Depósito de Investigación de la Universidad de Sevilla

https://idus.us.es/

FLEXOR: A support tool for efficient and seamless experiment data processing to evaluate musculo-articular stiffness

Paris-Garcia, F.[1,], Ruiz-Zafra, A.[2], Noguera M.[3], Barroso-Caro, A.[4]*

[1] Department of Sports and Computer Science, Section of Physical Education and Sports, Faculty of Sports Sciences, University Pablo de Olavide, ES-41013 Seville, Spain
[2] Department of Computer Engineering, University of Cádiz, Avda. Universidad de Cádiz, nº 10, ES-11519. Campus de Puerto Real (Cádiz)
[3] Software Engineering Department, University of Granada, Spain, ETSIIT, Periodista Daniel Saucedo Aranda, s/n, 18014, Granada, Spain
[4] School of Engineering, University of Seville, Camino de los Descubrimientos, s/n, 41092 Sevilla, Spain.

* *Corresponding author.* Tel.: ; fax: +0-000-000-0000.
E-mail address: fparis@upo.es

ARTICLE INFO

ABSTRACT

Background and objective: The evaluation of musculo-articular stiffness (MAS) is an increasingly demanded procedure with applications in different fields, such as sports performance and lower limbs injury prevention. However, this task is non-automated, time-consuming and error-prone due to manual handling of data streams and files across several software applications. Despite the fact that process automation of validated procedures helps to prevent errors, there is still a lack of easy-to-use tools for analysis, management and visualization of MAS trials.

Methods: In the present work a tool called FLEXOR has been developed which applies mathematical methods and novel algorithms to automatically adjust curves of data streams for MAS analysis decreasing substantially time employed and errors. This tool permits to define different adjustment parameters, detect curve peaks and valleys, and display the results on the fly. FLEXOR has been implemented through a component-based software development (CBSD) process. All physiological fundamentals for the biomechanical measurement have been included in the tool developed. To describe the integration of all required components a 4+1 view model architecture has been used. The installation guide, the FLEXOR software and some data samples can be found on its GitHub repository (https://github.com/FlexorSoftware/flexor).

Results: A multiplatform software tool to simplify traditional complex and manual procedures for MAS analysis is obtained. The tool turns them into a simple all-in-one procedure, reducing processing times from hours to a few minutes. The methodology was tested on multiple datasets generated by previous tools in former procedures as well as on real-time trials in the laboratory, showing identical results.

Conclusion: The results show that the developed tool can accomplish an unfilled essential task in the analysis, management and visualization of MAS measurement. The presented software tool empowers analysts to handle the different studies, investigate different parameters related to each experiment and even test with different output parameters in each experiment, enabling real-time trials and shared studies between different analysts.

1.   Introduction

Free vibration techniques for evaluating the responsiveness of a 1-degree-of-freedom dynamic damped system of the musculo-articular stiffness (MAS) of the plantar flexor muscles have become validated procedures and are widely contrasted in the scientific literature [1-4]. This procedure models the observed mechanical response of the main muscle-tendon unit (MTU) involved in the plantar flexor movement [5]. The MAS is associated with the evaluation of the functional capacity linked to the physical activity of a subject, and its interpretation has multiple applications, such as sports performance measurement and clinical applications (e.g., normative values of clinical populations [9,10] and the risk factor of musculoskeletal injuries in lower limbs [11]). However, the applicability and widespread use of MAS has been limited thus far due to different issues. On the one hand, the measurement of this parameter requires a non-invasive, non-stress laboratory test, which implies the use of appropriate in-lab infrastructure, ad hoc mechanical devices and specific signal sensing units and devices [12]. In addition, it requires efficient treatment and processing of the signal data obtained from laboratory studies through different signal filtering techniques, mathematical transformations and the use of several software systems and tools. Nevertheless, and to the best of the authors' knowledge, there do not exist specific tools to support analysts throughout the signal and analysis procedure.

The absence of suitable software systems (e.g. all-in-one frameworks) to support the entire process results in the use of different, ad hoc and mutually independent software tools for each stage in the signal processing (signal capturing, signal filtering, and signal transformation, as well as signal manipulation). That is, for each one of such purposes, a specific software tool (e.g., LabVIEW, Mathematica, SPSS) is used in a standalone manner. The lack of integration between such tools, due to interoperability issues, represents a severe handicap for experiment manipulation. This problem also arises in other domains where the proprietary and non-open nature of some of the most popular software solutions hinder the use of their functionalities to foster interoperability between applications [13,14]. Thus, the definition of downstream signal analysis processes becomes affected since the translation of the output (e.g., files or byte streams) from different tools has to be done manually. As a matter of fact, the processing of data of one subject lasts around one or two hours on average. The situation with a sample procedure for the case of plantar flexor MAS measurement, where different software tools are needed throughout the analysis procedure (LabVIEW, Mathematica, and custom database software) can be observed in the next diagram (see Figure 1).

Signal analysts, physicians and sport-medicine practitioners interested in analysing MAS properties require automated tools that provide integrated suites of signal-processing functionalities to speed up the process in general, but also functionalities to easily configure and adjust ad-hoc signal parameters.
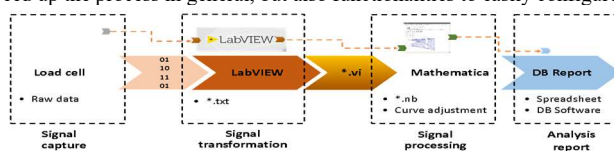

Fig. 1 - Sample procedure for plantar flexor MAS measurement using a load cell and different software tools

The aim is to observe the results in real-time because, normally, part of the processing and analysis of signals consists of obtaining a function curve that has been approximated by means of numerical methods. Thus, it is often necessary to try different adjustment parameters and check the results in real-time (i.e., as they are tested), as in a simulation setting. Otherwise, setting different adjustments and parameters entails going from one software application to another or the explicit relaunch of codified subroutines. Then, this type of analysis becomes impracticable or even unfeasible, besides incredibly cumbersome. Let us consider the burden that managing different files across different software tools by hand would imply. Again, without appropriate automated technical support, this task is rather time-consuming and error-prone [15], [16].

In the present paper, FLEXOR is presented as an integrated, seamless, software solution that implements several procedures, such as signal processing, data management methods, algorithms and flexible functionalities, to support different adjustments that empower physicians and sport-medicine practitioners to observe and measure the responsiveness (i.e., MAS) of plantar flexor muscle-tendon units (MTU) while enabling trial-and-error experimentation with different adjustment parameters and without having to move between different software applications.

In this sense, FLEXOR improves the efficiency and effectiveness of the MAS analysis procedure, integrating all phases of the acquisition, processing and post-processing of the signal received from sensing devices in a timely manner. Likewise, FLEXOR automates certain tasks that were performed manually in the past, such as the selection of appropriate data set intervals for subsequent analysis. In this sense, FLEXOR provides the following clear benefits:

1.   Simplification and time reduction of experimentation procedures.
2.   Quick checks of curve adjustments and parameterization on the fly.
3.   Simplification of experiment management and reduction of the number of errors due to human mistakes.
4.   Improved interoperability with potential third-party software analysis and applications through data representation for experimentation samples in a technology-independent format, such as JSON. Although desirable, comparison with other experimentation datasets from other research laboratories it is not feasible at present, due to the lack of standardization in data representation and contents, but this is a first step towards that direction.
5.   Multi-platform operation derived from the fact that FLEXOR is implemented in Java. This implies that FLEXOR is fully operable in most popular operating systems, i.e., Windows, Linux-Unix families and Mac OS.

Analytical and numerical processes implemented in the FLEXOR software allow some decision making processes (for example, the valuable data to use in the curve fitting) to be automated, which avoids additional human manipulations and errors.

This paper has been organized as follows. Section 2 analyses the background of the proposal, showing the most relevant research works in the area. Section 3 summarizes the theory and methods of this work. Section 4 presents the proposal to automatize the plantar-flexor stiffness assessment, introducing FLEXOR. An in-the-wild testing example of the proposal is described in Section 5 and the discussion about the results as well as the proposal are explained in Section 6. In the end, the conclusion of the paper and the future work are presented.

Nomenclature

MAS   musculo-articular stiffness
CBSD   component-based software development
MTU   muscle-tendon unit

2. Background

The MAS measurement and analysis is an important concern in many research avenues related to healthcare and wellbeing, such as sports performance. In this sense, there exists plenty of recent research concerning novel approaches to enhance stiffness assessment methods and techniques, as they involve different processing software including complex numerical analysis. In this section, it is presented a brief analysis of the most relevant pieces of work so far.

One of the most referenced paper in the field was published by Ditroilo et al. [5], who present a deep state of the art about the MAS assessment, describing the different measurement and analysis 'considerations' such as how to calculate the stiffness and the perturbations in the measurement process.

The representativity of the value of MAS is a controversial matter. It is accepted to assign the term MAS to what is obtained from a free-oscillation technique, Ditroilo et al [5]. MAS involves passive stiffness and stretch reflex and what is measured is an apparent value involving both effects. The contribution of these two effects to an apparent value of the stiffness is also a controversial question. A study of Blackburn et al. [6] showed that "passive stiffness accounted for 25% of the variance in the stiffness measure, with the free oscillation technique, while the musculature supported a load about 30% of maximal voluntary contraction". Additionally, Granata et al. [7] and Blackburn et al. [8] have shown that under active conditions, the effect of the passive stiffness is negligible. For each 'consideration', the authors proposed the most used and reliable method, exposing the mathematical approach to solve or calculate the different parameters required to obtain the desired measurement. However, all proposed methods entail the use of different software for each analysis step.

Another reference work that presents a survey about this topic was published by Fukashiro et al. [3], where the different methods available proposed by the authors over the years are briefly described.

The literature review of the work of the most active groups and authors shows that many authors have proposed ad-hoc solutions, improving legacy proposal presented in [5] and [3], with new and promising approaches based on calculation and optimization [17-19]. So far, these works have focused on the challenge itself, that is, how to measure the MAS, proposing the mathematical procedures required. Other approaches, focused on the analysis of postural control or limb movement from simple models, require the calculation of the endpoint stiffness. The stiffness of a joint has associated a series of physiological mechanisms and is accepted that this stiffness plays a fundamental role in the control both of posture as movement. [20] presents a study of time-domain and frequency-domain, multiple-input, multiple-output (MIMO) linear system identification techniques that can be used to estimate the dynamic endpoint stiffness of a multi-joint limb.

However, none of them describes how the different mathematical procedures are implemented, and if any software of tools is used to implement, develop or automate the solving method. In addition, most of the proposals do not describe how to gather the measures required, only describing a scheme picture to show the movement and position of somebody limbs, but not insight about the particular hardware and tools required. Some research works are more software-oriented, describing the ad-hoc solution proposed together with the software and tools used to implement the proposal, as well as test or example to validate the proposal for MAS measurement and analysis.

For instance, the research work presented in [21] describes a proposal to measure MAS of the ankle where Matlab is used as software to solve the calculation of stiffness properties. Other proposals require different private-vendor software, such as the project presented in [22] to measure MAS in the leg using Matlab and Motion Monitor software; or the project presented in [23], which combines Matlab and Origin.

However, although these projects describe standalone and fully-supported solutions, the use of third-party software (Matlab, Motion Monitor, Origin) arises two main drawbacks: 1) the different software-based solutions involve manual procedures, where the person responsible for the test has to carry out different tasks: use several software, generate different outputs, use them as input to other software, disseminate the information and finally, understand the appropriate interpretation; 2) the use of different private-vendor software solutions hinders the interoperability and integration of the system with other third-party systems and software.

Both drawbacks cause a long time required to carry out a full trial (experiment), because it requires many steps or tasks (e.g. import, export data, check the reliability of the data, implementation, etc.), but also increase the possibility of making mistakes. The manual management of the information along the process by trials supervisors, the load of files, selection of subsets, change of parameters, etc., fosters errors and mistakes, resulting in useless trials.

To the best of our knowledge, after a deep review in the literature, there are no either commercial or non-commercial solutions which solve these drawbacks, simplifying the MAS measurement and analysis with the proper tools ensuring the automation of the process, reducing time required and the error promptness.

With this premise, a novel solution is proposed in this paper, concluding with a tool to automatize and simplify the MAS measurement, called FLEXOR. This tool implements a novel mathematical and optimization procedure (describe in Section 4) and required a well-tested and patented machine specially designed and built for this purpose (see Figure 2).

Fig. 2 - Mechanical device for measurement

3. Computational methods and theory

This section contains a description of the theoretical fundamentals of our work. Here we describe the assumptions and mathematical bases required to carry out MAS analysis procedures and which are supported by the FLEXOR tool.

Taking as a reference an equilibrium static position, a perturbation induced in the lower part of the leg produces an involuntary free oscillation of the foot around the ankle. This vibration, in which the plantar flexor muscles are involved (mainly the Triceps Surae), can be extremely well modelled as a one-degree-of-freedom damped system. To measure the mechanical parameters (i.e., stiffness and viscosity) of the MTU, a mechanical device was conceived and manufactured, as well as software intended to easily capture, process and analyse the mechanical signals of tests on the legs and allow these mechanical parameters to be determined. The details regarding how the measurement device was manufactured and validated and the procedure to make the measurements can be found in [24]. The device and the procedure gave rise to the patent ES 2 377 640 B1. Figure 2 shows several snapshots of the measurement device from different angles (left-hand side and middle of the Figure) and a snapshot during a test with a study subject (right-hand side of Figure 2).

In Figure 3, a schematic representation of the mechanical device (Figure 3.a) is depicted, together with a physical model and schematic representation of the damped one-degree-of-freedom system that the foot and Triceps surae form (Figure 3.b).

M represents the mass placed over the knee, *k* and c the Triceps surae stiffness and viscosity, respectively, and r and R the rearfoot and forefoot arm lengths, respectively.

The test procedure gives rise to a direct recording of force vs. time. To extract the MAS parameters from this time register, two different adjustment processes are required.

The measurement of the mechanical properties of the plantar flexor muscles takes place through several steps. In this section, the different tasks and calculations involved are described.
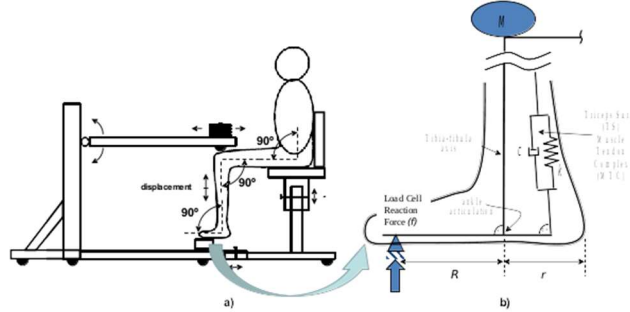

Fig. 3. a) - Scheme of the measurement mechanical device. b) Scheme (detail) of the foot

A force versus time experimental record (see Figure 4) is obtained by using the device introduced schematically in Figure 3 as the first stage of the procedure. The curve represents the force variation, which only takes approximately one second, due to the oscillation of the lower limb produced by the impact on the knee. See [24] for further details of the procedure.

A force versus time experimental record (see Figure 4) is obtained by using the device introduced schematically in Figure 3 as the first stage of the procedure. The curve represents the force variation, which only takes approximately one second, due to the oscillation of the lower limb produced by the impact on the knee. See [24] for further details of the procedure.
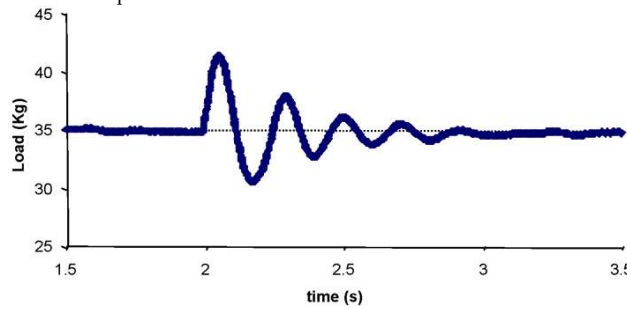

Fig. 4. Force-vs-time experimental record

This free oscillation behaviour can be extremely well captured by using a one-degree-of-freedom damped system. The modelled force-vs.-time response has the form shown in (1).

(1)
$$F_m = e^{-\gamma t}(A_F sin\omega_D t + B_F cos\omega_D t) + Mg$$

Equation (1) has five parameters, all of them having a clear physical meaning. ω_D is the oscillation frequency, γ is the damping parameter, *M* is the oscillating mass, *A* and *B* are coefficients associated with the oscillation amplitude, and g is the acceleration due to gravity. The best set of values for (ω, γ, M, A, B) are obtained by a least square fitting method, minimizing the sum of quadratic differences between the experimental data and (1) [24,25]. It is necessary to take into account two important facts that make this first step of the calculation tedious when done manually:

1. The determination of five parameters is associated with the short period of time that holds the oscillation after the impact (between 1 and 1.5 seconds). Depending on the characteristics of the impact and the reflex response of the subject, the initial part of the experimental curve has to be discarded, and the last part of the oscillation is of low quality because the "noise" of the signal equals the value of the oscillation.
2. The longer the good-quality experimental data, the better, but this selection of the set of experimental data to be used is not a straightforward task to be performed automatically.

For each test, using the five calculated parameters (ω, γ, M, A, B) and the anthropometric information of the forefoot and rearfoot distances (R and r, respectively), a new pair of processed data, namely, f and k, are obtained. f represents the force registered at the load cell, which is correlated with the force passing along the Achilles tendon, and k is the stiffness of the MTU. As the complete experimental procedure requires different loads to evaluate the contribution of the different elements included in the MTU (tendon and muscle), the number of tests increases up to approximately thirty or forty tests for each leg (e.g., four tests per load and 8 loads). All of the tests for the same load will give rise to pairs of (*f, k*), which should not be equal and instead should have some dispersion.
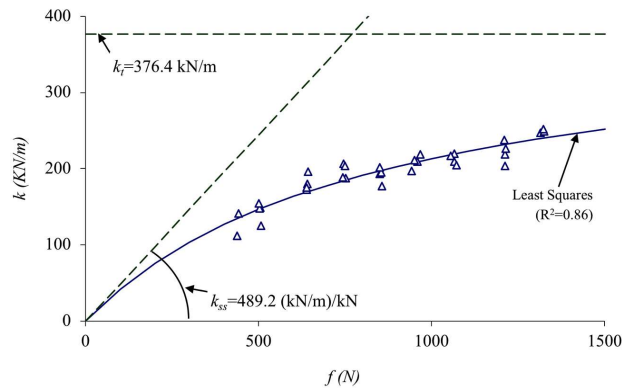
Fig. 5. Results obtained from the 2nd step of the adjustment process.

When the load changes, f and k change accordingly, f because it has a direct relationship with the load and k because of the nature of muscles. Thus, after the 30 or 40 tests for each leg, a set of points (dots) will appear (see Fig 5).
Again, according to Hill's model [26], this cloud of points can be well fitted by means of the following expression, shown (as the solid line) in Figure 5:

$$(2) \qquad k = \frac{k_t k_{ss} f}{k_t + k_{ss} f}$$

Where k_{ss} is the unitary stiffness (stiffness per unit load) of the soleus and kt is the stiffness of the Achilles tendon. While k_t can be reasonably constant irrespective of the force passing along it, k_{ss} depends on the value of the force. That is the reason why, as the force increases, the value of the global k changes (see Figure 5).
It is important to understand that the individual constituents (k_{ss} and k_t) of the global stiffness (k) do not change with the applied load, but the non-linearity of equation 2 makes k to change when the load f changes.
This issue is the key aspect which allows to evaluate the stiffness of each constituent from the global measurements of the Triceps Surae stiffness (k).

Once again, the best determination of k_t and k_{ss} can be achieved by a least square fitting method by using (2) with the experimental data (see Figure 5). When fitting the experimental data, k_{ss} and k_t take a physical meaning: k_{ss} represents, from a mathematical point of view, the slope of the curve for f→0, and k_t represents the asymptote of the curve for f→∞. Although k_{ss} and k_t are defined, from a mathematical point of view, in extreme situations (f→0 in the case of k_{ss} and f→∞ in the case of k_t), the physical meaning of both parameters is quite clear.

After this second fitting step, which has to be performed only once for the complete set of tests (each test being a curve such as the one depicted in Figure 4), the values of k_t and k_{ss} are obtained.

4. Software Design

A description of the software is presented in this section. Here, the software architecture as well as the mathematical methods implemented are discussed, the user interface is showed, and the functionality of the software is described.

4.1. Motivation

The procedure described above to measure the MAS of the plantar flexor muscles is carried out through different measuring, processing and dissemination third-party software from different vendors (e.g., LabVIEW, Mathematica, Microsoft Excel), where different technicians may be involved. The use of a unique software simplifies the whole procedure. The use of several software applications markedly increases the time required to obtain a ready-to-use sample (the first-phase outcome mentioned in Section 3), where the intermediate steps between software applications should be processed manually to ensure that the output of a software works as an input for the next software.

The first-phase outcome files should be collected as input for the second phase mentioned in Section 3. In addition to the time required to process and obtain the set of ready-to-use samples, these should be organized manually, which involves checking the order of the sample, subject, weight applied, etc. and thus further complicating the data management to obtain the results of the study.

In this section, we introduce FLEXOR, a Java-coded software application intended to assist in the analysis of MAS from a validated measurement device specifically devised for such purpose. FLEXOR handles the capture of the signal received from a load cell placed in the measurement device and transform it for its subsequent processing through different numerical methods. FLEXOR aims to support the method presented in Section 3, addressing the drawbacks related to the use of several programs and simplifying a four-step process into an all-in-one, seamless and transparent procedure.

4.2. Architecture of FLEXOR

FLEXOR consists of several software components that address different concerns and functionalities [27]. To describe the main components and functionalities of the FLEXOR architecture, we will make use of the *4+1 view model* [28]. The *4+1 model* describes the software design using four views

to represent its design (i.e., logical, process, physical and development views) and a fifth view, called the scenario(s) view, intended to illustrate and validate the software (see Figure 6). The fifth view, i.e., the 'Scenario' view, will be presented in Section 5 'Application Example' to illustrate the main results provided by the FLEXOR software.

The *4+1 model* is preferred in this work to describe FLEXOR because it enables software engineers and developers to learn the system from different practical points of view, linking design with implementation [29]. In this sense, FLEXOR could be reutilized to solve similar problems, and the algorithms designed could be adapted, implemented and applied in other systems designed for similar purposes.
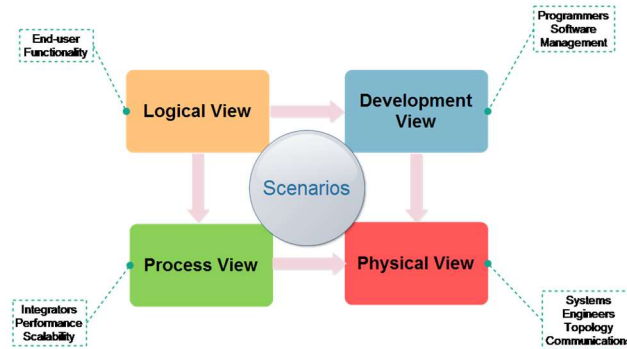

Fig. 6. The 4+1 view model

4.3. Logical View

The logical view captures the functionalities provided by the software application to its end-users (e.g., sport-medicine practitioners, therapists, doctors, etc.). Most FLEXOR functionalities are served or activated from its graphical user interface (GUI). It can be observed in the snapshot of the main FLEXOR GUI when the application is started, which consists of a set of control panels (see Figure 7). From these control panels, it is possible to manage single experiments with one subject and groups of them, check the load cell signal or apply the numerical calculations already described in Section 3 (phases one and two).

Most of the main functionalities of FLEXOR have been conceived around the concepts of *sample* and *project*. A *sample* consists of the data set received from the load cell placed in the measurement device after a perturbation has been provoked in the plantar flexor muscles of the subject under study, together with some other information related to start and end times of the experiment, weights used when provoking the perturbation and the leg under study (i.e., the left leg or right leg). The data set received from the load cell corresponds to set of force-time points sensed during the experiment after provoking the perturbation, and on such a data set is to be performed the first adjustment phase presented in Section 3. All information and data regarding each sample are represented using JSON (JavaScript Object Notation), an open-standard, platform-independent file format, which permits interoperability with third-party applications [30].

Alternately, a *project* is a data structure that encompasses a set of samples and allows one to segment and group them as the analyst might require. In addition, each project has a name, a date, the list of samples (self-structured) and the parameters needed for the second adjustment. Projects are also represented in JSON.
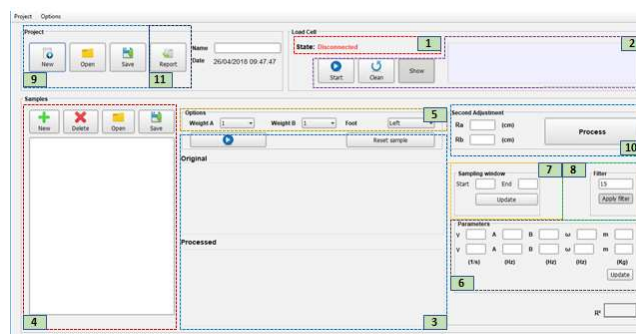

Fig. 7. FLEXOR GUI

Figure 7 shows the FLEXOR GUI, where the main functionalities are marked and described as follows:

1. *Load-cell (measurement device) connection*: Detects, automatically, if the load-cell is connected to (or disconnected from) the computer where FLEXOR is running.
2. *Data streaming*: Capture data streams from the experimentation measurement device in real-time, showing a line chart with information regarding forces and times provided by the load-cell placed in the machine. This chart could be used by the analysts to check if the load cell reacts properly to the interaction with the subject of the study. Furthermore, the reception of data streams can be stopped at any time.
3. *MAS sampling*: Using a start-stop option, the analyst is able to capture a specific set of values of a specific interval of time, that is, a sample. This sample is shown on a different force-timeline chart.

4. *Sample management*: The software enables the ability to capture the data stream from the load cell in real-time in different samples, adding new samples at any time or removing previously created ones. Additionally, it is possible to export and import a sample with a JSON format using a custom structure stored in a file with the extension *fxs* ("FLEXOR Sample"). This enables the interoperability between studies or the ability to conduct a study with a subject at a deferred period of time, collecting independent samples.
5. *Sample options management*: Each study concerns a specific leg (left or right) to perform the different activities with different weights according to the instructions of the analyst. These parameters can be reflected and associated in each sample.
6. *Phase-one calculation*: The software calculates, automatically, phase one as presented in Section 3, obtaining the five parameters (ω, γ, M, A, B). However, the analyst can calculate phase one using custom values for the parameters in the panel instead of the automatic values calculated.
7. *Create sub-samples (sampling window)*: The analyst can select sub-samples of a sample to apply the phase-one mathematical procedure to a specific part of the data captured.
8. *Filter-sample*: Apply a low-pass filter to clean the original sample or get the data in a raw form.
9. *Project management*: At any time, the analyst is able to save all of the study, saving all of the samples under a unique project in an external file using JSON notation with a specific format and stored in an *fxp* file (after "FLEXOR Project"), enabling the ability to reconstruct or continue the analysis at any time.
10. *Phase-two calculation*: Once different samples have been recorded, the Moment arm of the Achilles tendon (r) and the moment arm of the second metatarsal head (R) values are introduced, and the software computes the phase-two procedure as described in Section 3.
11. *Report generation*: The software is able to generate a PDF file with all the information regarding the study: personal and physiological information regarding the subject (name, weight, photo, age, etc.), information regarding the study (name, date, number of samples) and information related with phase one and two, that is, a table with all the results of phase one and the information and chart with phase-two results.

4.4. Process View

This view addresses the processes of the system and how they communicate, that is, the different steps to conduct an experiment using the FLEXOR software. The next figure shows a flow diagram that shows the main steps in the use of the system (see Figure 8). The procedure to conduct a study through FLEXOR is a sequential set of steps, starting with the setup of a new project and concluding with the result of the study as a report. During the process, the analyst guides the participant with several orders and, through FLEXOR, obtains the required values to determine the MAS.
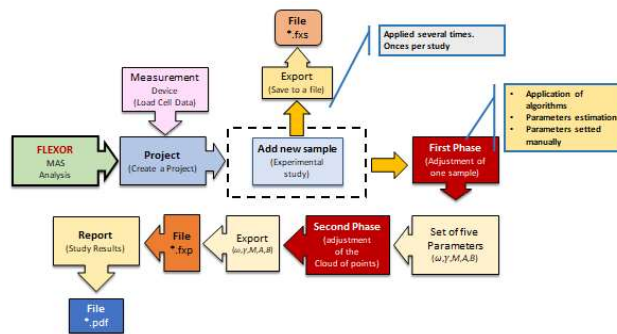

Fig. 8. Flow diagram to illustrate a study conducted in FLEXOR

These values are obtained using the mathematical procedures described in Section 3; however, in FLEXOR, a specific solution has been implemented, and which is described as follows.

In the original procedure described in Section 3, one of the main steps is the application of a least square approach to approximate a solution to (1) to determine the value of the five parameters: ω, γ, M, A and B (1). This was done using *Mathematica*, a symbolic computation program. The same mathematical procedure was attempted in the implementation, testing two different approaches supported by open-source libraries: Nelder-Mead [31] and BOBYQA [32]. However, the application of both algorithms to estimate the value of the five parameters did not work properly, obtaining a poor estimation of the values, which have an important margin of error, compared with the values calculated using *Mathematica*. Because neither supported method provided a reliable solution and the full implementation of a numerical optimization algorithm required great effort, an algorithm to address this issue and achieve the estimation of the five parameters was designed and implemented. The analytic function algorithm procedure, which also contains new algorithms, is designed as follows:

1. *Smooth the original sample using an infinite impulse response filter (IIR)*. Using a low-pass filter (or IIR filter), each point of the sample, namely v_i, is processed to leave out the noise, applying a smooth procedure, Eq. (3). The superscripts "new" and "old" represent the filtered signal and the non-filtered one, respectively. $s \geq 1$ is a parameter for the smoothing of the signal, if $s=1$ no filtering is applies and if $s \to \infty$ the filtered signal is a constant equal to the first value of the series. In the implementation, the smooth value s is predefined but can be changed through the FLEXOR GUI. The following diagram (see Figure 9) represents the implemented algorithm to smooth a sample.

(3)
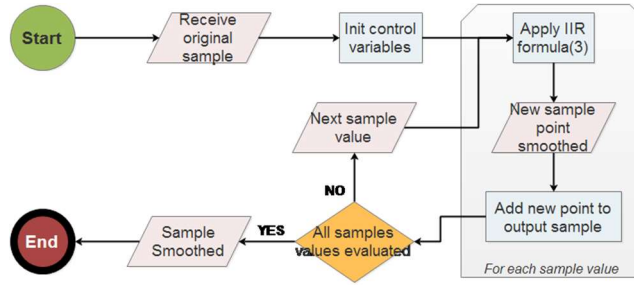$$v_i^{new} = v_{i-1}^{new} + \frac{v_i^{old} - v_{i-1}^{new}}{s}$$

Fig. 9. Algorithm to smooth a sample

2. *Crests detection or peak finding*. Once the sample is smoothed, the next step is to find the peaks (or crests) of the wave/sample (the sample is represented as a sinusoidal wave; see Figure 4). The custom algorithm implemented to detect the peaks of the wave/sample uses an upper and lower threshold-based method, with static values because the load cell has a static frequency. The algorithm is illustrated in the Figure 10.
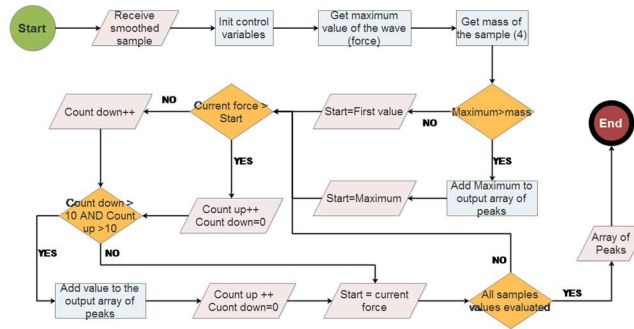

Fig. 10. Upper and lower threshold-based method

3. *Obtain a sub-sample (sub-wave)*. With the peaks detected, we obtain a subsample of the original sample starting a half-second before the first peak and ending a half-second after the last peak. In this case, we take 500 sample points, because the sampling rate is 1 kHz. The points considered before and after are necessary to have the full wave because the peaks are, usually, in the middle of the wave.

4. *Evaluate the average force (mass) (F_0)*, where *a* is the position (in the force *vs* time file) of the first value and *b* the position of the last value. That is, the total number of values of the sample (4).

$$(4) \qquad\qquad F_0 = \frac{\sum_{i=a}^{b} f_i}{(b-a)}$$

Calculate the frequency of the subsample (ω) using (5), where *T* is the oscillation period:

$$(5) \qquad\qquad \omega = \frac{2\pi}{T}$$

5. *Calculate the damping of the subsample (γ) with equation (6)*, where F_{max} represents the peaks of the oscillation shown in Figure 4, F_0 is the average value, and *T* is the oscillation period shown in Figure 4. The reduction of the peak values in one oscillation allows to estimate the damping coefficient *γ*. The peaks can be easily detected in an automatically way.

$$(6) \qquad\qquad \gamma = \frac{ln(\frac{F_{max_i}-F_0}{F_{max_{i+1}}-F_0})}{T}$$

6. *Nelder-Mead*. Using the three calculated parameters (F_0, *ω* and *γ*), the software estimates the two missing parameters (amplitudes *A* and *B*) using the Nelder-Mead optimization algorithm.

As a result, we obtain the two amplitude parameters, getting the complete function and making the adjustment (see Figure 11).
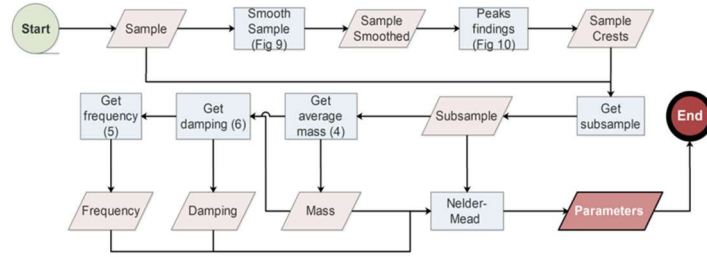
Fig. 11. First-phase algorithm implemented in FLEXOR

This algorithm is performed for each sample, obtaining the five parameters. To carry out the second phase, with the five parameters and according to the procedure described in Section 3, we obtain the values for *F* (overall force), *K* (overall stiffness), *f* (force along the MAS) and *k* (MAS stiffness), using the values *R* and *r* (forefoot and rearfoot moment arms), fixed manually by the analyst and using the following formulas (7-10):

$$F = m \cdot g \qquad (7)$$

Where *F* is the overall force, *m* represents the weight of the whole oscillating system, which includes the lower limb and the extra weight added on the knee and *g* the acceleration of gravity.

$$K = m(\omega^2 - \gamma^2) \qquad (8)$$

Where *K* is the overall stiffness (whole oscillating system) and ω and γ are respectively, the oscillation frequency and the damping coefficient (equation 1). See [33] for further information.

$$f = F\left(\frac{R}{r}\right) \qquad (9)$$

Where *R* and *r* are the moment arm lengths of the foot (Figure 3b).

$$k = K\left(\frac{R}{r}\right)^2 \qquad (10)$$

The aim is to obtain, for each sample, the pair (*f,k*), obtaining a set of points. These sets of points are used to estimate an approximation to (1), applying the Nelder-Mead algorithm and getting the values K_{ss} and K_t, which are the values required to estimate the MAS.

With K_{ss} and K_t, the set of points (*f,k*) and (2), we are able to represent the chart as a result of the study. The diagram of that represent the procedure of the second phase is shown in Figure 12.
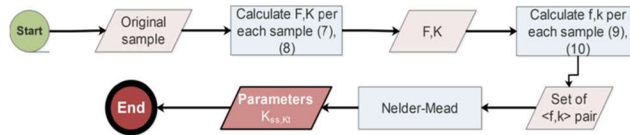

Fig. 12. Diagram of the Second-phase algorithm

4.5. Physical View

From a system engineer's point of view, the software system is composed by:
1. *Experimentation measurement device (Load-cell)*: Described before, the experimentation measurement device (where the load cell is hosted) is used to perform the different studies with the subjects or participants. The machine is used as a data generator from the actions performed by the subjects during the study.
2. *Software (FLEXOR)*: A cross-platform application, described across this section, with the aim of allowing integral management of the procedure before, supporting from the data capture of the load cell to the generation of the final report with the information of the subject, that is, the results of each study (first phase) as well as the final result (second phase).
3. *Application Server*: Computer where FLEXOR is running and the machine is plugged in through a serial port (USB). The next figure provides an overview of the improved MAS analysis setting that FLEXOR enables (see Figure 13).
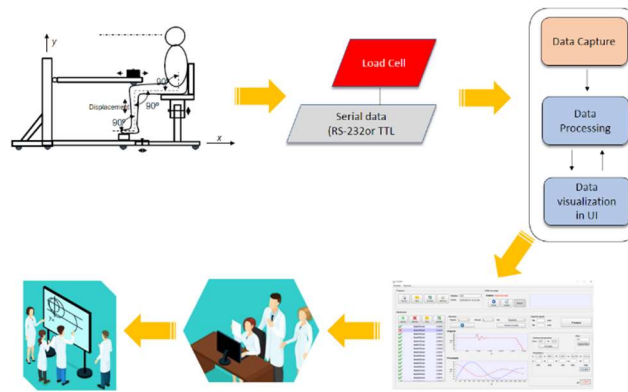
Fig. 13. Overview of the MAS analysis process enabled by FLEXOR

4.6. Development view

The development view illustrates the view from a software perspective through the different devices and artefacts to help programmers or software engineers understand the internal software structure. FLEXOR has been developed following a component-based software development (CBSD) approach [34], where each software component focuses on a specific concern of the system, such as samples, projects, communication with the machine and data dissemination, among others. Following a CBSD approach, it is possible to reuse a software component in a different system or add new functionalities to a specific software without affecting the rest of the software components, enhancing not just the extensibility and reusability but also the maintenance of the system.

The use of CBSD to design and implement a system as independent modules is very common in software engineering projects, due to its features such as re-usability, easy to adoption, encapsulation, etc. [35-39].

The next figure shows the software components of FLEXOR, deployed in an application server, which is connected to the experiment artefact (see Figure 14).
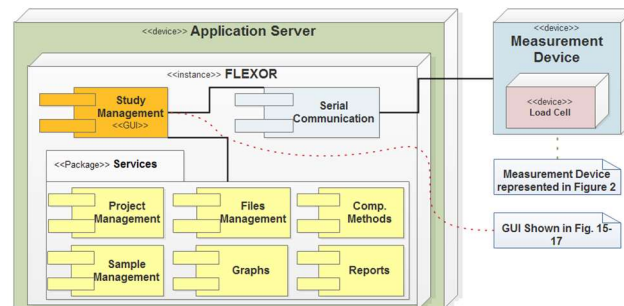

Fig. 14. Software components of FLEXOR

The title of each software component describes the functionalities supported related to a specific concern required to provide the different FLEXOR functionalities. The description of each software component is as follows:

1. *Serial Communication.* The interaction between the load cell (hardware) and the software application is a key requirement in this scenario. This software component is in charge of managing the connection with the load cell, collecting the data provided, and handling this information to be provided to the other software components in a reliable manner.
2. *Graphs.* This software component provides the functionalities related with the dissemination of the data collected in real-time from the machine, as well as the data processed according to the different procedures implemented to achieve the process described in Section 3. This dissemination can be used to check if the load cell is sending information properly or if the procedures are consistent with the expected outcomes.
3. *Files Management.* The software enables one to represent a full study or part of it as data structures (in projects and samples), which can be stored in independent files. This software component manages these files, allowing one to save projects and samples as files and load projects and files from files, in JSON format.
4. *Computational Methods.* This software component provides the different mathematical aspects required to implement the procedure described before, such as mathematical formulas. In addition, the different algorithms are implemented in this software component, such as first and second-phase related algorithms or the algorithms and the upper and lower threshold-based method (e.g. shown in Figure 9-12).
5. *Reports.* The generation of a final report, in PDF format, is the outcome of an experiment to evaluate the study and conclude the MAS. This component allows the creation of the reports using the result of the study, as well as personal and physical information of the subject of the study.
6. *Samples Management.* The samples are managed by this software component to create a new one, import/export from/to files, delete an existing one, apply the first phase adjustment (using the algorithms component), etc.
7. *Project Management.* This software component provides the functionalities related to the sequential procedure described in Figure 8, namely: create a new project, create several samples, run the first and second phase according to Section 3, etc.

5. Application Example

This section presents the results for a particular hands-on case (fifth view or scenarios view), illustrating the implementation of the previous views. In addition, a full trial conducted using FLEXOR with one subject is described in this section.

To show the main functionalities of the FLEXOR system, three different cases have been selected to display the most significant part of the studies during one project carried out with one subject using the entire range of loads and for the left leg (the range of loads used depends on the characteristics of the subject).

Fig. 15. Automatic first-phase calculation

Figure 15 shows an example of the first-phase fitting (the red curve is the experimental data, and the blue line the representation of (1)). The load range used (0-50 kg) in one project is displayed on the left side of the screenshot. All of them have the approval criterion (green check symbol) of the software, which means that the fitting parameter R^2>0.9. The details of the window range adjustment and the details of the parameters used are shown on the right side of the screenshot. In the present case, all tests carried out give rise to excellent adjustments.

A first improvement of the program consists of making the automatic estimations of the three parameters (ω, γ and M) using the recorded data, and second, automatically select the optimum interval of the experimental data by evaluating the fitting quality by means of R^2 values. As the perturbation caused by the impact is easy to identify, the program automatically discards the very first experimental points and makes different interval trials in the oscillation time.

As an example of the potential of the software, the analyst can change manually one or more parameters of the adjustment (see Figure 16, where a parameter has been forced to take an incorrect value and the fitting curve does not fit the experimental data). When this occurs, the approval criterion cannot be established (the R^2 value is lower than 0.9), as shown in the last row of the left-hand side of the figure. In some cases, when the experimental data have not enough quality, the analyst can manually modify any of the five parameters or even the window of adjustment to obtain the best result (R^2 values close to the unit).
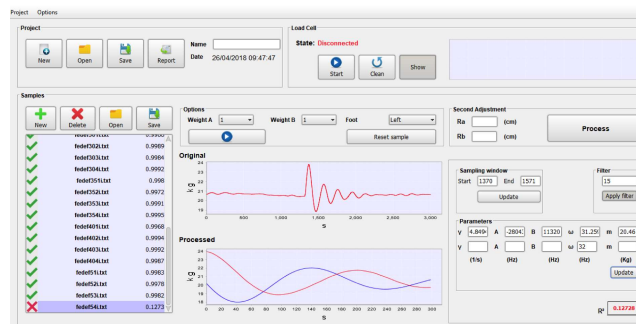

Fig. 16. Result of changing one or more parameters of the first phase of adjustment

Finally, the second phase of the adjustment is displayed (see Figure 17). In the smaller window, the cloud of points generated from the first phase of the adjustment can be displayed. The second phase of the adjustment depends on two parameters: contribution of the Achilles Tendon (green line) as a constant value and the contribution of the Soleus (orange line).
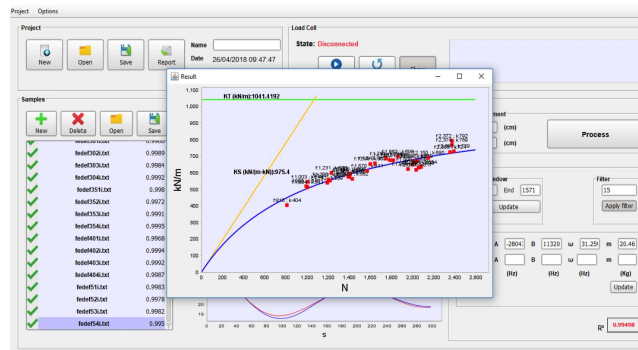
Fig. 17. Second phase of the adjustment

6. Discussion

FLEXOR obtains the mechanical response of the muscle-tendon units using the free vibration technique of a one-degree-of-freedom damped system linked to the ankle joint. This damped vibration measured by a load cell hosted in a measurement ad hoc device (where different degrees of freedom related to the position of the subject in the measurement device must be controlled carefully) is graphically displayed in a wave shape in the FLEXOR GUI. In addition, through the integrated GUI of FLEXOR, the analyst is able to conduct a full study to estimate MAS parameters, manage several studies under the same subject, adjust on-the-fly estimated results, generate deliverable results, etc.

In the present paper, FLEXOR software has been introduced as a single Java software application for MAS measurement and as an analysis tool. Before FLEXOR was developed, the acquisition of experimental data was performed by means of one tool developed in LabVIEW®. The least squares method was conducted for each step using another code based on Mathematica®, and the final statistics and report were done with an Excel® spreadsheet. So far, MAS analysis procedures were alike and made use of these or similar tools for each analysis stage. All these factors greatly complicated the analysis of the data, making this procedure not operational for clinical or sports applications.

FLEXOR has a flexible design so as to be adaptable to future requirements, adding new functionalities through a software component-based approach, as well as providing the information generated in different studies in an interoperable way using JSON for datasets representations. This enables multi-stakeholder scenarios, where different types of analysts with different FLEXOR installations (e.g., different application servers, platforms and technologies) are able to conduct joint studies in a remote manner, just exchanging FLEXOR files (i.e., *.fxs and *.fxp).

So far, several studies have been conducted in the lab to analyse the reproducibility of the results compared to the one used in previous procedures and the results obtained by FLEXOR software. To carry out this comparison, load-cell raw data (files) from previous studies conducted using the former procedure have been used. Each file contains the list of points (force, time) of data that comes from the load cell. When the file is imported into FLEXOR, the software automatically parses the file into a sample and calculates the first-phase adjustment. Importing the list of files that corresponds to a project obtained from one subject, the same study can be simulated (and the procedure) through the FLEXOR software. This approach permits to validate that FLEXOR works in a proper manner, checking if the results are equal to the previous studies conducted. In summary, the implemented code allows the MAS estimation with the followings highlights:

1. Process the complete set of experimental record files (between 30 and 40 typically) in a few mouse clicks.
2. Make the first optimum least squares adjustment (by evaluating the R^2 parameter in different time intervals of the record) for each record and obtaining the pair (f,k).
3. Carry out the second least squares adjustment of the (f,k) values to finally obtain the values of k_{ss} and k_t.
4. Generate a complete report with all statistical values, plots, experimental data giving rise to bad values (outliers), etc.

Finally, FLEXOR has been developed in a flexible manner, using component-based software development (CBSD) where different software components have been developed to support the different functionalities required. In addition, FLEXOR handles experimental data using custom and flexible data structures, organized into samples and projects, which are represented in a language-independent and lightweight data-interchange format, allowing to be used by third-party software and enabling the conduction of shared studies between different analysts who use FLEXOR software.

7. Conclusions and further work

So far, MAS analysis used to consist of a complex, trial-and-error-based estimation procedure involving several stages and software applications (such as device interfaces, numerical computing environments and spreadsheet editors), defining a flow of tasks whose interlinkage had to be managed by hand.
Through a friendly GUI-based solution, FLEXOR supports and further automates analysis of MAS, using suitable hardware (experimentation measurement device load cell), addressing the different steps required for signal processing, optimizing data processing and simplifying a traditionally four-step process down to a one-step process, enabling the automatic estimation of MAS analysis and reducing, significantly, the time required to carry out a complete study from one or two hours to few minutes.
Improvements of FLEXOR include estimations of parameters (ω, γ and M) and subsequent optimum selection of raw experimental data. Perturbations originated by the impact are automatically discarded by the software to improve data to be postprocessed.
Through FLEXOR, the analysts manage the entire MAS analysis, adding as many samples (or studies) as required, allowing the adjustment of the different variables or parameters in each study and on-the-fly, import/export studies, checking the information received from the load cell, etc. FLEXOR has been designed to support these functionalities using a one-click approach to simplify their use and improve the usability. FLEXOR handles the different studies using custom and well-defined data structures, which are stored as files, giving rise to multistakeholder collaborations where different analysts in different places and studies/projects can share FLEXOR files to conduct shared studies or exchange parts of studies. In addition,

FLEXOR files are implemented using an interoperable notation to allow third-party applications to manage MAS analysis studies and enabling programmers/developers to support the MAS analysis procedure in FLEXOR format in their own applications. Several studies have been carried out in the lab, with study subjects and with files generated from former studies, to check the reproducibility of the results, showing the same outcomes as those obtained using the trial-and-error-based estimation. Nonetheless, the conduction of new studies is mandatory to test the acceptance and usability of FLEXOR.

The simplicity of the mathematical functions used in the whole procedure (least squares fitting) makes no difference in accuracy between the previous proposal and FLEXOR.

As for future work, field studies must be conducted with a representative sample of users (gender, age, weight, size) to validate the reliability of FLEXOR in the results obtained, its suitability as a software application and the enhancement of the MAS analysis (reducing the time required, improving user's comfort, etc.).

Furthermore, the improving of FLEXOR enables the conduction of shared studies through cloud technology without the need to exchange files, where analysts are able to use FLEXOR loading studies from a cloud-supported repository or database in real-time, anytime and anywhere.

Future improvements of FLEXOR will target the use of this software and adapting it to commonly used technologies, such as mobile computing or web applications. New functionalities will be implemented (e.g., a third-party API) to use FLEXOR through custom mobile applications or web platforms, enabling the conduction of shared studies in multistakeholder scenarios, in a remote manner and in real-time.

Competing interests

The authors declare that there are not competing interests.

Supplementary Material

Supplementary materials (the FLEXOR tool, installation guide, and some data examples) can be found online at https://github.com/FlexorSoftware/flexor.

Funding

REFERENCES

[1] F. Paris-Garcia, A. Barroso, and F. Paris, "A study of the reproducibility and reliability of the musculo-articular stiffness of the ankle joint," J. Mech. Behav. Biomed. Mater., vol. 82, pp. 145–153, Jun. 2018.

[2] J. Babic and J. Lenarcic, "In vivo determination of triceps surae muscle-tendon complex viscoelastic properties," Eur. J. Appl. Physiol., vol. 92, no. 4–5, pp. 477–484, 2004.

[3] S. Fukashiro, M. Noda, and a Shibayama, "In vivo determination of muscle viscoelasticity in the human leg.," Acta Physiol. Scand., vol. 172, no. 4, pp. 241–8, Aug. 2001.

[4] K. a McLachlan, A. J. Murphy, M. L. Watsford, and S. Rees, "The interday reliability of leg and ankle musculotendinous stiffness measures.," J. Appl. Biomech., vol. 22, no. 4, pp. 296–304, Nov. 2006.

[5] M. Ditroilo, M. Watsford, A. Murphy, and G. De Vito, "Assessing musculo-articular stiffness using free oscillations: theory, measurement and analysis.," Sports Med., vol. 41, no. 12, pp. 1019–32, Dec. 2011.

[6] Blackburn J.T., Padua D.A., Riemann B.L., et al. The relationships between active extensibility, and passive and active stiffness of the knee flexors. J Electromyogr Kinesiol, vol. 14 no. 6, pp. 683-91. 2004.

[7] Granata K.P., Wilson S.E., Massimini A.K. et al. Active stiffness of the ankle in response to inertial and elastic loads. J Electromyogr Kinesiol, vol. 4, no. 5, pp. 599-609. 2004.

[8] Blackburn J.T., Padua D.A., Guskiewicz K.M. Muscle stiffness and spinal stretch reflex sensitivity in the triceps surae. J AthI Train 2008, vol. 43 no. 1, pp. 29-36. 2008.

[9] A. Faria, R. Gabriel, J. Abrantes, R. Brás, and H. Moreira, "Triceps-surae musculotendinous stiffness: relative differences between obese and non-obese postmenopausal women.," Clin. Biomech. (Bristol, Avon), vol. 24, no. 10, pp. 866–71, Dec. 2009.

[10] A. Faria, R. Gabriel, J. Abrantes, R. Brás, and H. Moreira, "The relationship of body mass index, age and triceps-surae musculotendinous stiffness with the foot arch structure of postmenopausal women.," Clin. Biomech. (Bristol, Avon), vol. 25, no. 6, pp. 588–93, Jul. 2010.

[11] D. Wang, G. De Vito, M. Ditroilo, D. T. P. Fong, and E. Delahunt, "A comparison of muscle stiffness and musculoarticular stiffness of the knee joint in young athletic males and females," J. Electromyogr. Kinesiol., vol. 25, no. 3, pp. 495–500, 2015.

[12] M.-B. Ibáñez, S. Escolar, R. Iskandar, K. Viera, and C. Delgado-Kloos, "Integrating Biomedical Sensor Data into a Simulation Learning Environment for Children Newly Diagnosed with Diabetes," J. Univers. Comput. Sci., vol. 23, no. 6, pp. 539–559, Jun. 2017.

[13] De Souza, J. K. S., da Silva Pinto, M. A., Vieira, P. G., Baron, J., & Tierra-Criollo, C. J. "An open-source, FireWire camera-based, Labview-controlled image acquisition system for automated, dynamic pupillometry and blink detection". Computer methods and programs in biomedicine, 112(3), 607-623. 2013.

[14] J. H. Horng, "Hybrid MATLAB and LabVIEW with neural network to implement a SCADA system of AC servo motor," Adv. Eng. Softw., vol. 39, no. 3, pp. 149–155, 2008.

[15] A. Y. Chen, T. Y. Lu, M. H. M. Ma, and W. Z. Sun, "Demand Forecast Using Data Analytics for the Preallocation of Ambulances," IEEE J. Biomed. Heal. Informatics, vol. 20, no. 4, pp. 1178–1187, Jul. 2016.

[16] López-Fernández, H., Araújo, J. E., Jorge, S., Glez-Peña, D., Reboiro-Jato, M., Santos, H. M., ... & Capelo, J. L. "S2P: A software tool to quickly carry out reproducible biomedical research projects involving 2D-gel and MALDI-TOF MS protein data." Computer methods and programs in biomedicine, 155, pp. 1-9. 2018.

[17] Blackburn, J. T., Padua, D. A., Weinhold, P. S., & Guskiewicz, K. M. "Comparison of triceps surae structural stiffness and material modulus across sex". Clinical Biomechanics, 21(2), 159-167.2006.

[18] McLachlan, K. A., Murphy, A. J., Watsford, M. L., & Rees, S. "The interday reliability of leg and ankle musculotendinous stiffness measures". Journal of Applied Biomechanics, 22(4), 296-304.2006.

[19] Pruyn, E. C., Watsford, M. L., & Murphy, A. J. "Validity and reliability of three methods of stiffness assessment". Journal of Sport and Health Science, 5(4), 476-483.2016.

[19] Fouré, A., Cornu, C., McNair, P. J., & Nordez, A. "Gender differences in both active and passive parts of the plantar flexors series elastic component stiffness and geometrical parameters of the muscle–tendon complex". Journal of Orthopaedic Research, 30(5), 707-712.2012

[20] Perreault, E.J., Kirsch, R, F, Acosta, A. M. "Multiple-input, multiple-output system identification for characterization of limb stiffness dynamics". Biol. Cybern. 80, pp. 327-337. 1999.

[21] Faria, A., Gabriel, R., Moreira, H., Camacho, T., Brás, R., & Ditroilo, M. "The effect of sex and localised fatigue on triceps surae musculoarticular stiffness". European journal of sport science, 18(4), 483-490. 2018

[22] Waxman, J. P., Schmitz, R. J., & Shultz, S. J. "The interday measurement consistency of and relationships between hamstring and leg musculo-articular stiffness". Journal of applied biomechanics, 31(5), 340-348.2015

[23] Morales-Artacho, A. J., Lacourpaille, L., & Guilhem, G. "Effects of warm-up on hamstring muscles stiffness: Cycling vs foam rolling". Scandinavian journal of medicine & science in sports, 27(12), 1959-1969.201

[24] F. París-García, A. Barroso, J. Cañas, J. Ribas, and F. París, "A critical study on the experimental determination of stiffness and viscosity of the human triceps surae by free vibration methods.," Proc. Inst. Mech. Eng. H., vol. 227, no. 9, pp. 935–54, Sep. 2013.

[25] F. Paris-Garcia, a. Barroso, M. Doblare, J. Canas, and F. Paris, "Evaluation of the stiffnesses of the Achilles tendon and soleus from the apparent stiffness of the triceps surae," Proc. Inst. Mech. Eng. Part H J. Eng. Med., vol. 229, no. 1, pp. 28–39, 2015.

[26] A. V Hill, "The heat of shortening and the dynamic constants of muscle," Proc. R. Soc. London. Ser. B-Biological Sci., vol. 126, no. 843, p. 136, 1938.

[27] M. W. Maier, D. Emery, and R. Hilliard, "Software architecture: Introducing IEEE standard 1471," Computer (Long. Beach. Calif)., vol. 34, no. 4, pp. 107–109, 2001.

[28] P. B. Kruchten, "The 4+1 View Model of Architecture," IEEE Softw., vol. 12, no. 6, pp. 42–50, 1995.

[29] C. Yang, P. Liang, and P. Avgeriou, "A systematic mapping study on the combination of software architecture and agile development," J. Syst. Softw., 2016.

[30] H. Honko, V. Andalibi, T. Aaltonen, J. Parak, M. Saaranen, J. Viik, and I. Korhonen, "W2E-Wellness Warehouse Engine for Semantic Interoperability of Consumer Health Data," IEEE J. Biomed. Heal. Informatics, vol. 20, no. 6, pp. 1632–1639, Nov. 2016.

[31] J. A. Nelder and R. Mead, "A simplex method for function minimization," Comput. J., vol. 7, no. 4, pp. 308–313, 1965.

[32] M. J. D. Powell, "The BOBYQA algorithm for bound constrained optimization without derivatives," Cambridge NA Rep. NA2009/06, Univ. Cambridge, Cambridge, pp. 26–46, 2009.

[33] Clough RW and Penzien J. Dynamics of structures. New York. McGraw-Hill, 1975.

[34] C. Szyperski, J. Bosch, and W. Weck, "Component-Oriented Programming," in Object-Oriented Technology ECOOP'99 Workshop Reader, 1999, pp. 184–192.

[35] de Moraes, J. L. C., de Souza, W. L., Pires, L. F., & do Prado, A. F. "A methodology based on openEHR archetypes and software agents for developing e-health applications reusing legacy systems." Computer methods and programs in biomedicine, 134, pp. 267-287. 2016.

[36] Fasquel, J.B., Brocker, G., Moreau, J., Papier, N., Agnus, V., Koehl, C., Soler, L. and Marescaux, J., "A modular and evolutive component oriented software architecture for patient modeling". Computer Methods and Programs in Biomedicine, 83(3), pp. 222-233. 2006.

[37] Benitez A, Ruiz Zafra Á, Callejas Z, Medina-Medina N, Benghazi K and Noguera M. "A general architecture for the development of virtual coaches for healthy habits monitoring and encouragement" Sensors, 19(1). 2019

[38] Ruiz-Zafra, Á., Noguera, M., Benghazi, K., & Ochoa, S. F. "A model-driven approach for wearable systems developments". International Journal of Distributed Sensor Networks, 11(10), 637130. 2015

[39] Ruiz-Zafra, Á., Benghazi, K., Noguera, M., & Garrido, J. L. "Zappa: An open mobile platform to build cloud-based m-health systems". In Ambient intelligence-software and applications (pp. 87-94). Springer, Heidelberg. 2013