgrade, Serbia. ✉email: mitar.simic@uns.ac.rs

electrical circuits, composed of electrical elements (such as resistors, capacitors and inductors), to represent multi-frequency bioimpedance data and then investigates the relationship of the model/component values to underlying physical and electrochemical changes in the tissue. An equivalent electrical circuit using three circuit components based on the Cole-impedance equation is widely used for various clinical applications¹⁷, such as body composition analysis¹⁸. While there are numerous studies that present the portable and wearable bioimpedance measurement devices, there are few works focused on the development of the bioimpedance processing techniques suitable for such devices¹⁹. A possible reason for this limited research may be that the Cole-impedance model (presented later in section "Proposed estimation method") requires the use of a fractional-order differential equation^{20,21}. Processing techniques with the Cole-impedance model have utilized complex optimizing algorithms for parameter estimation, such as non-linear least squares (NLLS)²², or meta-heuristic optimization algorithms²³. A limitation of these techniques is that the accuracy, execution time and convergence are dependent on the quality of the initial values provided by the user.

Therefore, the development of bioimpedance processing techniques for parameter estimation of the Cole-impedance model that are suitable for deployment on portable embedded-hardware based devices is still an emerging topic. The aims of this paper are to formulate and evaluate a novel algorithm that is suitable for deployment on portable embedded-hardware and does not require initial-values as inputs from the user.

The rest of this work is organized as follows: The proposed estimation method is presented in section "methods and materials", as well as the characterization procedure and used hardware/software. The main experimental results are given in section "Results and discussion" with a specific focus on comparing the proposed algorithm with recent methods/hardware¹⁹. Further, a performance comparison of the proposed algorithm using three microcontroller-based platforms (Arduino Mega2560, Raspberry Pi Pico and XIAO SAMD21) is presented. The overall conclusions are given in section "Conclusion" with discussion about the achieved results and the practical importance of the presented method and embedded-hardware platforms.

Methods and materials

Proposed estimation method. The Cole-impedance model is composed of three circuit elements (two resistors and a constant phase element) with impedance represented by four parameters (R_∞ , R_1 , C and α) given by Eq. (1):

$$\underline{Z}(\omega_i) = R(\omega_i) + jX(\omega_i) = R_\infty + \frac{R_1}{1 + R_1 C (j\omega_i)^\alpha} \quad (1)$$

where the real part of impedance, resistance, is labelled as $R(\omega_i)$, while the imaginary part, or reactance, is labelled as $X(\omega_i)$. The constant phase element is represented by two parameters (C and α) with voltage/current characteristics modeled by a fractional-order differential equation (of order α). Note that when tissue bioimpedance is measured across a limited frequency band with a fixed number (N) of datapoints, the frequency (ω) can be represented as a vector (ω_i) where i is an integer ranging from 1 to N . To simplify the presentation of the development of our method, the Cole-impedance model (Eq. (1)) is reduced to the compact form given by:

$$\underline{Z}(\omega_i) = R_\infty + \frac{R_1}{1 + R_1 C (j\omega_i)^\alpha} = \frac{a_1 + a_2 (j\omega_i)^\alpha}{1 + b_1 (j\omega_i)^\alpha} \quad (2)$$

where a_1 , a_2 and b_1 are coefficients with values:

$$a_1 = R_\infty + R_1 \quad (3)$$

$$a_2 = R_\infty R_1 C \quad (4)$$

$$b_1 = R_1 C \quad (5)$$

Our estimation method uses values of $R(\omega_i)$ and $X(\omega_i)$ obtained from the decomposition of $\underline{Z}(\omega_i)$ into its real and imaginary parts. All further equations assume positive values of $X(\omega_i)$, which is achieved by multiplication of the actual $X(\omega_i)$ with -1 prior to estimation. Both $R(\omega_i)$ and $X(\omega_i)$ are given with Eq. (6) and Eq. (7):

$$R(\omega_i) = \frac{\omega_i^\alpha (a_1 b_1 + a_2) \cos\left(\frac{\alpha\pi}{2}\right) + \omega_i^{2\alpha} a_2 b_1 + a_1}{1 + \omega_i^{2\alpha} b_1^2 + 2b_1 \omega_i^\alpha \cos\left(\frac{\alpha\pi}{2}\right)} \quad (6)$$

$$X(\omega_i) = \frac{\omega_i^\alpha \sin\left(\frac{\alpha\pi}{2}\right) (a_1 b_1 - a_2)}{1 + \omega_i^{2\alpha} b_1^2 + 2b_1 \omega_i^\alpha \cos\left(\frac{\alpha\pi}{2}\right)} \quad (7)$$

As there are four parameters to the Cole-impedance model (a_1 , a_2 , b_1 and α) that are also represented in (6) and (7), an analytical solution for parameter estimation requires two additional equations. Our third equation is the first derivative with respect to ω of quotient $R(\omega)/X(\omega)$ given by:

$$G'(\omega) = \frac{\partial}{\partial \omega} \{G(\omega)\} = \frac{\partial}{\partial \omega} \left\{ \frac{R(\omega)}{X(\omega)} \right\} = \frac{\alpha (a_1 - a_2 b_1 \omega_i^{2\alpha})}{(a_2 - a_1 b_1) \omega_i^{\alpha+1} \sin \frac{\alpha\pi}{2}} \quad (8)$$

While (8) is not directly measurable, on the discrete set of ω_i it can be approximated using $R(\omega_i)$ and $X(\omega_i)$ by a numerical procedure (central differentiation method, for example):

$$G'(\omega_i) = \frac{G(\omega_{i+1}) - G(\omega_{i-1})}{\omega_{i+1} - \omega_{i-1}} = \frac{\frac{R(\omega_{i+1})}{X(\omega_{i+1})} - \frac{R(\omega_{i-1})}{X(\omega_{i-1})}}{\omega_{i+1} - \omega_{i-1}} \tag{9}$$

In the Cole-impedance model, given by (1), the parameter α is limited to the range $[0,1]$ and parameters R_∞ , R_1 , and C are limited to positive real values. Therefore, a_1 , a_2 and b_1 will be positive as well. Our estimation method solves the set of Eqs. (6)–(8) to extract parameters a_1 , a_2 , and b_1 as a function of α , while the exact value of α is determined by a brute-force procedure to minimize the value of the defined error function.

To reduce equation length, the notation of R_i , X_i and G_i for $R(\omega_i)$, $X(\omega_i)$ and $G(\omega_i)$, respectively, was used. From (6)–(8) the values of a_1 , a_2 , and b_1 are determined at each ω_i as follows:

$$a_1(\omega_i) = - \left(X_i \cdot \left(\omega_i^{(2-\alpha)} \cdot \sin((\pi \cdot \alpha)/2)^2 \cdot (R_i^2 \cdot \alpha^2 + X_i^2 \cdot \alpha^2 - G_i^2 \cdot X_i^2 \cdot \omega_i^2 \cdot \sin((\pi \cdot \alpha)/2)^2) \right)^{(1/2)} \right. \\ \left. + R_i^2 \cdot \alpha \cdot \omega_i^\alpha \cdot \sin((\pi \cdot \alpha)/2)^2 + X_i^2 \cdot \alpha \cdot \omega_i^\alpha \cdot \sin((\pi \cdot \alpha)/2)^2 - G_i \cdot R_i \cdot X_i \cdot \omega_i \cdot \omega_i^\alpha \cdot \sin((\pi \cdot \alpha)/2)^2 \right) / \\ \left(\omega_i^\alpha \cdot \sin((\pi \cdot \alpha)/2) \cdot (X_i \cdot \alpha \cdot \cos((\pi \cdot \alpha)/2) - R_i \cdot \alpha \cdot \sin((\pi \cdot \alpha)/2) + G_i \cdot X_i \cdot \omega_i \cdot \sin((\pi \cdot \alpha)/2)) \right) \tag{10}$$

$$a_2(\omega_i) = - \left(R_i \cdot \sin((\pi \cdot \alpha)/2) \cdot \left(\omega_i^{(2-\alpha)} \cdot \sin((\pi \cdot \alpha)/2)^2 \cdot (R_i^2 \cdot \alpha^2 + X_i^2 \cdot \alpha^2 - G_i^2 \cdot X_i^2 \cdot \omega_i^2 \cdot \sin((\pi \cdot \alpha)/2)^2) \right)^{(1/2)} - \right. \\ \left(-X_i \cdot \cos((\pi \cdot \alpha)/2) \cdot \left(\omega_i^{(2-\alpha)} \cdot \sin((\pi \cdot \alpha)/2)^2 \cdot (R_i^2 \cdot \alpha^2 + X_i^2 \cdot \alpha^2 - G_i^2 \cdot X_i^2 \cdot \omega_i^2 \cdot \sin((\pi \cdot \alpha)/2)^2) \right)^{(1/2)} \right. \\ \left. + G_i \cdot X_i^2 \cdot \omega_i^{(\alpha+1)} \cdot \sin((\pi \cdot \alpha)/2)^3 + G_i \cdot R_i \cdot X_i \cdot \omega_i^{(\alpha+1)} \cdot \cos((\pi \cdot \alpha)/2) - G_i \cdot R_i \cdot X_i \cdot \omega_i^{(\alpha+1)} \cdot \cos((\pi \cdot \alpha)/2)^3 \right) \\ \left. / \left(\omega_i^{(2-\alpha)} \cdot (X_i \cdot \alpha \cdot \cos((\pi \cdot \alpha)/2) \cdot \sin((\pi \cdot \alpha)/2) - R_i \cdot \alpha \cdot \sin((\pi \cdot \alpha)/2)^2 + G_i \cdot X_i \cdot \omega_i \cdot \sin((\pi \cdot \alpha)/2)^2) \right) \tag{11}$$

$$b_1(\omega_i) = - \left(\left(\omega_i^{(2-\alpha)} \cdot \sin((\pi \cdot \alpha)/2)^2 \cdot (R_i^2 \cdot \alpha^2 + X_i^2 \cdot \alpha^2 - G_i^2 \cdot X_i^2 \cdot \omega_i^2 \cdot \sin((\pi \cdot \alpha)/2)^2) \right)^{(1/2)} \right. \\ \left. + X_i \cdot \alpha \cdot \omega_i^\alpha + G_i \cdot X_i \cdot \omega_i \cdot \omega_i^\alpha \cdot \cos((\pi \cdot \alpha)/2) \cdot \sin((\pi \cdot \alpha)/2) \right) / \\ \left(\omega_i^{(2-\alpha)} \cdot (X_i \cdot \alpha \cdot \cos((\pi \cdot \alpha)/2) - R_i \cdot \alpha \cdot \sin((\pi \cdot \alpha)/2) + G_i \cdot X_i \cdot \omega_i \cdot \sin((\pi \cdot \alpha)/2)) \right) \tag{12}$$

The obtained arrays of $a_1(\omega_i)$, $a_2(\omega_i)$, and $b_2(\omega_i)$ are averaged to calculate unique values (\hat{a}_1 , \hat{a}_2 and \hat{b}_1) for a fixed value of α , which is followed by a calculation of the error function value. Next, the value of α is increased by a defined step (a step of 0.01 is used here because it provides 1% resolution) and the procedure to solve (6)–(8) with averaging is repeated.

The final value of α from this estimation process is selected as that specific value that minimizes the error function. In this paper we will analyze two error functions for this step. The first, referred to in this work as Method 1, estimates the real ($R_{est}(\omega_i)$) and imaginary ($X_{est}(\omega_i)$) impedance components using the current α and estimated \hat{a}_1 , \hat{a}_2 and \hat{b}_1 from (10)–(12) plugged into (6) and (7). Final value of α is selected such that sum of mean relative error between the estimated and measured real and imaginary part is minimal:

$$\operatorname{argmin}_\alpha \left\{ \frac{1}{N} \sum_{i=1}^N \left| \frac{R_{est}(\alpha, \omega_i) - R(\omega_i)}{R(\omega_i)} \right| + \frac{1}{N} \sum_{i=1}^N \left| \frac{X_{est}(\alpha, \omega_i) - X(\omega_i)}{X(\omega_i)} \right| \right\} \tag{13}$$

Our second approach, referred to in this work as Method 2, selects α such that the mean absolute error between measured and estimated impedance modulus is minimal:

$$\operatorname{argmin}_\alpha \left\{ \frac{1}{N} \sum_{i=1}^N \left| \frac{\sqrt{R_{est}^2(\alpha, \omega_i) + X_{est}^2(\alpha, \omega_i)} - \sqrt{R^2(\omega_i) + X^2(\omega_i)}}{\sqrt{R^2(\omega_i) + X^2(\omega_i)}} \right| \right\} \tag{14}$$

Finally, the estimated values of model parameters (\hat{R}_∞ , \hat{R}_1 and \hat{C}) are calculated using \hat{a}_1 , \hat{a}_2 and \hat{b}_1 , and solving the set of Eqs. (3)–(5) which gives:

$$\hat{R}_\infty = \hat{a}_2 / \hat{b}_1 \tag{15}$$

$$\hat{R}_1 = (\hat{a}_1 \hat{b}_1 - \hat{a}_2) / \hat{b}_1 \tag{16}$$

$$\hat{C} = \hat{b}_1^2 / (\hat{a}_1 \hat{b}_1 - \hat{a}_2) \tag{17}$$

For better distinction from the reference values (R_∞ , R_1 , C and α) the estimated values of model parameters are labeled as (\hat{R}_∞ , \hat{R}_1 , \hat{C} and $\hat{\alpha}$).

Hardware platforms for estimation validation. The proposed estimation method was developed and initially tested on a Dell G5 laptop (Core i7 9th generation) with 2.6 GHz clock speed and 16 GB of RAM. After development, the algorithm was ported to three embedded hardware platforms (Arduino Mega2560, Raspberry Pi Pico (RPI Pico) and XIAO SAMD21), shown in Fig. 1, for evaluation of parameter estimation performance.

The Arduino Mega2560 is an embedded platform that utilizes the AVR ATmega2560 8-bit microcontroller. An important feature of the ATmega2560 is the majority of instructions are executed in a single clock cycle, which can provide up to 16 Million Instructions Per Second (MIPS) at 16 MHz²⁴. However, the ATmega2560 does not have built-in hardware support for floating point numbers (so floating-point operations require conversion to integer-order operations). There is 256 kB of on-board flash memory, 8 kB of RAM, and 4 kB of EEPROM storage. Overall dimensions (width × length) are: 101.52 mm × 53.3 mm with an approximate weight of 37 g. The approximate retail price of the Arduino Mega2560 platform is 48.95 USD²⁵.

The Raspberry Pi Pico is an embedded platform that utilizes dual ARM Cortex-M0 + RP2040 microcontrollers which support clock speeds up to 133 MHz (with a default clock configuration of 125 MHz). The on-chip SRAM is organized into six independent banks allowing simultaneous parallel access. The Raspberry Pi Pico does not support floating point arithmetic in hardware, but it has an interpolator and integer divider peripherals with possible library optimizations. A limitation of the RP2040 is the low execution speed if the program code does not fit into the 16 kB cache. This is noticeable at application startup when nothing is cached. With the RP2040's default settings fetching a 32-bit value requires 80 SPI clock cycles (but can be improved to 3.3 MIPS at 133 MHz²⁶). In terms of storage, 2 MB of on-board flash and 256 kB of RAM are available. EEPROM is not included. The width and length are 51 mm and 21 mm, respectively, with an approximate weight of 9 g. The approximate retail price of the Raspberry Pi Pico is 4 USD²⁷.

The XIAO SAMD21 is an embedded platform that utilizes the SAM D21 family of 32-bit Cortex-M0 + ARM microcontrollers. The default clock speed of XIAO SAMD21 board is 8 MHz, but can be configured to 48 MHz with the implementation of the digital frequency-locked loop. The XIAO SAMD21 does not support floating point arithmetic in hardware²⁸. In terms of storage 256 kB of flash and 32 kB of RAM are available. Similar to the Raspberry Pi Pico an EEPROM is not included for XIAO SAMD21. The width and length are 23.5 mm and 17.5 mm, respectively, with an approximate weight of 9 g. The approximate retail price of the XIAO SAMD21 is 5.40 USD²⁹.

Comparing the specifications of the three embedded platforms, the Raspberry Pico board has the lowest price (as of accessed date from their online store), fastest clock speed, and highest amount of RAM and flash memory. However, ArduinoMega2560 board offers a greater number of available analog-to-digital (ADC) channels (but with lower resolution than RPi Pico and XIAO SAMD21) and the highest number of general-purpose input/output (GPIO) pins. Moreover, the Arduino Mega2560 has built in Electrically Erasable Programmable Read-Only Memory (EEPROM) which is useful for storing coefficients and look-up tables available after reset or powering off the device. In terms of size, the smallest area is occupied by the XIAO SAMD21 which can be very important for the wearable applications.

Software environments. For developing and testing the estimation algorithm on the laptop computer, the MATLAB R2013b programming/computing platform was used. No additional toolboxes or toolbox functions were utilized in the development of the algorithm to support porting of the algorithm to other software tools and programming environments.

For the testing of the algorithm on the embedded hardware, the Arduino IDE 1.8.5 programming platform was used. The algorithm code for the embedded hardware used the *math* library to access the mathematical functions for manipulating floating-point numbers necessary for the algorithm. The same source code was compiled for each of the embedded platforms without any modifications.

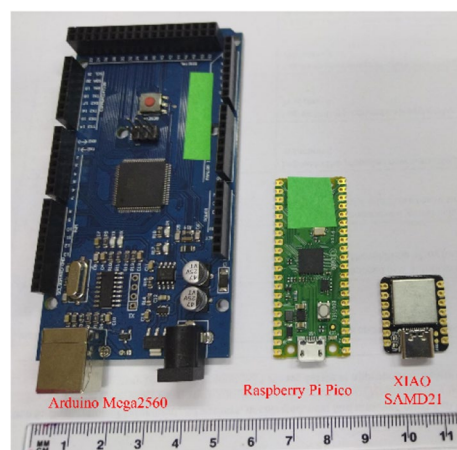


Figure 1. Used embedded-hardware platforms: Arduino Mega2560, Raspberry Pi Pico and XIAO SAMD21.

Synthetic datasets for performance evaluation. For the algorithm performance evaluation in terms of accuracy and execution time multiple impedance datasets were generated. Ten synthetic datasets were created using (6) and (7) with randomly chosen reference values of model parameters within the ranges: $100 \Omega < R_\infty < 1000 \Omega$, $10 \Omega < R_l < 1000 \Omega$, $1 \mu\text{F} < C < 10 \mu\text{F}$ and $0.5 < \alpha < 1.0$). The specific ranges for parameters were chosen to represent values typical of bioimpedance applications. The total number of datapoints in each dataset was $N=256$ and datapoints were logarithmically spaced in the frequency range from 3 kHz to 1 MHz (selected to represent the range and number of datapoints generated by measurements with an ImpediMed SFB7 device, widely used in bioimpedance applications). The ideal datasets had 1% random noise added to both parts of impedance (R and X) to represent bioimpedance data more accurately from tissues. For reference, these synthetic datasets are shown in Fig. 2 as solid blue lines. It is important to note that the resistance and reactance in Fig. 2 capture a wide range of values across the generation space, which ensures that the evaluation of the proposed method reflects a realistic range and not a unique or single case.

An additional synthetic dataset was created using (6) and (7) with reference values $R_\infty = 84.40 \Omega$, $R_l = 39.20 \Omega$, $C = 2.31 \mu\text{F}$ and $\alpha = 0.747$ also with $N = 256$ datapoints, frequency range of 3 kHz–1 MHz, and 0.25% random noise level to replicate a test case from reference¹⁹ for comparison. Freeborn evaluated the performance of a

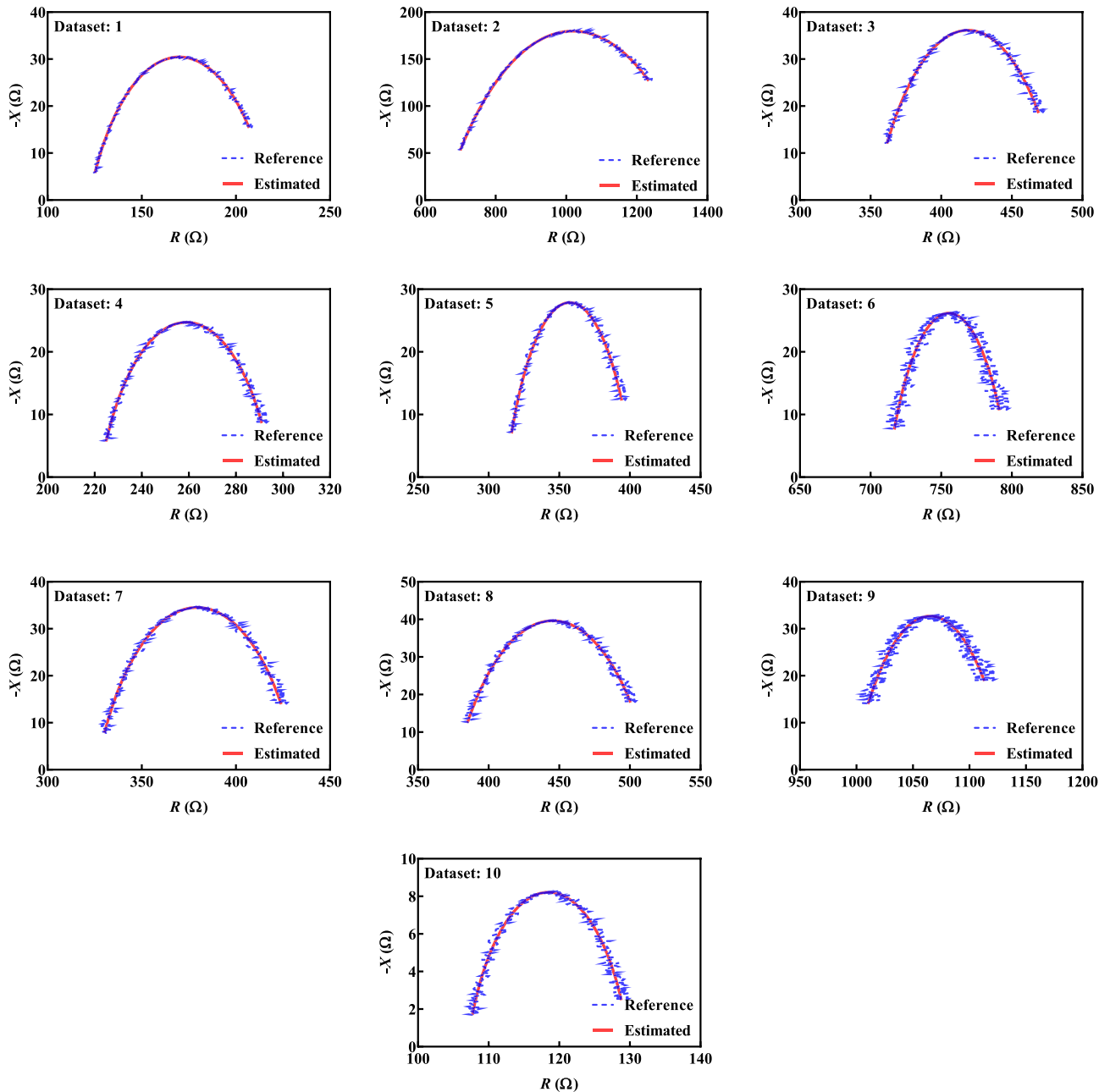


Figure 2. The Nyquist plots of reference and estimated values.

least-squares parameter estimation method on both desktop/laptop and Raspberry Pi 3 platforms¹⁹ providing values to compare the performance of the proposed algorithm against.

To also evaluate the effect that noise has on the proposed algorithm, synthetic datasets using the reference values ($R_\infty = 84.40 \Omega$, $R_l = 39.20 \Omega$, $C = 2.31 \mu\text{F}$ and $\alpha = 0.747$) were generated with: 0%, 0.25%, 1%, 5% and 10% random noise.

After that, initial dataset defined with $R_\infty = 84.40 \Omega$, $R_l = 39.20 \Omega$, $C = 2.31 \mu\text{F}$, $\alpha = 0.747$, $N = 256$ and frequency range (3 kHz–1 MHz) was divided in which lower frequency range (3 kHz–53 kHz) was used for estimation of \hat{a}_1 , while the part from 43 to 68 kHz was used for estimation of \hat{a}_2 and b_1 .

To evaluate the effect of the number of frequency points (within a fixed frequency band) on the algorithm performance, datasets using $R_\infty = 84.40 \Omega$, $R_l = 39.20 \Omega$, $C = 2.31 \mu\text{F}$ and $\alpha = 0.747$ with $N = 256, 100, 20, 10, 6$, and 3 were generated using the fixed frequency band from 3 kHz to 1 MHz.

Results and discussion

The PC-based evaluation of the estimation method. The first evaluation of the estimation algorithm compared the execution time and parameters using the two error function definitions given by (13) and (14) when implemented in MATLAB and executed on the study laptop. These error functions are referred to as “Method 1” for (13) and “Method 2” for (14). The estimated and reference values for the synthetic 10 datasets (with 0.25% noise) are given in Table 1 for both methods. The execution time (t_{exe}), as determined using the stopwatch timer embedded in MATLAB software (*tic* and *toc* functions), is also given in Table 1.

As it can be seen from Table 1, both methods estimated values with very high accuracy, however Method 1 required smaller execution times with smaller errors. Method 1 was, in average faster for 3.5%. Therefore, Method 1 is utilized for the remainder of evaluations in this work because of its better performance. The greater execution time of Method 2 is attributed to the greater number of complex calculations in the procedure, requiring 6

Dataset number		t_{exe}	Model parameters			
1	Reference		$R_\infty = 122.00 \Omega$	$R_l = 96.00 \Omega$	$C = 1.90 \mu\text{F}$	$\alpha = 0.72$
	Method 1	0.143 s	$\hat{R}_\infty = 121.98 \Omega$	$\hat{R}_l = 95.97 \Omega$	$\hat{C} = 1.90 \mu\text{F}$	$\hat{\alpha} = 0.72$
	Method 2	0.151 s	$\hat{R}_\infty = 121.98 \Omega$	$\hat{R}_l = 95.97 \Omega$	$\hat{C} = 1.90 \mu\text{F}$	$\hat{\alpha} = 0.72$
2	Reference		$R_\infty = 652.00 \Omega$	$R_l = 734.00 \Omega$	$C = 1.50 \mu\text{F}$	$\alpha = 0.58$
	Method 1	0.097 s	$\hat{R}_\infty = 651.89 \Omega$	$\hat{R}_l = 733.83 \Omega$	$\hat{C} = 1.50 \mu\text{F}$	$\hat{\alpha} = 0.58$
	Method 2	0.097 s	$\hat{R}_\infty = 651.89 \Omega$	$\hat{R}_l = 733.83 \Omega$	$\hat{C} = 1.50 \mu\text{F}$	$\hat{\alpha} = 0.58$
3	Reference		$R_\infty = 352.00 \Omega$	$R_l = 134.00 \Omega$	$C = 3.20 \mu\text{F}$	$\alpha = 0.63$
	Method 1	0.096 s	$\hat{R}_\infty = 351.94 \Omega$	$\hat{R}_l = 133.97 \Omega$	$\hat{C} = 3.20 \mu\text{F}$	$\hat{\alpha} = 0.63$
	Method 2	0.101 s	$\hat{R}_\infty = 350.15 \Omega$	$\hat{R}_l = 139.92 \Omega$	$\hat{C} = 3.55 \mu\text{F}$	$\hat{\alpha} = 0.62$
4	Reference		$R_\infty = 222.00 \Omega$	$R_l = 74.00 \Omega$	$C = 1.20 \mu\text{F}$	$\alpha = 0.75$
	Method 1	0.142 s	$\hat{R}_\infty = 221.96 \Omega$	$\hat{R}_l = 73.98 \Omega$	$\hat{C} = 1.20 \mu\text{F}$	$\hat{\alpha} = 0.75$
	Method 2	0.147 s	$\hat{R}_\infty = 221.96 \Omega$	$\hat{R}_l = 73.98 \Omega$	$\hat{C} = 1.20 \mu\text{F}$	$\hat{\alpha} = 0.75$
5	Reference		$R_\infty = 312.00 \Omega$	$R_l = 91.00 \Omega$	$C = 2.02 \mu\text{F}$	$\alpha = 0.70$
	Method 1	0.098 s	$\hat{R}_\infty = 311.94 \Omega$	$\hat{R}_l = 90.98 \Omega$	$\hat{C} = 2.02 \mu\text{F}$	$\hat{\alpha} = 0.70$
	Method 2	0.099 s	$\hat{R}_\infty = 310.67 \Omega$	$\hat{R}_l = 95.15 \Omega$	$\hat{C} = 2.24 \mu\text{F}$	$\hat{\alpha} = 0.69$
6	Reference		$R_\infty = 712.00 \Omega$	$R_l = 87.00 \Omega$	$C = 2.12 \mu\text{F}$	$\alpha = 0.69$
	Method 1	0.097 s	$\hat{R}_\infty = 711.87 \Omega$	$\hat{R}_l = 86.98 \Omega$	$\hat{C} = 2.12 \mu\text{F}$	$\hat{\alpha} = 0.69$
	Method 2	0.101 s	$\hat{R}_\infty = 710.48 \Omega$	$\hat{R}_l = 90.71 \Omega$	$\hat{C} = 2.34 \mu\text{F}$	$\hat{\alpha} = 0.68$
7	Reference		$R_\infty = 326.00 \Omega$	$R_l = 107.00 \Omega$	$C = 1.18 \mu\text{F}$	$\alpha = 0.73$
	Method 1	0.146 s	$\hat{R}_\infty = 325.99 \Omega$	$\hat{R}_l = 106.99 \Omega$	$\hat{C} = 1.18 \mu\text{F}$	$\hat{\alpha} = 0.73$
	Method 2	0.151 s	$\hat{R}_\infty = 325.94 \Omega$	$\hat{R}_l = 106.97 \Omega$	$\hat{C} = 1.18 \mu\text{F}$	$\hat{\alpha} = 0.73$
8	Reference		$R_\infty = 376.00 \Omega$	$R_l = 139.00 \Omega$	$C = 1.98 \mu\text{F}$	$\alpha = 0.66$
	Method 1	0.098 s	$\hat{R}_\infty = 375.93 \Omega$	$\hat{R}_l = 138.97 \Omega$	$\hat{C} = 1.98 \mu\text{F}$	$\hat{\alpha} = 0.66$
	Method 2	0.100 s	$\hat{R}_\infty = 375.93 \Omega$	$\hat{R}_l = 138.97 \Omega$	$\hat{C} = 1.98 \mu\text{F}$	$\hat{\alpha} = 0.66$
9	Reference		$R_\infty = 996.00 \Omega$	$R_l = 139.00 \Omega$	$C = 6.98 \mu\text{F}$	$\alpha = 0.56$
	Method 1	0.097 s	$\hat{R}_\infty = 995.82 \Omega$	$\hat{R}_l = 138.97 \Omega$	$\hat{C} = 6.98 \mu\text{F}$	$\hat{\alpha} = 0.56$
	Method 2	0.101 s	$\hat{R}_\infty = 992.08 \Omega$	$\hat{R}_l = 149.15 \Omega$	$\hat{C} = 8.58 \mu\text{F}$	$\hat{\alpha} = 0.54$
10	Reference		$R_\infty = 107.00 \Omega$	$R_l = 23.00 \Omega$	$C = 2.28 \mu\text{F}$	$\alpha = 0.79$
	Method 1	0.144 s	$\hat{R}_\infty = 106.98 \Omega$	$\hat{R}_l = 22.99 \Omega$	$\hat{C} = 2.28 \mu\text{F}$	$\hat{\alpha} = 0.79$
	Method 2	0.154 s	$\hat{R}_\infty = 106.98 \Omega$	$\hat{R}_l = 22.99 \Omega$	$\hat{C} = 2.28 \mu\text{F}$	$\hat{\alpha} = 0.79$

Table 1. Performance comparison of the proposed method in case of data with 1% of noise.

multiplications of real numbers and calculation of three square roots. The Nyquist plots of the Cole impedance using the reference and estimated values with Method 1 are shown in Fig. 2. As it can be seen from Fig. 2, there is good visual agreement between the reference and estimated values for all 10 datasets.

Comparison with reference¹⁹. The comparison of reference values and estimated values using our estimation method and¹⁹ is given in Table 2. As it can be seen from Table 2, our method estimated values with lower relative errors for all parameters except R_{∞} .

When compared to the CNLS-method¹⁹, the main advantages of the proposed method are the elimination of initial values of model parameters (to initiate the least squares search approach) and the elimination of the need for specific functions or toolboxes. For example, the initial values in¹⁹ are generated with 100 random sets followed by monitoring of the estimation accuracy and selection of the optimal initial values. This approach is time consuming, and it is not optimized for wearable devices based on embedded hardware. Moreover, CNLS-based method required specific the SciPy library for the Python implementation¹⁹.

The impact of noise in EIS data. All impedance measurement devices will have different levels of accuracy (based on their underlying design) and different levels of noise (based on both design and operating environment). Since noise will always be present in a bioimpedance dataset, the performance of the estimation algorithm in terms of parameter accuracy was evaluated using the reference dataset ($R_{\infty}=84.40\ \Omega$, $R_1=39.20\ \Omega$, $C=2.31\ \mu\text{F}$ and $\alpha=0.747$)¹⁹ with five noise levels: 0%, 0.25%, 1%, 5% and 10%. Using the proposed method with these datasets, the estimated and reference impedances are shown in Fig. 3. The estimated values for the cases with 0% and 0.25% were: $\hat{R}_{\infty}=84.62\ \Omega$, $\hat{R}_1=38.65\ \Omega$, $\hat{C}=2.25\ \mu\text{F}$ and $\hat{\alpha}=0.75$. With increased noise to 1%, there is a very small increase in the estimation errors: $\hat{R}_{\infty}=84.60\ \Omega$, $\hat{R}_1=38.64\ \Omega$, $\hat{C}=2.25\ \mu\text{F}$ and $\hat{\alpha}=0.75$. Most importantly, even in cases of higher noise levels (5% and 10%), the estimated values are still close to the reference: $\hat{R}_{\infty}=84.55\ \Omega$, $\hat{R}_1=38.61\ \Omega$, $\hat{C}=2.25\ \mu\text{F}$ and $\hat{\alpha}=0.75$ and $\hat{R}_{\infty}=84.48\ \Omega$, $\hat{R}_1=38.57\ \Omega$, $\hat{C}=2.25\ \mu\text{F}$ and $\hat{\alpha}=0.75$, respectively. This supports the assumption that even with a noisy dataset the proposed algorithm can accurately estimate the Cole-impedance parameters.

Parameter	Reference	¹⁹ , (MATLAB 2016a on Dell Precision 3240)	RE (%) Ref. ¹⁹	This work (MATLAB 2013b on Dell G5)	RE (%) This work
R_{∞} (Ω)	84.40	84.56	0.193	84.62	0.255
R_1 (Ω)	39.20	38.65	1.412	38.65	1.402
C (μF)	2.31	2.01	12.84	2.25	2.65
α	0.747	0.759	1.558	0.75	0.40

Table 2. Performance comparison of our estimation method and CNLS method¹⁹.

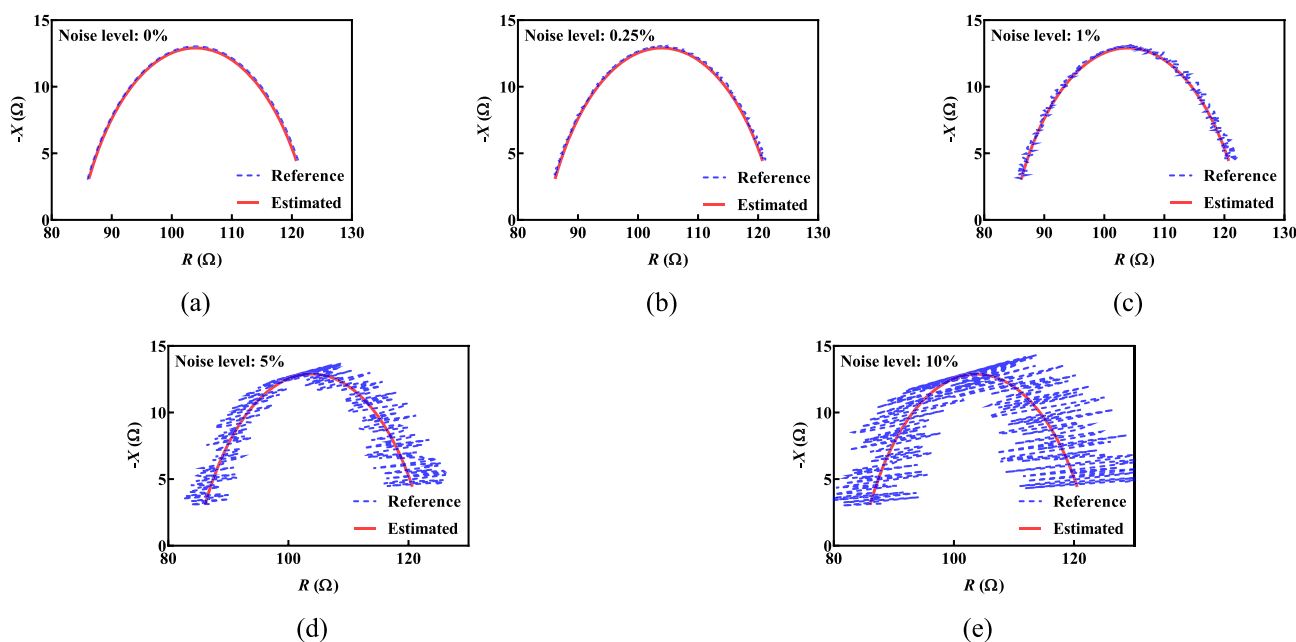


Figure 3. The Nyquist plots of reference and estimated values for five different noise levels: (a) 0%, (b) 0.25%, (c) 1%, (d) 5% and (e) 10%.

The frequency distribution of the estimated parameters \hat{a}_1 , \hat{a}_2 and \hat{b}_1 . The parameters of the Cole-impedance model have different frequency ranges over which they dominate the overall impedance. For example, R_∞ is the theoretical resistance at high (e.g. infinite) frequency, while R_1 is equal to $R_0 - R_\infty$, where R_0 is the theoretical resistance at low (e.g. $\omega = 0$ rad/s) frequency. This results from the impedance of the constant phase element (C) which is very high at low frequency (resulting in the majority of the applied current for an impedance measurement to flow through R_1), while it has very low impedance at high frequency (resulting in the majority of the current flowing through it and not R_1). The contribution of C to the overall impedance is expected to be most recognizable around the characteristic frequency (inverse of the time constant $(R_1 C)^{1/\alpha}$). Therefore, it can be expected that elements of the arrays $a_1(\omega_i)$, $a_2(\omega_i)$ and $b_1(\omega_i)$ will also have different frequency distributions.

Our initial analysis of this effect utilizes the reference dataset ($R_\infty = 84.40 \Omega$, $R_1 = 39.20 \Omega$, $C = 2.31 \mu\text{F}$ and $\alpha = 0.747$)¹⁹ with 0.25% noise, 3 kHz–1 MHz frequency band, and $N = 256$ logarithmically spaced points. Estimated values of the Cole-model parameters using Method 1 applied to this dataset are: $\hat{R}_\infty = 84.62 \Omega$, $\hat{R}_1 = 38.65 \Omega$, $\hat{C} = 2.25 \mu\text{F}$ and $\hat{\alpha} = 0.75$. The frequency distributions of elements of arrays $a_1(f_i)$, $a_2(f_i)$ and $b_1(f_i)$ are presented with blue dots in the Fig. 4 (a)–(c). These values are calculated at different frequencies but for the final estimated value of $\alpha = 0.75$. The average value (μ) of all presented values is shown as a red solid line in each graph. The standard deviation (σ) and ratio of σ/μ in % are also indicated with text labels on those figures. As it can be seen in Fig. 4a, the deviation of estimated values of $a_1(f_i)$ from the average (μ) is increasing at higher frequencies. Observing the deviations of estimated values of $a_2(f_i)$ and $b_1(f_i)$ in Fig. 4b and c, the deviations are the smallest in the central frequency band; approximately 30–100 kHz for $a_2(f_i)$ and 60–100 kHz for $b_1(f_i)$.

In the next step only lower frequencies (from $i = 1$ to $N/2 - 1$) are used for $a_1(f_i)$ estimation and central frequencies (from $i = N/2 - 10$ to $N/2 + 10$) for $a_2(f_i)$ and $b_1(f_i)$. These specific frequency ranges are chosen based on the subjective analysis of results shown in Fig. 4a–c. Further analysis is needed for a generalized approach to the selection of frequency-points, however the bands selected here do illustrate how reducing the frequency band to subsets where the behavior is dominated by that parameter improves its estimation. To illustrate, estimated values of model parameters using these reduced frequency bands are: $\hat{R}_\infty = 84.47 \Omega$, $\hat{R}_1 = 39.10 \Omega$, $\hat{C} = 2.23 \mu\text{F}$ and $\hat{\alpha} = 0.75$. Note this improves the estimation accuracy with reduction of relative errors from 0.255% to 0.088%, and from 1.412% to 0.26% for \hat{R}_∞ and \hat{R}_1 , respectively. The frequency distributions of elements of arrays $a_1(f_i)$, $a_2(f_i)$ and $b_1(f_i)$ from the reduced frequency estimation process are presented with blue dots in the Fig. 4d–f. Ratios of σ/μ for $a_1(f_i)$, $a_2(f_i)$ and $b_1(f_i)$ are decreased from 0.38%, 2.45% and 2.15% to 0.15%, 0.23% and 0.12% respectively, indicating much more stable estimations. Therefore, there is a possibility that the complete set of measured frequencies are not needed to estimate all parameters which can reduce the computational cost of the algorithm. But further development of this approach is required to develop approaches to identify the optimum frequency band based on the collected measurements (which is planned for future works).

Estimation with reduced datapoints. The parameter estimation with the proposed algorithm was analyzed using datasets with a frequency range from 3 kHz to 1 MHz with 256 logarithmically spaced points to facilitate direct comparison with reference¹⁹. However, reducing the number of frequency points in dataset may improve the performance of portable/wearable devices with constrained resources in terms of both memory

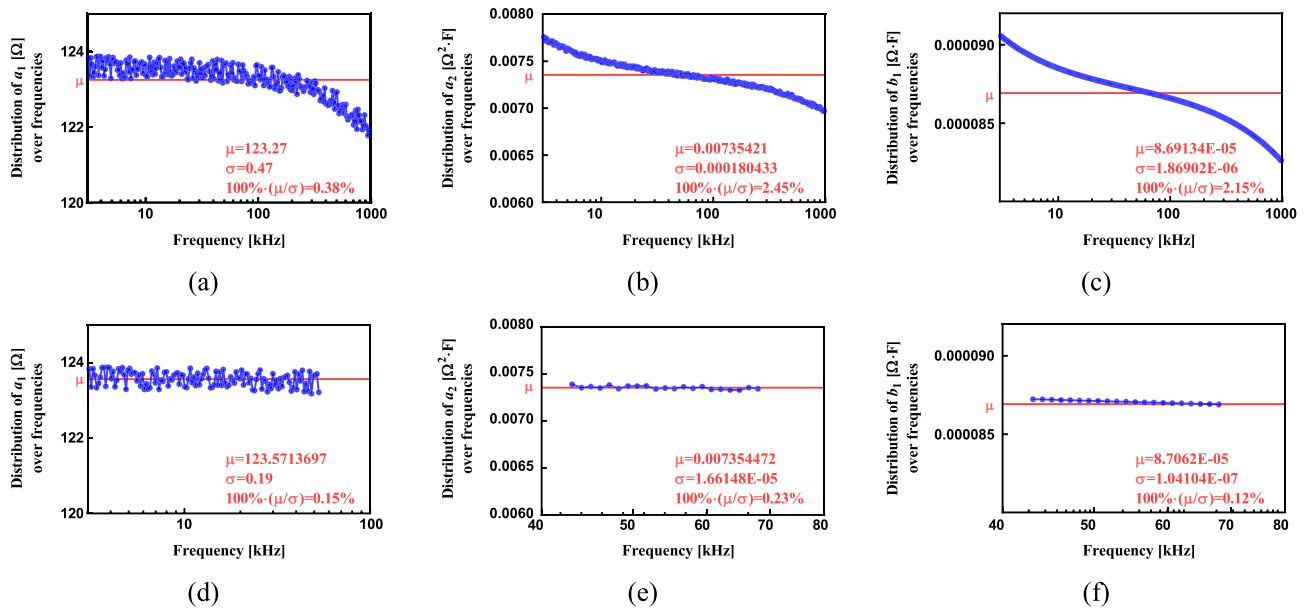


Figure 4. The frequency distribution of estimated values with Eqs. (10)–(12) in complete frequency range: (a) parameter a_1 , (b) parameter a_2 , (c) parameter b_1 ; and in reduced frequency range: (d) parameter a_1 , (e) parameter a_2 , (f) parameter b_1 .

and power. To evaluate the effect of reducing datapoints on the algorithm performance, the proposed estimation method was applied to the reference dataset ($R_\infty = 84.40 \Omega$, $R_1 = 39.20 \Omega$, $C = 2.31 \mu\text{F}$ and $\alpha = 0.747$)¹⁹ with 0.25% noise, while number of points initially was $N = 256$, and then was decreased to 100, 20, 6, and 3. The estimated parameters for each case were: $N = 256$ ($\hat{R}_\infty = 84.63 \Omega$, $\hat{R}_1 = 38.65 \Omega$, $\hat{C} = 2.25 \mu\text{F}$, $\hat{\alpha} = 0.75$), $N = 100$ ($\hat{R}_\infty = 84.63 \Omega$, $\hat{R}_1 = 38.65 \Omega$, $\hat{C} = 2.25 \mu\text{F}$, $\hat{\alpha} = 0.75$), $N = 20$ ($\hat{R}_\infty = 84.15 \Omega$, $\hat{R}_1 = 39.99 \Omega$, $\hat{C} = 2.49 \mu\text{F}$, $\hat{\alpha} = 0.74$), $N = 10$ ($\hat{R}_\infty = 84.58 \Omega$, $\hat{R}_1 = 38.81 \Omega$, $\hat{C} = 2.54 \mu\text{F}$, $\hat{\alpha} = 0.74$), $N = 6$ ($\hat{R}_\infty = 84.80 \Omega$, $\hat{R}_1 = 39.50 \Omega$, $\hat{C} = 2.94 \mu\text{F}$, $\hat{\alpha} = 0.73$), and $N = 3$ ($\hat{R}_\infty = 84.56 \Omega$, $\hat{R}_1 = 39.03 \Omega$, $\hat{C} = 2.21 \mu\text{F}$, $\hat{\alpha} = 0.75$). The $N = 3$ case is presented because it represents the theoretical minimum for the parameter estimation because three points are needed using the central difference approximation of the first derivative. Frequency range from 10 to 500 kHz was used for $N = 6$ ³⁰. The Nyquist plots of the reference and estimated impedance for each value of N are shown in Fig. 5a–f. Each case shows very good agreement with the ideal reference values confirming that reducing datapoints (even down to the theoretical limit) did not significantly impact the algorithm accuracy.

Embedded hardware-based estimations. The noisy datasets used in subsection "Comparison with reference"¹⁹ to test the proposed algorithm on a laptop computer were also used to test the algorithm on three embedded hardware platforms. The reference values of resistance and reactance were stored as arrays in the flash memory of the microcontrollers. Identical program code was compiled for each of the three platforms (Arduino Mega2560, Raspberry Pi Pico and XIAO SAMD21) using the Arduino IDE. After successful compiling and programming, the algorithm was executed on each platform. The comparison of estimated parameter values using our method deployed on the laptop, our method deployed on the embedded hardware platforms, and the CNLS-based method from¹⁹ deployed on laptop and Raspberry Pi 3 board are given in Table 3. The proposed method was optimized for execution on the embedded hardware by altering the brute-force search/step of α (originally proposed as a search from 0 to 1 in steps of 0.01). Here, a course search was performed with an α -step of 0.1 (from 0.1 to 0.9) with a second fine search performed with an α -step of 0.01. The fine search was performed using the course search value with the lowest error (α_1) for the precise search in the range from $\alpha_1 - 0.1$ to $\alpha_1 + 0.1$. This approach slightly increases the utilized flash and RAM resources, but reduces the number of iterations to find optimal value of α . Consequently, the consumed energy is reduced which is a very important consideration for portable and wearable devices. For example, with a fixed step of 0.01 to sweep the range from 0 to 1 requires 101 iterations. But if the two-step estimation is performed, then 11 iterations are needed for the 0.1 step course sweep and 21 steps for the precise sweep (32 iterations total). Therefore, total execution time and energy consumption are reduced by approximately one-third.

From the results provided in Table 3, our method deployed on the Raspberry Pi Pico (operating on clock speed of 133 MHz) required slightly higher t_{exe} than the more powerful Raspberry Pi 3 board with clock speed of 1.2 GHz³¹. This highlights the benefit of using the proposed algorithm which did not require Python (an interpreted language) and supporting libraries. Note that the estimation accuracy is similar for all analyzed methods, confirming that the proposed algorithm has similar performance to established methods.

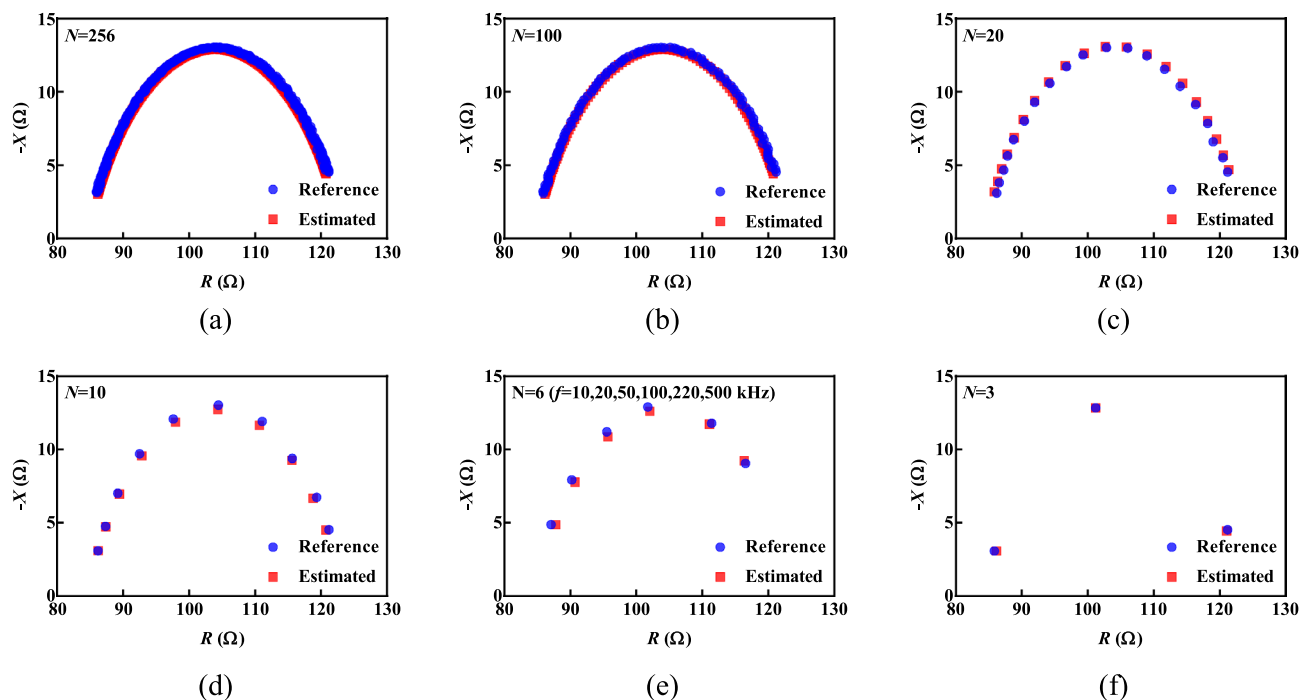


Figure 5. The Nyquist plots of reference and estimated values for different number of measurement points: (a) 256, (b) 100, (c) 20, (d) 10, (e) 6 and (f) 3.

	R_c (Ω)	R_f (Ω)	C (μF)	α	t_{exe} (s)
Reference	84.40	39.20	2.31	0.747	NA
CNLS ¹⁹ , (MATLAB 2016a on Dell Precision 3240)	84.56	38.65	2.01	0.759	14.76
CNLS ¹⁹ , (Python 3 on Raspberry Pi 3)	84.47	38.98	2.21	0.751	31.18
Our method (MATLAB 2013b on Dell G5)	84.62	38.65	2.25	0.75	0.15
Our method (Arduino Mega2560)	84.62	38.65	2.25	0.75	22.11
Our method (Raspberry Pi Pico)	84.62	38.65	2.25	0.75	10.05
Our method (XIAO SAMD21)	84.62	38.65	2.25	0.75	124.67

Table 3. Comparison of estimated values of model parameters using different hardware platforms. NA Not applicable.

For each of the three tested embedded platforms, the default configuration was used during program execution. This approach was adopted so that results represented performance for users using the platforms “as it is” without specialized configuration. We hypothesize that further improvements in the algorithm performance can be achieved by optimizing the configuration of the each platform, but future research is needed to confirm.

An interesting observation is that the compiled programs for each of the embedded platforms required different program code size and SRAM. This is expected because of the different microcontroller families and architectures (8-bits and 32-bits). Almost 66 kB of flash was required for the deployment on the Raspberry Pi Pico, but only 55kB and 16 kB of flash were required for the XIAO SAMD21 and Arduino Mega2560, respectively. Deployment on the Raspberry Pi Pico needed a lower percentage of available RAM (12%) when compared to 67% for the Arduino Mega2560. By default, the compiler does not provide usage of RAM in case of XIAO boards preventing its comparison here.

Because current draw and power consumption are important considerations for portable/wearable systems (with aims to reduce hardware size and increase battery life) the current/energy consumption of each hardware platform executing the proposed algorithm was measured. The current consumption of each embedded platform was obtained using a Siglent SPD3303C DC power supply and Sanwa CD770 multimeter. The Siglent SPD3303C was set to output 5 V and the Sanwa CD770 was connected in series (as an ammeter) with the SPD3303C and embedded platform. The current consumption was consistent across program execution and estimation of the parameters. The XIAO SAMD21 had the lowest input current (13.29 mA), followed by the Raspberry Pi Pico (22.18 mA) and Arduino Mega2560 (53.3 mA). Using the current measurements, the energy consumption to perform the estimation (obtained by multiplication of the measured current consumption and estimation time given in Table 3) was 1.11 J for the Raspberry Pico, 5.97 J for the Arduino Mega2560, and 8.73 J for the XIAO SAMD21.

Conclusion

In this work, we proposed a novel method for estimation of the Cole-impedance model parameters from bioimpedance datasets that does not use complex least squares methods. The proposed method showed a very good estimation accuracy, with improvements in terms of estimation speed when compared to related work from the literature. Further, increases in dataset noise and reductions in the number of datapoints in a dataset did not significantly degrade the accuracy of the algorithm.

This method is developed to support portable/wearable bioimpedance instruments with on-board processing requirements and limited computational hardware. As such, the performance of the algorithm was evaluated on three embedded platforms, the Arduino Mega2560, Raspberry Pi Pico and XIAO SAMD21. All platforms were shown capable of executing the proposed algorithm with accuracy comparable to laptop-based implementations (validating the algorithm for use on embedded hardware). Overall, the Raspberry Pi Pico was the platform with both the fastest execution time and smallest energy consumption.

Overall, the novelty and contributions of our work are: (1) a method for parameter estimation of the Cole-bioimpedance model suitable for deployment on low-complexity units (such as 8-bits microcontrollers), and (2) performance comparison of three embedded hardware-based platforms for identification of an appropriate system for future portable bioimpedance systems using the proposed algorithm.

Our future work will be directed towards the development of an integrated system for bioimpedance measurements and *in-situ* Cole-model parameter estimation. Moreover, we plan to use the presented method with experimentally obtained bioimpedance to resolve a limitation of the current work. Bioimpedance data collected from real tissues (and not synthetically generated datasets) is influenced by factors including electrode impedance (impacting low frequency measurements) and parasitic capacitances (impacting high frequency measurements). These factors can degrade the “fit” of the impedance data to the Cole-impedance model and is expected to degrade performance of the proposed algorithm. To resolve, future methods that identify features of data degradation and either correct or remove (prior to estimation of the Cole-impedance parameters) are needed. Further efforts will also investigate how to develop algorithms to fit more generalized and complex models beyond the Cole-impedance model to increase the possible range of bioimpedance datasets to which it can be applied.

Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Received: 8 January 2023; Accepted: 20 March 2023

Published online: 28 March 2023

References

- Kanoun, O. Impedance spectroscopy: From laboratory instrumentation to field sensors. *IEEE Instrum. Meas. Mag.* **23**, 4–7 (2020).
- Kassanos, P. Bioimpedance sensors: A tutorial. *IEEE Sensors J.* **21**, 22190–22219 (2021).
- Blanco-Almazan, D., Groenendaal, W., Cathoor, F. & Jane, R. Wearable bioimpedance measurement for respiratory monitoring during inspiratory loading. *IEEE Access* **7**, 89487–89496 (2019).
- Kusche, R. & Ryschka, M. Combining bioimpedance and EMG measurements for reliable muscle contraction detection. *IEEE Sens. J.* **19**, 11687–11696 (2019).
- Freeborn, T. J., Regard, G. & Fu, B. Localized bicep tissue bioimpedance alterations following eccentric exercise in healthy young adults. *IEEE Access* **8**, 23100–23109 (2020).
- Sel, K., Osman, D. & Jafari, R. Non-invasive cardiac and respiratory activity assessment from various human body locations using bioimpedance. *IEEE Open J. Eng. Med. Biol.* **2**, 210–217 (2021).
- Al-Surkhi, O. I. & Naser, R. Y. Detection of cell morphological changes of ischemic rabbit liver tissue using bioimpedance spectroscopy. *IEEE Trans. Nanobiosci.* **17**, 402–408 (2018).
- Li, J. *et al.* An approach for noninvasive blood glucose monitoring based on bioimpedance difference considering blood volume pulsation. *IEEE Access* **6**, 51119–51129 (2018).
- Vela, L. M., Kwon, H., Rutkove, S. B. & Sanchez, B. Standalone IoT bioimpedance device supporting real-time online data access. *IEEE Internet Things J.* **6**, 9545–9554 (2019).
- Hersek, S. *et al.* Wearable vector electrical bioimpedance system to assess knee joint health. *IEEE Trans. Biomed. Eng.* **64**, 2353–2360 (2016).
- Critcher, S. & Freeborn, T. System performance and user feedback regarding wearable bioimpedance system for multi-site knee tissue monitoring: Free-living pilot study with healthy adults. *Front. Electron.* **13**.
- Ferreira, J., Pau, I., Lindecrantz, K. & Seoane, F. A handheld and textile-enabled bioimpedance system for ubiquitous body composition analysis. An initial functional validation. *IEEE J. Biomed. Health Inform.* **21**, 1224–1232 (2017).
- Burns, R. P., Dunning, J. & Fu, M. J. A low-cost bioimpedance phase angle monitor for portable electrical surface stimulation burn prevention. *IEEE Trans. Circuits Syst. II(68)*, 1118–1122 (2021).
- Wu, Y., Jiang, D., Habibollahi, M., Almarri, N. & Demosthenous, A. Time stamp—A novel time-to-digital demodulation method for bioimpedance implant applications. *IEEE Trans. Biomed. Circuits Syst.* **14**, 997–1007 (2020).
- Mosychuk, V. S., Timoshenko, G. V., Sharpan, O. B., Tkachuk, B. V. & Tomashevskyi, R. S. Bioimpedance monitoring of dialysis patients during ultrafiltration. in *2016 IEEE 36th International Conference on Electronics and Nanotechnology (ELNANO)* 236–239 (IEEE, 2016). <https://doi.org/10.1109/ELNANO.2016.7493056>.
- Kalvoy, H. & Sauter, A. R. Detection of intraneural needle-placement with multiple frequency bioimpedance monitoring: A novel method. *J. Clin. Monit. Comput.* **30**, 185–192 (2016).
- Freeborn, T. J. A survey of fractional-order circuit models for biology and biomedicine. *IEEE J. Emerg. Sel. Topics Circuits Syst.* **3**, 416–424 (2013).
- Kyle, U. Bioelectrical impedance analysis?part I: review of principles and methods. *Clin. Nutr.* **23**, 1226–1243 (2004).
- Freeborn, T. J. Performance evaluation of raspberry Pi platform for bioimpedance analysis using least squares optimization. *Pers. Ubiquit. Comput.* **23**, 279–285 (2019).
- Cole, K. S. Permeability and impermeability of cell membranes for ions. in *Cold Spring Harbor symposia on quantitative biology* vol. 8 110–122 (Cold Spring Harbor Laboratory Press, 1940).
- Tsirimokou, G., Psychalinos, C. & Elwakil, A. *Design of CMOS Analog Integrated Fractional-Order Circuits*. (Springer International Publishing, 2017). <https://doi.org/10.1007/978-3-319-55633-8>.
- Seoane, F., Buendía, R. & Gil-Pita, R. Cole parameter estimation from electrical bioconductance spectroscopy measurements. in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology* 3495–3498 (IEEE, 2010).
- Yousri, D. A., AbdelAty, A. M., Said, L. A., AboBakr, A. & Radwan, A. G. Biological inspired optimization algorithms for cole-impedance parameters identification. *AEU-Int. J. Electron. Commun.* **78**, 79–89 (2017).
- Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V datasheet. <https://www.microchip.com/wwwproducts/en/ATMEGA2560>, Accessed on 1st November 2022.
- Arduino Mega 2560 R3. <https://www.sparkfun.com/products/11061>, Accessed on 26th August 2022.
- RP2040 datasheet. <https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf>, Accessed on 1st November 2022.
- Raspberry Pi Pico. <https://www.sparkfun.com/products/17829>, Accessed on 26th August 2022.
- SAM D21/DA1 Family - Low-Power, 32-bit Cortex-M0+ MCU with Advanced Analog and PWM. https://ww1.microchip.com/downloads/en/DeviceDoc/SAM_D21_DA1_Family_DataSheet_DS40001882F.pdf, Accessed on 1st November 2022.
- Seeed Studio XIAO Samd21. <https://www.seeedstudio.com/Seeeduino-XIAO-Arduino-Microcontroller-SAMD21-Cortex-M0+-p-4426.html>, Accessed on 9th September 2022.
- Al-Surkhi, Omar. I., Riu, P. J., Vazquez, F. F. & Ibeas, J. Monitoring Cole-Cole Parameters During Haemodialysis (HD). in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* 2238–2241 (IEEE, 2007). <https://doi.org/10.1109/IEMBS.2007.4352770>.
- Digikey- Raspberry Pi3. <https://www.digikey.be/en/products/detail/raspberry-pi/RASPBERRY-PI-3/6152799>, Accessed on 15th June 2022.

Acknowledgements

This research was funded through the European Union's Horizon 2020 research and innovation programme under grant agreement No. 854194. T.B.Š. is grateful to the Ministry of Education, Science and Technological Development of Serbia for helping this research through technological project 451-03-47/2023-01/200103.

Author contributions

Conceptualization, M.S.; methodology, M.S., T.J.F., and T.B.S.; software, M.S. and T.B.S.; validation, M.S., T.J.F., T.B.S., A.K.S. V.J and G.M.S.; formal analysis, M.S., T.J.F., and T.B.S.; investigation, M.S.; resources, G.M.S.; data curation, M.S.; writing—original draft preparation, M.S.; writing—review and editing, M.S., T.J.F., T.B.S., A.K.S.

V.J and G.M.S.; visualization, M.S.; supervision, V.J. and G.M.S.; project administration, G.M.S.; funding acquisition, G.M.S. All authors have read and agreed to the submitted version of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023