

## Article

# Unjumbling Procedure in the Algorithmic Analysis of Biomechanical Torques Induced by Electrical Stimulation: Case Study of the Lower Limb

Matthew J. Taylor <sup>1,2,\*</sup>, Ché Fornusek <sup>3</sup>, Philip de Chazal <sup>2</sup> and Andrew J. Ruys <sup>1</sup>

<sup>1</sup> School of Aerospace, Mechanical and Mechatronic Engineering, Faculty of Engineering and IT, The University of Sydney, Camperdown, NSW 2006, Australia; andrew.ruys@sydney.edu.au

<sup>2</sup> Charles Perkins Centre, The University of Sydney, Camperdown, NSW 2006, Australia; philip.dechazal@sydney.edu.au

<sup>3</sup> Discipline of Exercise and Sport Science, Sydney School of Health Sciences, Faculty of Medicine and Health, The University of Sydney, Camperdown, NSW 2006, Australia; che.fornusek@sydney.edu.au

\* Correspondence: mj.taylor.1290@gmail.com; Tel.: +61-478-033-859

**Abstract:** Functional Electrical Stimulation (FES) uses electrical pulses to cause muscles to contract synthetically. When muscles contract due to stimulation, torques are produced around joints attached to the muscle. It is important to understand torques being produced, for reasons such as safety and indirect fatigue measurement. Given the translatability of FES devices into the home for therapy, having ways to non-invasively measure muscle responses to stimulation is essential for understanding and diagnosing the biomechanical response of the human body. Here, we present data from a stimulation experiment examining knee joint torques (KJTs) arising when quadriceps are subjected to electrical stimulation. A novel algorithm for computing and summarizing KJT data into a series of simplified parameters was developed using MATLAB software. From this, we draw some conclusions about the effect of changing the stimulation duty cycle on the resultant KJT. We believe this method will provide researchers with a tool to measure torque in a semi-automated, convenient fashion.

**Keywords:** biomechanics; signal processing; isometric; torque; home exercise; functional electrical stimulation



**Citation:** Taylor, M.J.; Fornusek, C.; de Chazal, P.; Ruys, A.J. Unjumbling Procedure in the Algorithmic Analysis of Biomechanical Torques Induced by Electrical Stimulation: Case Study of the Lower Limb. *Appl. Sci.* **2022**, *12*, 5711. <https://doi.org/10.3390/app12115711>

Academic Editor: Hanatsu Nagano

Received: 11 March 2022

Accepted: 26 May 2022

Published: 3 June 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The technique of Functional Electrical Stimulation (FES) is one where pulse trains are delivered to muscles by a stimulator, for example, to augment their functional ability [1]. These pulse trains are delivered by placing electrodes over the surface of an individual's skin. Electricity runs between the electrodes, stimulating the nerves, which causes contraction of the muscles. This allows muscles to artificially contract. Therefore, stimulation allows for muscular exercise to occur when it may otherwise not be possible [2,3]. In more recent times, it has also been used to alleviate muscle weakness during the pandemic [4]. During electrical stimulation of muscles, the movement of muscles produces torques around the muscle's associated joints. Therefore, an aim of FES biomechanics is to estimate these torques when muscles are undergoing stimulation. This allows for an indirect measurement that details how the stimulation elicits a mechanical response from a limb. This may also have applications in clinical research where an indirect measure of muscle strength is required. Isokinetic dynamometry has, for example, been used in strength assessments in research examining knee osteoarthritis [5,6].

Joint torques may be measured by use of isokinetic dynamometry. Machines such as the Biodex can allow for measurements to be made that reflect the functionality of muscles [7]. Typically, a limb is strapped to a machine that measures the torque around a joint, which articulates due to the action of connected muscles. Muscles develop force, which pulls on connected bones and joints via tendinous connection. When the isokinetic

dynamometer is used to measure stimulation applied to a stationary muscle, it is referred to as *isometric dynamometry*, as during isometric muscle contraction, the muscle's length remains unchanged. When these movements occur, a dynamometer acquires a set of torque–time data. Previous work by our group [8] developed a series of metrics and plots to quantify such data over a period of 25 min stimulation. However, also of interest to FES research is acute stimulation, where pulse trains are delivered over the course of a few minutes.

In other upcoming work stemming from this original method, each experimental condition involved the testing of one duty cycle of stimulation for a 25 min period. The duty cycle is the time ON compared to the time OFF of the pulse trains [9]. The work performed was an adapted protocol of Gentz and Moore [10], whereby pulse trains were to be delivered using a duty cycle of 1 s ON:3 s OFF (1:3) and comparisons to be made across different multiples of the 1:3 duty cycle (e.g., 1:6, 3:9). However, one major limitation encountered was the large variability in torques generated when the same duty cycle was tested on different days. As such, a period of “chronic” stimulation for 20 min may elicit markedly different KJTs (if the thighs are to be stimulated) across different days despite the same parameters being used for electrical stimulation. Our goal was to therefore measure KJTs elicited when the quadriceps was subjected to the 1:3 duty cycle and its multiples, randomized in one session and across different days.

If pulse trains are delivered in one session, they must be randomized to reduce experimental bias. For example, if a researcher chooses the order in which pulse trains are delivered, there may be some unconscious bias when selecting which trains are delivered when. In their study of alternating current duty cycles, Szecsi and Fornusek [11] described stimulation being delivered in “blocks”, comprised of different types of alternating current delivered at different duty cycles. Herein, a randomized method inspired by this paper is used to design a protocol delivering different duty cycles of pulsed current to the quadriceps.

## 2. Methods

### 2.1. Electrical Stimulation Protocol and Randomized Design

#### 2.1.1. Details of the Case Study Design

The volunteer for this work was a healthy male, age 25, who had no history of lower limb pathology preventing participation in the case study. The work was approved by the University of Sydney Human Research Ethics Committee (Project No.: 2016/798). The participant engaged in the case study with full consent. The same randomized pattern was delivered to the quadriceps on the human subject on different occasions. The KJT data generated were processed using a novel algorithm. This algorithm was developed to analyse torque–time waveforms resulting from isometric dynamometry under these conditions. The goal of this work was, thus, two-fold: (1) present a novel algorithmic method that can be used by researchers to process data from isometric dynamometry experiments, and (2) to infer some conclusions about the effect of changing ON and OFF time (duty cycle) when the quadriceps is subjected to stimulation, with the main effect variable being the KJT that arises as a result of stimulation. In a diagnostic sense, it is envisaged that this work could be used by researchers in understanding limb biomechanics during electrical stimulation. This is of particular importance when using electrical stimulation in patient populations who have an absence of sensation, as limb biomechanics may be used to help understand the safety of electrical stimulation exercise.

#### 2.1.2. Experimental Design and Rationale

An experiment was designed to analyse torque responses when the quadriceps was subjected to randomized pulses delivered at different duty cycles. The same duty cycles (1:3, 2:6, 3:9, 1:6, 1:9, 2:3, and 3:3) were used as in previous work by Gentz and Moore [10]. However, as a modification to their protocol, all duty cycles were delivered in each session of this work, using a randomized pattern.

### 2.1.3. Randomization Procedure

The method of randomizing pulses was a modified form of a procedure found in a recent study of Functional Electrical Stimulation by Szecsi and Fornusek [11]. In their paper, Szecsi and Fornusek discuss providing “eight stimulation blocks” and “...each block contained eight stimulation periods that corresponded to the (duty cycle AC) “conditions”” (p. 149). This protocol informed the development of a randomized stimulation sequence in which to deliver the seven duty cycles of interest.

To generate the randomized pattern, each duty cycle was assigned a number (1:3—1, 1:6—2, 1:9—3, 2:3—4, 3:3—5, 2:6—6, 3:9—7). Sample numbers are presented for the purposes of illustrating the randomization procedure (Table 1), and columns are labelled A–D inclusive. Each duty cycle was assigned a random number using the RAND() function in Excel (column C, Table 1). Then, the relative order of those random numbers was used to assign a number 1–7 in column 1. This was achieved by use of an Excel formula that aimed to produce the order of column C relative to column A (i.e., the indexing column).

**Table 1.** Setting up the randomization procedure. Column A—duty cycle permutation, Column B—the duty cycle in complete numeric form (e.g., 16 = a duty cycle of 1:6), Column C—a randomly generated number, Column D—the duty cycle permutations (referenced by their number in Column A), randomly assorted on a basis of the random number in Column C.

Column A	Column B	Column C	Column 1
1	13	0.56	3
2	16	0.15	7
3	19	0.47	5
4	23	0.97	1
5	33	0.54	4
6	26	0.64	2
7	39	0.21	6

The aim of the initial randomization procedure was also to yield seven randomized columns, such as column 1. The goal was to produce a  $7 \times 7$  matrix of randomized duty cycles (denoted by numbers 1 to 7 inclusive). No duty cycle could appear more than once in either row or column. To achieve this, the next six columns (i.e., 2–7 inclusive, Table 2) were randomized based on column D and a series of auxiliary matrices with randomized numbers. In summary:

Seven auxiliary matrices were set up, each with six rows and two columns. In each matrix, there were two columns. The first had six of the seven numbers of the sequence 1–7, with one missing in each column (Table 3). The second had a random number specified by RAND(). This process was performed as follows.

A formula was typed into the first cell of column 2. This formula was a conditional formula that instructed access to one of the seven auxiliary matrices on a basis of the value in column 1 of that row. For example, in row 1, column 1 of Table 2, the value is 3. In row 1 of column 2, the formula would instruct to access the auxiliary matrix of Table 3 where the value of 3 is missing (highlighted in yellow, for example). This cell would then be filled with the order of the first cell in the yellow column.

The formula was then dragged across the row, and as so, it would access the auxiliary matrix of interest in a downward fashion, producing the relative randomized order of each number across the row of Table 2.

This process would be repeated for each row of Table 2, accessing a different auxiliary matrix for each row.

This ensured that the numbers 1–7 appeared once in each row.

**Table 2.** Randomized Protocol Generator. Process adapted from Szecsi and Fornusek (2014).

	Partition	Rand No	1	2	3	4	5	6	7
1	13	0.56	3	6	7	2	4	1	5
2	16	0.15	7	2	5	1	6	4	3
3	19	0.47	5	1	7	4	2	3	6
4	23	0.97	1	6	5	7	2	4	3
5	33	0.54	4	7	3	5	6	1	2
6	26	0.64	2	1	4	3	5	7	6
7	39	0.21	6	3	5	2	4	1	7

**Table 3.** Auxiliary Matrices for Randomization Purposes.

7	6		5		4		3		2		1		
1	0.72	1	0.96	1	0.92	1	0.24	1	0.48	1	0.88	2	0.41
2	0.16	2	0.77	2	0.40	2	0.80	2	0.17	3	0.56	3	0.59
3	0.73	3	0.96	3	0.74	3	0.37	4	0.64	4	0.65	4	0.21
4	0.02	4	0.78	4	0.79	5	0.28	5	0.62	5	0.44	5	0.99
5	0.42	5	1.00	6	0.75	6	1.00	6	0.86	6	0.03	6	0.86
6	0.45	7	0.69	7	0.63	7	0.88	7	0.60	7	0.06	7	0.97

The final randomization matrix (Table 4) differs from any examples (i.e., Tables 2 and 3), as RAND() changes each time an operation is performed on Excel. Initially, colour codes were assigned such that a series of programs would be hand-programmed into the stimulator to deliver four randomized duty cycles in sequence. For example, the first four numbers (in white, 3, 7, 5, 4) would be delivered by one stimulator program, then the next four (in black, 2, 6, 1, 2) by another. However, it was decided that each stimulation partition (numbered) would be delivered individually for practical purposes. In addition, it was also decided to only do the first three rows of the pattern. Therefore, the final stimulation pattern was that of Table 5.

**Table 4.** Randomized Pattern Generated.

Partition			1	2	3	4	5	6	7
1	13	0.6228	3	7	5	4	2	6	1
2	16	0.696862	2	5	7	6	4	1	3
3	19	0.125953	7	3	4	1	6	5	2
4	23	0.961222	1	4	5	6	7	2	3
5	33	0.517979	4	6	3	7	2	5	1
6	26	0.434722	5	6	7	4	1	2	3
7	39	0.169117	6	3	4	5	1	7	2

**Table 5.** Final Randomized Pattern Used.

1	2	3	4	5	6	7
3	7	5	4	2	6	1
2	5	7	6	4	1	3
7	3	4	1	6	5	2

#### 2.1.4. Precise Stimulation Parameters

For each partition of stimulation delivered (Table 5), three pulse trains were delivered, with a ramp pulse initially and a partial pulse after signifying the commencement of the fourth pulse (Figure 1). When the fourth pulse commenced, the stimulation was manually changed so that the pulse of the next partition could begin. In total, there were five pulse

trains delivered in most partitions (Figure 1). Each pulse train produced a torque plateau. The schematic shown in Figure 1 is for illustrative purposes: one ramp peak (Pr), three peaks (P1, P2, and P3), and one partial pulse train (Pp), whereby stimulation at the given duty cycle was stopped such that the next duty cycle could be delivered.



**Figure 1.** Sample partition of stimulation shown in terms of torque generated. Peaks in order as they appear (left to right): ramp peak (Pr), peak 1 (P1), peak 2 (P2), peak 3 (P3), and partial pulse train (Pp).

Prior to the administration of the random sequence of 21 partitions ( $7 \times 3$ ), a warm-up series of pulse trains were delivered that were 10 pulses of 1:3 stimulation. Stimulation was provided by a hand-held Med 4 (OttoBock Healthcare Products Austria GmbH, Vienna, Austria) stimulator. Stimulation was delivered at 45 mA, 30 Hz with a pulse width of 300  $\mu$ s. There was no ramp-up or ramp-down automatically administered (ramp-up = ramp-down = 0-s specified in the stimulator), decided during trial runs. There were multiple trials performed ( $n = 7$ ), with six of these ultimately being processed by the algorithm due to technical issues with one dataset. Torque was measured throughout the stimulation using a Biodex System 2 isokinetic dynamometer. The moment arm was configured at a knee joint angle of 60 degrees flexion.

## 2.2. Signal Processing and Algorithm Development

### 2.2.1. Data Processing

Custom-designed MATLAB algorithms were drafted, tested, and produced to process torque data from randomized trials. The data-processing procedure (Figure 2) summarizes how the data were processed. Initially, an algorithm was made; then, refinements had to be made as partitions in blocks were plotted out of order (Figure 2). Therefore, this procedure details both peak-plotting procedures and the unjumbling of torque plateaus related to the initial randomized stimulation sequence provided.

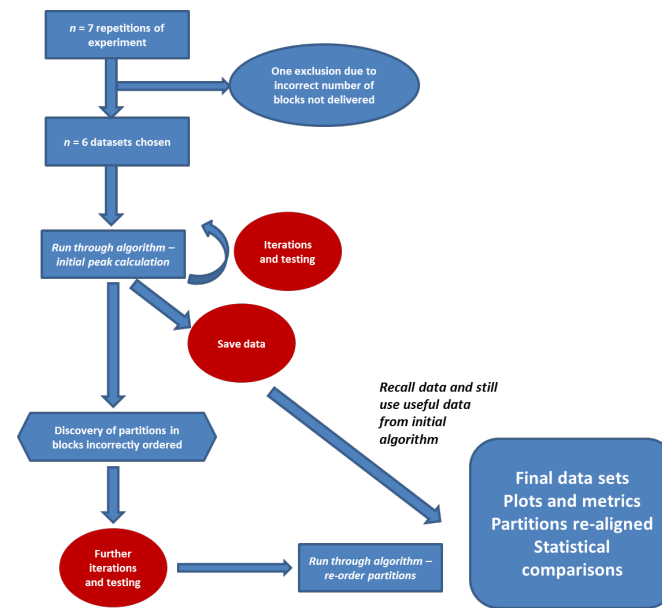
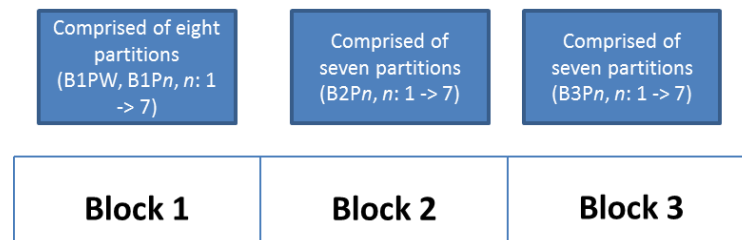


Figure 2. Summary of data processing.

### 2.2.2. Algorithm Description

For the purposes of discussion of how torque–time data were processed, a few terms must be defined, in terms of the stimulation delivered during experiments. Application of these terms in the context of torque–time waveforms is also depicted (Figure 3).



All three blocks have partitions. Each partition is one duty cycle delivered.  
All blocks have different order.

Figure 3. Stimulation delivered in randomized order.

Let:

A **partition** represent the five pulse trains delivered at a given duty cycle (one ramp pulse train, three pulse trains, and one partial pulse thereafter).

A **block** represent the total number of partitions where no duty cycle is repeated (i.e., seven partitions in this experiment).

The terms **partition** and **block** can refer to either (a) the electrical stimulation, or (b) the resultant torque waveform generated due to the stimulation. The block–partition notation used to describe pulse trains is presented in Figure 3. Three blocks of stimulation were delivered, each with seven partitions (one for each duty cycle). Block 1 also contained a warm-up partition. *BmPn* notation signifies block *m* partition *n*. B1PW was the warm-up partition in block 1. Therefore, in total, there were  $3 \times 7 + 1 = 22$  partitions of stimulation delivered.

The goal of algorithm development was, thus, to separate all duty cycles from torque–time data series and align all three repetitions from a given experiment (i.e., align duty cycle 1 from blocks 1, 2, 3, and then duty cycle 2, etc.). This was achieved by designing custom-made algorithms comprised of: a) large main pieces of code for “main” processing

(modules); b) associated functions to assist these modules in achieving their tasks (functions). A summary of all modules and functions is presented in Table 6. Algorithms were run and data processed as discussed below (next section).

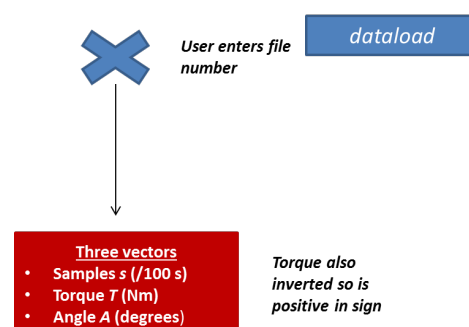
**Table 6.** Summary of Algorithm–Constituent Files and Main Purposes.

MODULES
<p><b>dataload:</b> Loads torque, time, and angle data for a given experimental run as input by the user.</p> <p><b>PeakFinder:</b> Baseline corrects data based on approximate torque due to gravity, finds peaks using findpeaks MATLAB function, requests user to remove peaks by running ginputremove or adding peaks by running ginputadd or ginputadd2 if further additions are required.</p> <p><b>PeakSplit:</b> Requests user to divide the entire torque–time waveform into three blocks, then asks the user to divide each block into partitions. Each block is plotted horizontally (i.e., rotated 90 degrees) so torque labels can be easily read.</p> <p><b>PeakAnalysis:</b> Exportation of blocks and partitions (all time/peak torque values) to Excel spreadsheet. Peak numbers and points defining blocks and partitions also exported.</p> <p><b>PeakAnalysis2:</b> Partitions of same number aligned from different blocks and plotted on one subplot (i.e., B1P1, B2P1, B3P1) plotted on same graph.</p> <p><b>PeakUnjumble:</b> Specified Excel sheet of previously exported metrics to be unjumbled. Runs blockreload and blockunjumble for each spreadsheet (i.e., experimental condition).</p> <p><b>PeakReplot:</b> Loads newly ordered partitions for each experimental condition. Runs blockreplot for each.</p>
FUNCTIONS
<p><b>ginputremove:</b> Requests the user to remove peaks from ROI.</p> <p><b>ginputadd:</b> Requests the user to add peaks from ROI.</p> <p><b>ginputadd2:</b> Requests the user to add peaks from ROI, subsequently.</p> <p><b>blockreload:</b> Reloads all blocks and partitions previously saved in Excel.</p> <p><b>blockunjumble:</b> Rewrites a new spreadsheet with partitions in correct order (i.e., that of the initial randomized sequence).</p> <p><b>blockreplot:</b> Generates a plot for the three repetitions of a given duty cycle by sub-plotting the partitions corresponding, as specified, by the randomization pattern.</p>

The modules and functions described in Table 6 may be summarized and connected by a series of control flow diagrams. These diagrams summarize pertinent features of each module and function so they may be understood by any researchers who wish to develop similar algorithms.

#### **dataload**—Initial Loading of Data

Upon acquisition of data from the Biodex, there are characteristically three datasets: samples (time), torque, and angle of the machine. In this experiment, the sampling rate of the Biodex was set at 100 Hz, so time samples were obtained every 1/100 of a second. Data were initially acquired from the Biodex in the .LVM format but saved as .XLSX. The dataload module (Figure 4) accesses these three datasets and imports them into MATLAB, based on the user-specified file number. It then inverts the torque, giving it a positive sign for ease of analysis thereafter.

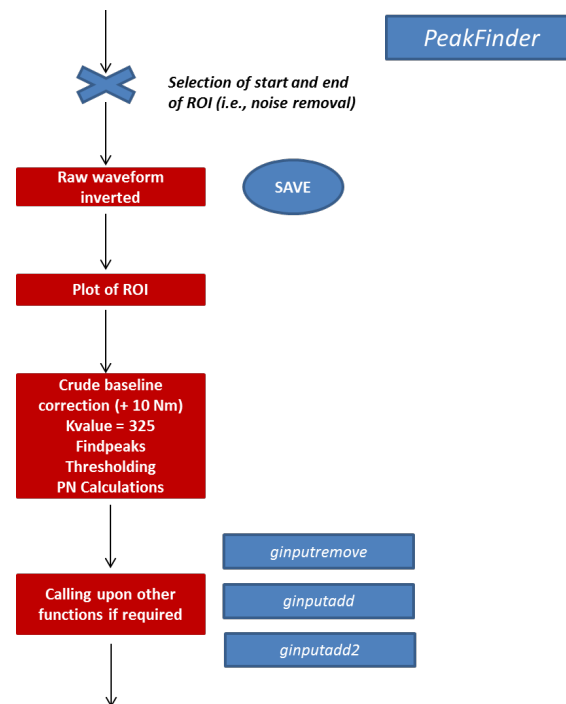


**Figure 4.** The dataload module.



### PeakFinder—Location of Peaks

This module (Figure 5) produces a form of the torque–time data in the region of interest (ROI), selected by the user manually. This ROI is the torque–time waveform with noise at the beginning and the end removed. It then converts the  $x$ -axis from samples (1/100 s) to time (s) and shifts the  $y$ -axis such that the torque of gravity is accounted for (baseline correction). In this code, the baseline correction value was taken as an approximate value of 10 Nm on the basis of prior calculations, which are the focus of other work.



**Figure 5.** The PeakFinder module.

The initial peak detection performed by the code is based on the built-in `findpeaks` function in MATLAB. This function requests a value that will define where the built-in function searches for a peak. In this code, it was referred to as the *Kvalue* (Figure 5). This was obtained by a trial-and-error approach. A similar “guessing” approach has also been used for example to determine vertical prominence value, which was adopted by McDonald and colleagues [12]. A range of *Kvalues* were trialed by changing the *Kvalue* in the preliminary code and seeing the resultant peak number *PN1*. This was then compared to the expected peak number, *EPN*, which was calculated by computing how many peaks were to be expected from the stimulation. Following this, the code thresholds the peaks such that those greater than or equal to 0.30 of the maximum of the peaks are displayed. The code then offers the option of adding, removing, or further adding peaks (Table 6).

### PeakSplit—Division of Peaks into Blocks and Partitions

This module (Figure 6) plots the torque–time data with superimposed peaks. Then, the user is asked to divide it up into three blocks, corresponding to the three blocks of stimulation administered to produce the torque (Figure 3). Following this, there are three new plots generated in succession—one showing the torque–time waveform for each block. Each graph is flipped 90 degrees clockwise, and labels are assigned to all peaks for easy inspection by the user. For each plot, the user is prompted to divide each into the respective number of partitions (eight for block 1, seven each for blocks 2 and 3). In both instances, division of the waveform into blocks and partitions is achieved by using a combination of the `ginput` and `pause()` functions in MATLAB, which allow selection of defining points for the user in an easy manner.



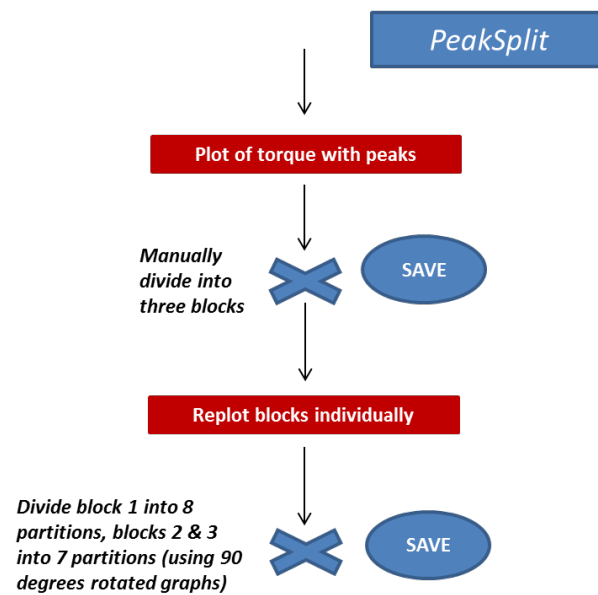


Figure 6. The PeakSplit module.

### PeakAnalysis and PeakAnalysis2—Saving of Peaks and Locations and Plotting

These two codes (Figure 7) save all the data obtained in the previous modules. In the first, PeakAnalysis takes all the block–partition data and exports them to an Excel spreadsheet (Figure 7, top). In the second, PeakAnalysis2 uses the block–partition data and generates subplots of peaks in partitions (Figure 7, bottom).

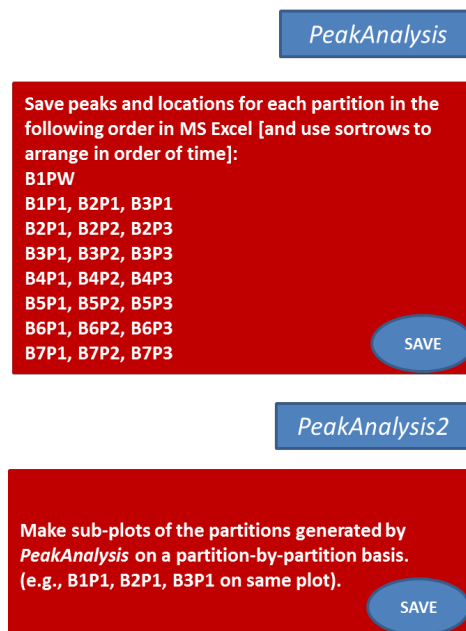


Figure 7. The PeakAnalysis and PeakAnalysis2 modules.

### PeakUnjumble and PeakReplot—Saving and Plotting of Peaks Properly Aligned

Following prior modules, block–partition data need to be aligned as per the randomization pattern (Table 5). This module (Figure 8) calls upon two functions that facilitate this and executes the procedure on an experimental run-by-run basis. The first, blockreload, reloads block–partition data from previously generated modules. The second, blockunjumble, resaves the data in a new format encoded on a basis of the initial randomized pattern. Then, PeakReplot (Figure 9) replots the properly ordered data.

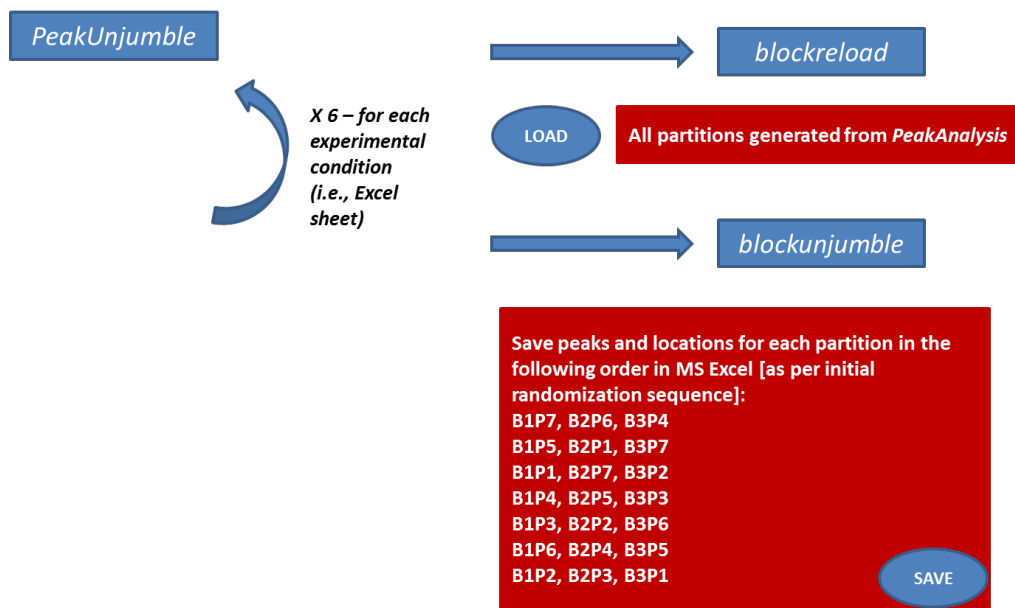


Figure 8. The PeakUnjumble module.

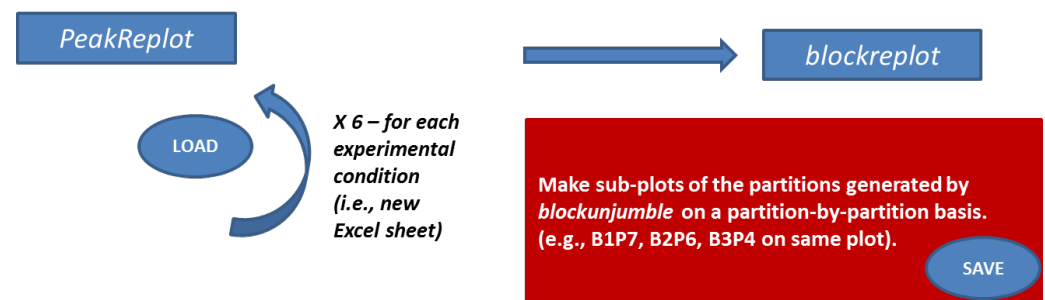


Figure 9. The PeakReplot module.

### 2.3. Data Analysis Methods

All six experimental runs were processed using the algorithms. Initial Excel spreadsheets were analysed, but then, the new codes unjumbling block-partition data had to be designed. Therefore, the final meaningful analyses were of unjumbled data.

Torque peaks and locations within each partition were analysed for each experimental run and then overall. A normalization torque  $T_n$  was found, firstly, by obtaining the average of contractions in the warm-up period, for all except the first contraction (ramp peak) and last contraction. For each duty cycle, there were three partitions (one from each block), and in each partition, approximately three peaks corresponding to those generated by automated stimulation (i.e., P1, P2, P3). An average peak value was found by taking the average peak across all three partitions for a given duty cycle (i.e., approximately nine peaks for most). The ramp peak  $P_r$  and partial peak  $P_P$  were excluded from this calculation. Then, for each duty cycle, the average peak value was divided by the normalization torque  $T_n$ . This yielded seven normalized average peak values for each experimental run (one for each duty cycle). Normalized average peak values were then averaged across all experimental runs for each duty cycle. Standard deviation, standard error, and relevant plots were generated from these metrics and compared. In addition to experimental stimulation and data analyses, a series of field notes were kept for some experimental runs (not discussed).

Peak counts were also a focus of the analysis. The number of peaks exported to Excel for each duty cycle was obtained by counting, for each experimental run, how many peaks there were (across all three partitions). These tallies were generated by using COUNT() in

Excel. In addition, peaks from plots generated by the algorithms were manually counted and tallied.

### 3. Results

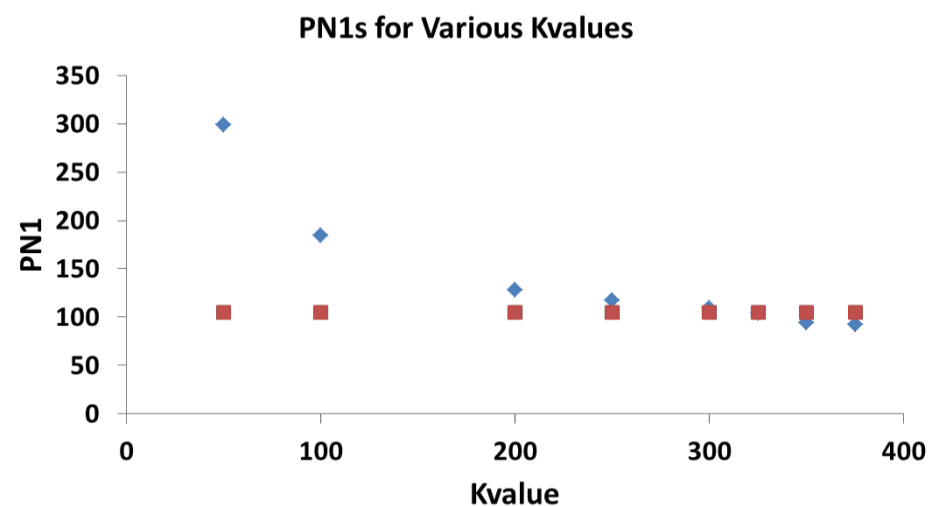
#### 3.1. Algorithmic Results

##### 3.1.1. Kvalue Iteration

One of the datasets was used to calculate a Kvalue that would produce a PN1 close to the EPN. The EPN was found in the code by the following formula:

$$\text{EPN} = \# \text{blocks} \times \# \text{partitions} (3 \text{ contractile peaks} + 1 \text{ ramp peak}) + 21 \text{ partial peaks from all partitions} = 105 \text{ peaks} \quad (1)$$

The number of peaks in the 1:3 warm-up was not included in this computation; however, the PN1's proximity to the EPN was a rough indicator of an appropriate Kvalue. The PN1 was the number of total peaks calculated by the algorithm. PN1s produced for various Kvalues are depicted in Figure 10.



**Figure 10.** Different PN1s produced by various Kvalues. Kvalue—the experimental “guess” as to the number of peaks, PN1—the resultant number of peaks computed by the algorithm when the corresponding Kvalue is entered into the code. Units on  $x$ - and  $y$ -axis—#peaks (dimensionless quantity).

##### 3.1.2. Plot Generation

Raw waveforms, torque waveforms with peaks superimposed, and block-partition diagrams were successfully produced by the algorithm. Shown here for the purpose of illustration is a torque waveform, with its corresponding waveform split into blocks and partitions. Further, the 1:3 peaks are used for illustrative purposes.

As shown in Figure 11, the algorithm split the total waveform into three blocks based on the applied randomized pattern. In each block, the resultant torque produced by each of the seven duty cycles applied is shown. Following the computation of peak values, these were taken by the algorithm, and the partitions from each block corresponding to a certain duty cycle were aligned. An example of the peaks produced by the 1:3 duty cycle from Run 2 are presented in Figure 12.

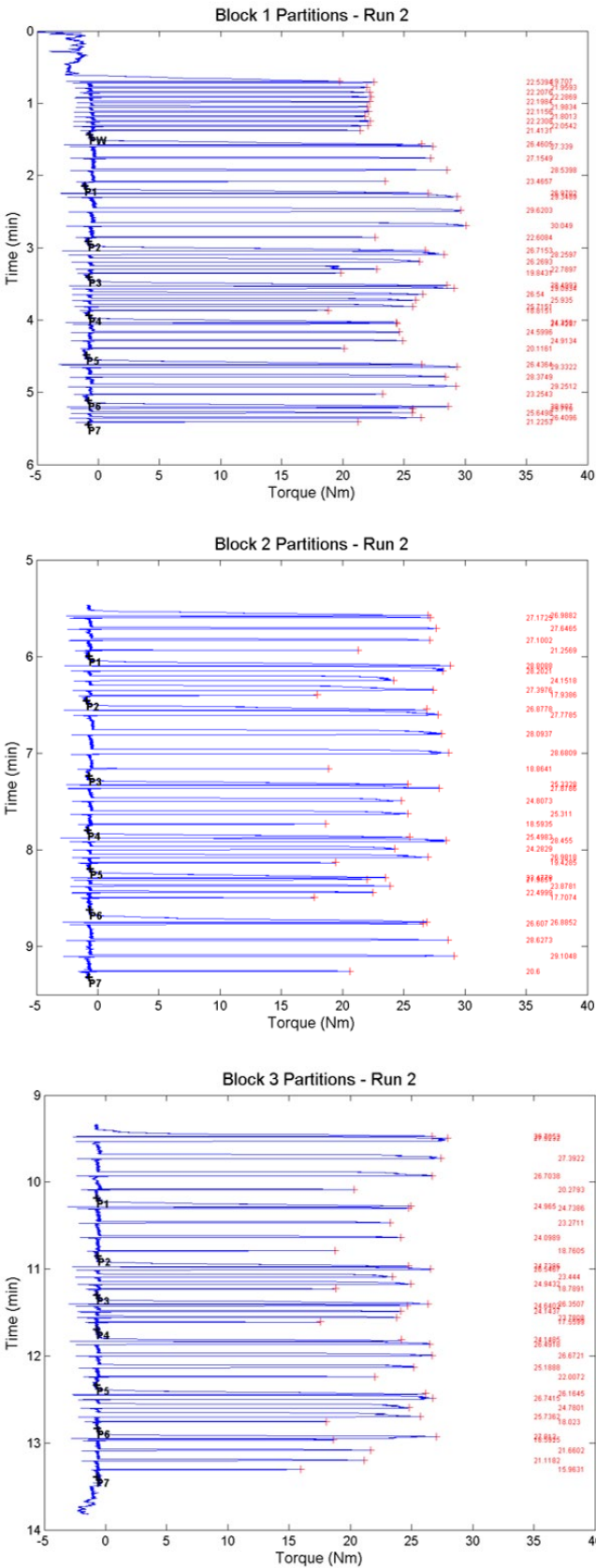
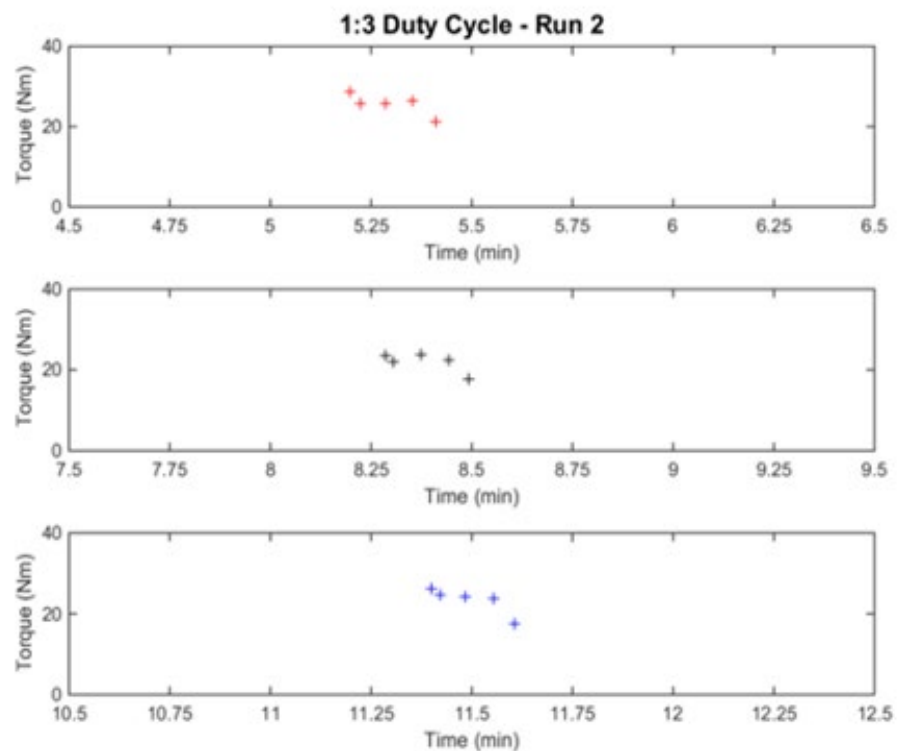


Figure 11. Individual peaks (+) for 1:3 duty cycle peaks of Run 2.



**Figure 12.** Peaks (+) produced by the 1:3 stimulation of each block.

### 3.1.3. Peak Counts

The algorithm also computed the number of peaks for each duty cycle of stimulation across the three partitions for a given run. The number of peaks exported to Excel by the algorithm were compared with those from manual tallying. Most of the peak counts were 15. Some values were slightly off, such as 14 peaks exported to Excel for 3:3 Run 6. Only one value had several more peaks exported to Excel than expected (25, 1:6 Run 4).

## 3.2. Experimental Results

### 3.2.1. Normalization Torques from Warm-Up

Normalization torques and the relevant statistical metrics of each warm-up contraction set (standard deviation, number of contractions, and standard error) were computed. There was considerable variation between the average torque generated from 10 warm-up contractions across different experiments on different days. The smallest was Run 3 ( $T_n = 11.304 \pm 0.313$  Nm), whereas the largest was Run 2 ( $T_n = 22.138 \pm 0.061$  Nm). Runs 4 and 7 were close in magnitude, with normalization torques of  $21.889 \pm 0.518$  Nm and  $21.674 \pm 0.120$  Nm, respectively. The values were then used to calculate ANT values.

### 3.2.2. Average Normalized Torques (ANT)

The average normalized torque (ANT) for each duty cycle across all three partitions, for each run, was computed. The smallest value was for the 1:3 duty cycle ( $ANT = 0.991 \pm 0.056$ ), whereas the largest was for the 3:9 duty cycle ( $ANT = 1.062 \pm 0.058$ ). There was little difference between the largest and smallest ANT values. In addition, the ANT values were plotted as functions of OFF, ON, and ON + OFF times (Figures 13–15, respectively). The strongest linear correlation was between ANT and ON + OFF time, with a coefficient of correlation,  $R^2 = 0.963$ . Similarly, the ANT values and OFF time permutation had a strong linear relationship, with an  $R^2 = 0.870$ . The correlation between ANT and the ON time permutation was less strong, with a coefficient of correlation  $R^2 = 0.605$ . All values were averages of three runs, except where there was a computation error (1:6 Run 4). Averages of this run were from only two of three partitions.

### ANT as a Function of OFF Time

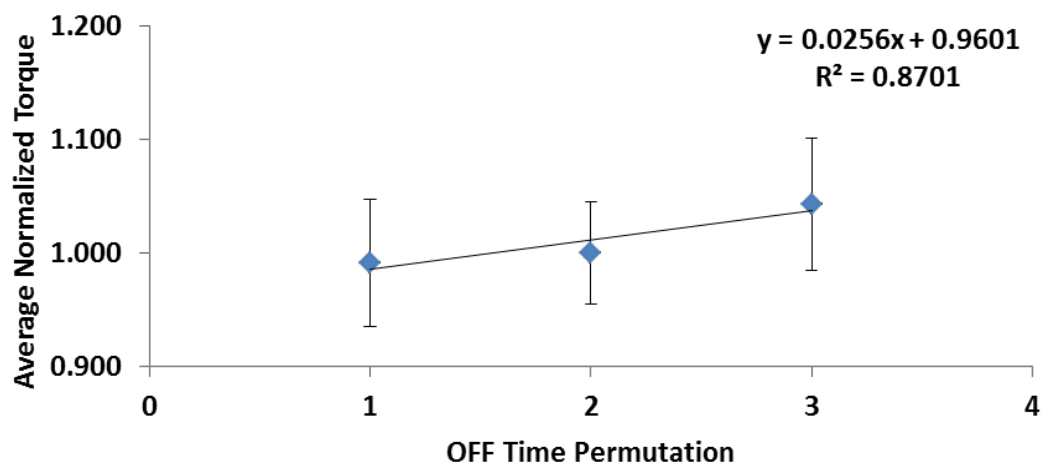


Figure 13. ANT as a function of OFF Time.

### ANT as a Function of ON Time

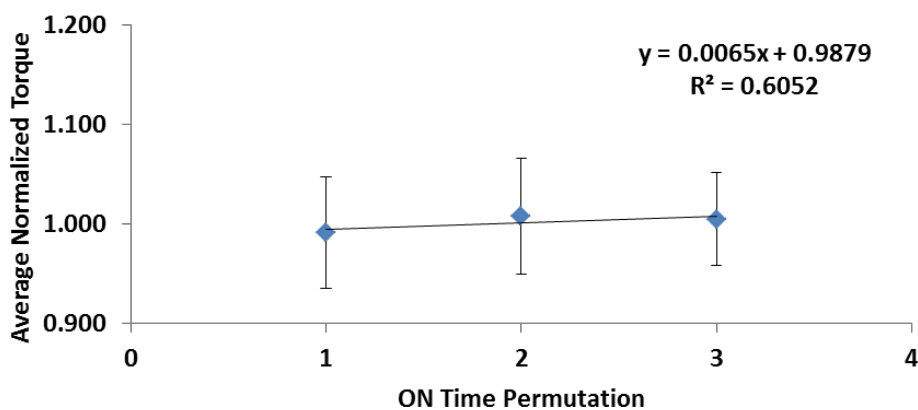


Figure 14. ANT as a function of ON Time.

### ANT as a Function of ON+OFF Time

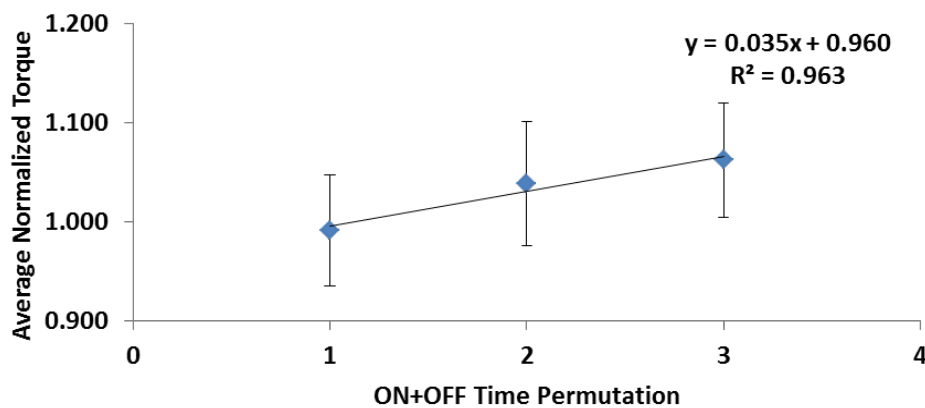


Figure 15. ANT as a function of ON + OFF Time. Average normalized torque (Nm), ON + OFF time permutation dimensionless (multiples of 1:3).

#### 4. Discussion

This study successfully developed an algorithm that may be used to compute torque peaks from isometric dynamometry data in an acute sense. The series of modules and functions developed enabled peak calculations over a short window of time. Furthermore, the randomization method inspired by Szecsi and Fornusek [11] was implemented to deliver pulse trains in a random fashion, reducing any bias of order. In addition, the use of block-partition notation and pulse train labelling within partitions may be used by future researchers as a framework upon which torques generated from electrical stimulation may be discussed.

A major function of this algorithm was peak detection of torque data. Using a modified form of previously developed code, in concert with the `findpeaks` function, enabled peaks to be detected with a high degree of accuracy. The use of the intrinsic `findpeaks` function has been used in various contexts to detect peaks in biological data. It has been used in gait analysis [13], muscle dynamics [14], surgical skills analysis [15], and sleep medicine to measure peaks in flow rate and epiglottic pressure data [16]. However, in our study, `findpeaks` was used for the case of torque peak detection in isometric dynamometry, which is less well-known. The `findpeaks` function is simple to implement and access via a single line command or the use of the Signal Processing Toolbox [17]. However, the presented algorithm significantly expanded upon the basic `findpeaks` function through the addition of a user-prompted set of programs (namely, `ginputadd`, `ginputremove`, `ginputadd2`). These user-prompted commands allowed for the removal of anomalous peaks or the addition of missing peaks in dynamometry data that cannot be picked up by the `findpeaks` function, which uses a periodic detection algorithm.

These algorithms put forward in this paper, therefore, may be adopted by other researchers who aim to analyse data that do not occur with regular periodicity, similar to the presented data, where the torque waveform had several partitions of data with varying periods in between. These tailor-designed functions allow for the selection or removal of peaks from torque data so that the magnitude of all contractions can be calculated.

Future extension of this work could also involve sharing of the code of the algorithm on platforms such as Github [18]. It could further be adapted as a Python version, which has the advantage of being available for free. This could increase the likelihood of professionals who may not necessarily be familiar with coding to be involved in programming for diagnostic purposes. This may be particularly advantageous in a health sciences research setting, provided that there is adequate research support for clinical staff from, for example, a biomedical engineer researcher.

This experimental work demonstrated that, at most, there were minor differences in torques generated during stimulation at different duty cycles, as reflected by the ANT metrics. There was a slightly larger ANT for duty cycles with both ON and OFF time doubled and trebled, in comparison with those in isolation. Perhaps there would have been substantially larger differences if stimulation had been delivered for a longer period of time.

In addition, the relationship between the degree of “ON” and “OFF” and the decrease in torques/ANT metrics (i.e., fatigue) may differ between acute and chronic stimulation. For example, Bergström and Hultman [19] demonstrated that continuous stimulation can cause less decrease in force than intermittent stimulation in their study using a 1:1 duty cycle delivered at 300  $\mu$ s and 20 Hz. Similarly, Chasiotis and colleagues [20] also suggested that intermittent stimulation may cause more fatigue than continuous over a shorter period of time. However, if a muscle is subjected to continuous stimulation for much longer, there will be much more fatigue than a continuous stimulation analogue. Therefore, the degree of “chronicity” in the application of stimulation at different duty cycles may influence the relative effects of ON, OFF, and ON + OFF time permutations in fatiguing the muscle of interest.



## 5. Conclusions

In this paper, a novel algorithm was presented that could calculate peak torques from torque–time data from a muscle that underwent electrical stimulation. The algorithm put forward was able to detect these peaks with minimal user assistance, then unjumble and align them based upon a series of randomized pulse trains delivered to the muscle. This algorithm and data analysed post hoc showed that modest differences in average normalized torque occur when stimulation is delivered in an acute, randomized fashion. Therefore, it has been demonstrated that the algorithm developed was a simple-to-use method by which torque peaks may be calculated from acute data and unjumbled when the muscle is subjected to randomized stimulation pulse trains. This will provide future researchers with a method by which they may be able to understand how randomizing duty cycles of stimulation can influence knee joint torque over an acute period of stimulation.

**Author Contributions:** Conceptualization, M.J.T., C.F., A.J.R. and P.d.C.; Data curation, M.J.T.; Formal analysis, M.J.T.; Investigation, M.J.T.; Methodology, M.J.T., C.F., P.d.C. and A.J.R.; Project administration, M.J.T., A.J.R. and C.F.; Resources, M.J.T., C.F., P.d.C. and A.J.R.; Supervision, C.F., P.d.C. and A.J.R.; Writing—original draft, M.J.T.; Writing—review & editing, M.J.T., C.F., P.d.C. and A.J.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Faculty of Engineering and IT and the Charles Perkins Centre at the University of Sydney, Australia.

**Institutional Review Board Statement:** The study was conducted in accordance with the Declaration of Helsinki and approved by the Ethics Committee of The University of Sydney (Project number: 2016/798, 6 October 2016).

**Informed Consent Statement:** Written consent has been obtained from the participant to publish this paper.

**Acknowledgments:** I also wish to thank the staff of the Sydney School of Health Sciences at the University of Sydney, Cumberland Campus, for their support with the experimental biomechanics component of this study. Finally, to all the biomedical engineers, allied health professionals, and clients who use FES—this work is for you.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Van der Scheer, J.W.; Goosey-Tolfrey, V.L.; Valentino, S.E.; Davis, G.M.; Ho, C.H. Functional electrical stimulation cycling exercise after spinal cord injury: A systematic review of health and fitness-related outcomes. *J. NeuroEng. Rehabil.* **2021**, *18*, 99. [[CrossRef](#)] [[PubMed](#)]
2. Banerjee, P.; Clark, A.; Witte, K.; Crowe, L.; Caulfield, B. Electrical stimulation of unloaded muscles causes cardiovascular exercise by increasing oxygen demand. *Eur. J. Cardiovasc. Prev. Rehabil.* **2005**, *12*, 503–508. [[CrossRef](#)] [[PubMed](#)]
3. Hamada, T.; Hayashi, T.; Kimura, T.; Nakao, K.; Moritani, T. Electrical stimulation of human lower extremities enhances energy consumption, carbohydrate oxidation, and whole body glucose uptake. *J. Appl. Physiol.* **2004**, *96*, 911–916. [[CrossRef](#)] [[PubMed](#)]
4. Carraro, U.; Marcante, A.; Ravara, B.; Albertin, G.; Maccarone, M.C.; Piccione, F.; Kern, H.; Masiero, S. Skeletal muscle weakness in older adults home-restricted due to COVID-19 pandemic: A role for full-body in-bed gym and functional electrical stimulation. *Aging Clin. Exp. Res.* **2021**, *33*, 2053–2059. [[CrossRef](#)] [[PubMed](#)]
5. Zhang, X.; Pan, X.; Deng, L.; Fu, W. Relationship between Knee Muscle Strength and Fat/Muscle Mass in Elderly Women with Knee Osteoarthritis Based on Dual-Energy X-Ray Absorptiometry. *Int. J. Environ. Res. Public Health* **2020**, *17*, 573. [[CrossRef](#)] [[PubMed](#)]
6. De Zwart, A.H.; van der Esch, M.; Pijnappels, M.A.; Hoozemans, M.J.; van der Leeden, M.; Roorda, L.D.; Dekker, J.; Lems, W.F.; van Dieën, J.H. Falls associated with muscle strength in patients with knee osteoarthritis and self-reported knee instability. *J. Rheumatol.* **2015**, *42*, 1218–1223. [[CrossRef](#)] [[PubMed](#)]
7. Drouin, J.M.; Valovich-McLeod, T.C.; Shultz, S.J.; Gansneder, B.M.; Perrin, D.H. Reliability and validity of the Biodex system 3 pro isokinetic dynamometer velocity, torque and position measurements. *Eur. J. Appl. Physiol.* **2004**, *91*, 22–29. [[PubMed](#)]
8. Taylor, M.J.; Fornusek, C.; de Chazal, P.; Ruys, A.J. All talk no torque—A novel set of metrics to quantify muscle fatigue through isometric dynamometry in Functional Electrical Stimulation (FES) muscle studies. In Proceedings of the 4th International Conference on Mechanical Engineering Research (ICMER2017), Kuantan, Malaysia, 1–2 August 2017. [[CrossRef](#)]
9. Baker, L.L.; Mcneal, D.R.; Newsam, C.; Waters, R.L.; Wederich, C.L. *Neuro Muscular Electrical Stimulation—A Practical Guide*, 4th ed.; Los Amigos Research Institute, Inc.: Downey, CA, USA, 2000.

10. Gentz, L.; Moore, C. Effect of duty cycle on fatigue during stimulation of the quadriceps muscle. *Phys. Ther.* **1988**, *68*, 834.
11. Szecsi, J.; Fornusek, C. Comparison of Torque and Discomfort Produced by Sinusoidal and Rectangular Alternating Current Electrical Stimulation in the Quadriceps Muscle at Variable Burst Duty Cycles. *Am. J. Phys. Med. Rehabil.* **2014**, *93*, 146–159. [[CrossRef](#)] [[PubMed](#)]
12. McDonald, S.C.; Brooker, G.; Phipps, H.; Hyett, J. The Identification and Tracking of Uterine Contractions Using Template Based Cross-Correlation. *Ann. Biomed. Eng.* **2017**, *45*, 2196–2210. [[CrossRef](#)] [[PubMed](#)]
13. Mazumder, O.; Kundu, A.S.; Lenka, P.K.; Bhaumik, S. Multi-channel Fusion Based Adaptive Gait Trajectory Generation Using Wearable Sensors. *J. Intell. Robot. Syst.* **2016**, *86*, 335–351. [[CrossRef](#)]
14. Hasson, C.J. Neural representation of muscle dynamics in voluntary movement control. *Exp. Brain Res.* **2014**, *232*, 2105–2119. [[CrossRef](#)] [[PubMed](#)]
15. Trejos, A.L.; Patel, R.V.; Malthaner, R.A.; Schlachta, C.M. Development of force-based metrics for skills assessment in minimally invasive surgery. *Surg. Endosc.* **2014**, *28*, 2106–2119. [[CrossRef](#)] [[PubMed](#)]
16. Nguyen, C.D.; Amatoury, J.; Carberry, J.C.; Eckert, D.J. An automated and reliable method for breath detection during variable mask pressures in awake and sleeping humans. *PLoS ONE* **2017**, *12*, e0179030. [[CrossRef](#)] [[PubMed](#)]
17. Khan, B.; Hodics, T.; Hervey, N.; Kondraske, G.V.; Stowe, A.M.; Alexandrakakis, G. Functional near-infrared spectroscopy maps cortical plasticity underlying altered motor performance induced by transcranial direct current stimulation. *J. Biomed. Opt.* **2013**, *18*, 116003. [[CrossRef](#)] [[PubMed](#)]
18. GitHub. Available online: <https://github.com/> (accessed on 11 March 2022).
19. Bergstrom, M.; Hultman, E. Energy cost and fatigue during intermittent electrical stimulation of human skeletal muscle. *J. Appl. Physiol.* **1988**, *65*, 1500–1505. [[CrossRef](#)] [[PubMed](#)]
20. Chasiotis, D.; Bergström, M.; Hultman, E. ATP utilization and force during intermittent and continuous muscle contraction. *J. Appl. Physiol.* **1987**, *63*, 167–174. [[CrossRef](#)] [[PubMed](#)]