A Gait Analysis Software as a Service

Roberto Aguiar Lima, Vera Regina Da Silva Maraes, Jairo Simao Santana Melo, Vladimir Franca Nogueira, and Lourdes Mattos Brasil, *Member, IEEE.*

***Abstract*—This paper describes the first implementation version of a human gait analysis Software as a Service (SaaS). This approach has as advantage, the software availability at web. After the software is implanted at a web server, users can access him from a recent web browser with support to HTML5. The software objective is to minimize the code development efforts by gait analysis researchers, as well as to be a useful tool for health professionals interested in human gait analysis. The software allows import positionals data from a third party motion captures system, that uses surface markers and video cameras, to plot markers spatial progression, angles, angular velocities and angular accelerations. Furthermore, it is possible to see and to interact with a 3D animation from markers. The software source code is available as free software, often receive new features and a new community is being created to maintain him.**

***Index Terms*—Gait analysis, software as a service, SaaS.**

I. Introduction

WITH the software web advent now is possible to create services, put them at central web servers and use them from any part of world. Furthermore, modern web browsers have become a truly platform, allowing rich interfaces creation, including graphics presentation and 3D animations, without the necessity of plugins installations. These two technologies, web browsers and web servers, can be used to build what is known as Software as a Service (SaaS). The SaaS advantages for customers and software developers are [1]: customers do not need to install the application; data associated with the service is kept centralized, so it is more protected; data can be collectively accessed by a group of users; big datasets and data that is frequently updated, are kept centralized and remote access to them are offered; only a single copy of the server software runs in a controlled hardware and operating system environment, which avoids compatibility problems, in addition, new versions of the software can be tested with a small fraction of the real customers without disturbing most customers.

Although there was gait analysis advancements by the middle of century XX, clinic gait analysis became broadly available only with the modern computer advent [2]. Actually there are a lot of software packages for this finality [3], but until now at century XXI, no software provider committed to deliver a gait analysis SaaS, in other words, health professionals or gait analysis researchers who want use software, have to use software installed at specifics hardware and operating systems, they have to be responsible by data backup, if new features are incorporated to new software versions, the software must be installed again, if they want share data, they must copy and send them to the destiny and others security concerns must be addressed too. All these problems can be minimized or until eliminated with a SaaS.

R. A. Lima, V. R. S. Maraes and L. M. Brasil are with Biomedical Engineering Post Graduate Program, University of Brasilia at Gama, DF.

J. S. S. Melo is with Federal District Court of Law, Brasilia, DF.

V. F. Nogueira is with Universidade of Brasilia at Gama, DF.

Manuscript received October ??, ????; revised December ?, ????.

To build a software it is necessary collect requisites, and a certain domain of the field must be addressed, at this case gait analysis. Thankfully, nowadays, the theme is quite documented [4]–[11]. Moreover, there are health professionals at the development team with much experience in gait analysis. With all this in mind, this paper describes the first implementation version of a gait analysis SaaS [12]. This software version can import data from a third party motion capture system, at this case data collected from video cameras using surface markers, the software also can name markers, define angles, using data from the markers, and plot markers progression at space, angles, angular velocities and angular accelerations. Moreover, the software presents a 3D animation from imported data and allows user interact with him.

II. Materials and Methods

Two researches environments were used to undertake the project. One was the Human Performance Lab (*Laboratrio de Performance Humana* - LPH) in Universty of Brasilia (*Universidade de Brasilia* - UnB) at Gama, and the other was the Health Informatics Lab (*Laboratorio de Informatica em Saude* - LIS) in UnB ant Ceilandia. Data acquisition was made in LPH and software development tasks at LIS.

The next subsections presents the process for data acquisition, the development process and software architecture general view.

A. Data Acquisition

The data are acquired by sixteen Qualisys Oqus MRI cameras, using the Qualisys Track Manager Software (QTM). The data are relative to surface markers along a patient's body.

For this paper a healthy patient, male, with age between 20 and 30 years old was selected. Fig. 1 summarizes the data acquisition process. First markers positions are defined. For this paper only the left trochanter, left knee and left tibia positions were considered. So the surface markers must be fixed at defined positions on patient'body body. The next step is acquire data using the cameras and QTM software. In this step the patient executes a comfortable gait cycle in front of cameras. Five gait samples were acquired. The last step is convert the acquired data to MATLAB format using the QTM software. It is necessary because this is the pattern of choice for the gait analysis software.

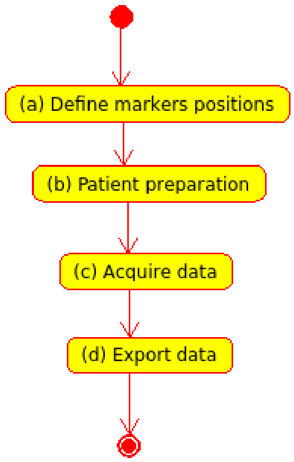


Fig. 1. Data acquisition process. (a) Positions of markers to be fixed in patient body are defined. (b) Markers are fixed at defined positions. (c) Patient executes some comfortable gait cycles in front of cameras. QTM software is used; (d) Data acquired are converted to MATLAB format using QTM software.

The process for data acquisition was approved by *Faculdade de Saude da UnB* ethics committee, process number N11911/12.

B. Development Process

A scrum [13]–[15] inspired process for development was implanted. The scrum is a agile method [16] that has as mainly characteristics to be iterative, incremental and change friendly. These characteristics are essentials to deal with software requirements volatility and hence with so common software changes.

The process flow is presented in Fig. 2. The process consists of iterations, called sprints. The sprints have a 2 weeks average duration. A product backlog is created and managed by the product owner. The product backlog is open to receive additions from all project stakeholders, at any time, but only the product owner can prioritize him, based in expected values for final users. Each item at product backlog is a user story [17]. Before a new iteration begins, there are a two phases meeting between the development team members. In the first phase, the last increment is presented and impediments occurred in last iteration are uncovered. In the second phase the development team selects items from the product backlog. These items will be implemented in the sprint and are called the sprint backlog. Lastly, an increment consisting in a piece of working software is delivered.

C. Software Architecture General View

The Fig. 3 shows the software architecture layered high level view. The layer web applications is responsible by application user interactions (Section II-C1). The layer REST web API is responsible by business logic, (Section II-C2). The document base layer is responsible by data application persistence (Section II-C3).

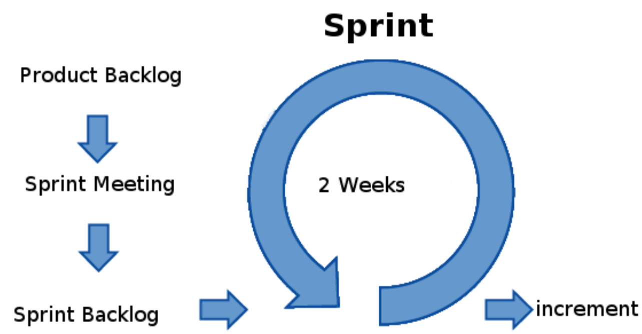


Fig. 2. Development process overview. Every two weeks a new increment, working software, is delivered. Adapted from [14].

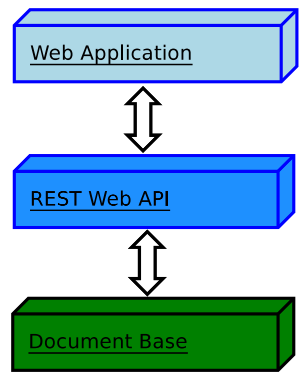


Fig. 3. Software architecture layered high level view.

1) Web Application Layer: This layer was designed to run in web browsers with HTML5 support. It was developed using Javascript, CSS and HTML languages. Moreover, the web development framework AngularJS was adopted [18], [19]. A great AngularJS adoption advantage is the directive creation possibility. Directives are developed components which can be embedded in a HTML template. Also, it was chosen to use the Angular-Material directive library. This library is based in the Material Design specification from Google, which describes about graphical design patterns and user interaction, it is based in the material metaphor [20]. The Angular-Material use, facilitates to build a user experience acceptable by health professionals.

To run graphical 3D animations, the ThreeJS library was chosen [21]. This is a high level library for computer graphics, which takes advantage from WebGL implementation in modern web browsers. The WebGL specification [22] uses the computer Graphical Processor Unit (GPU) natively, allowing to build applications which needs good performance to generate animations and graphics.

Fig. 4 shows the interaction between main components in web application layer. Note that this is only a logic representation. When a user interacts with the application using a web browser, they dispatches events detected by the AngularJS framework. The interaction can be a button click event, a mouse over event, a drag event or other. So the framework selects a controller component to process the event. Controllers

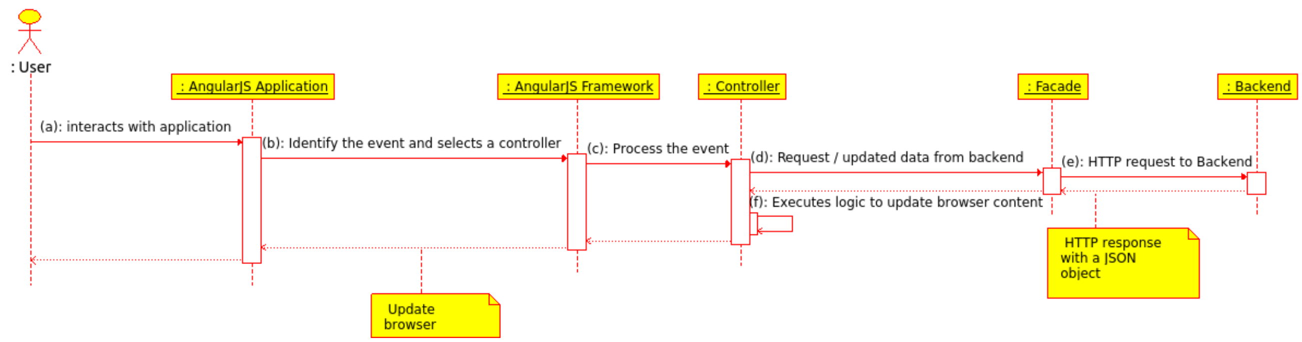


Fig. 4. Main web application layer components interactions. All these components, except the component backend, run inside the web browses. The component backend correspond to the infrastructure and components of REST web API layer. (a) A user interacts with the application, maybe click a button on screen; (b) The AngularJS framework detects the event, selects a controller component and dispatch the event for him; (c) The selected controller process the event; (d) If it is necessary to execute some action, as request more data, the controller requests to a facade component to requests the backend; (e) The selected facade component prepares and send a HTTP request to the backend, which responds with a JSON object; (f) The controller updates the model, variables used by the framework to show data at screen, and signals to the AngularJS framework to update the screen.

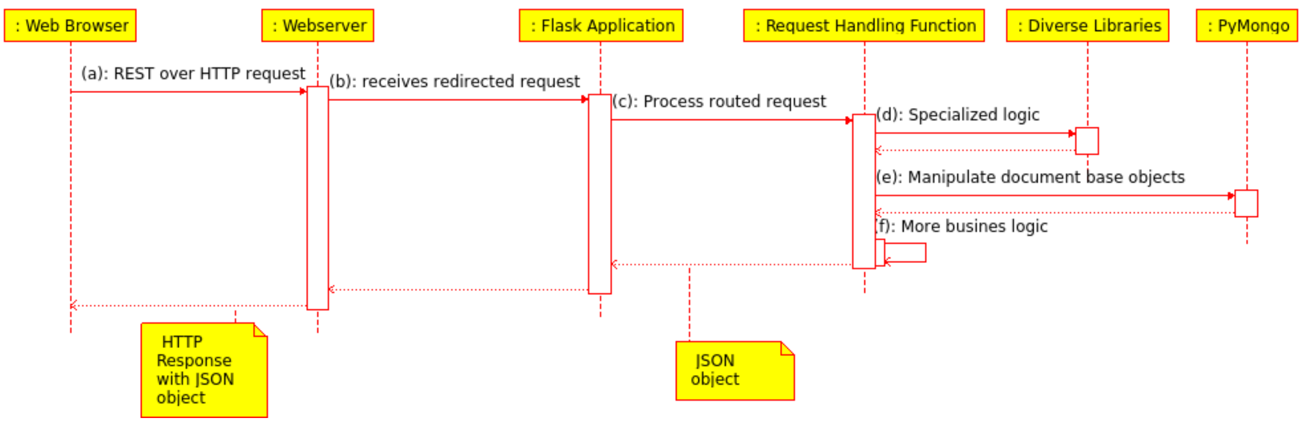


Fig. 5. Main REST web API layer components interactions. (a) From the user web browser a HTTP request is sent to the webserver running the web API; (b) The web server redirects the request to the Flask Application, in this case the REST web API implementation; (c) The flask application routes the request to the appropriate handling function; (d) If necessary the handling function access specialized libraries, like NumPy, Matplotlib or a library for business logic, for example, to calculate kinematics from gait data; (e) If data manipulation from document base layer is necessary, calls to PyMongo, a library for access the document base manager MongoDB, can be made; (f) The function handling can implement business logic too, but it is only advise for simple things. The return data is encapsulated like a JSON object end sent to the webserver to complete the original request from the web browser.

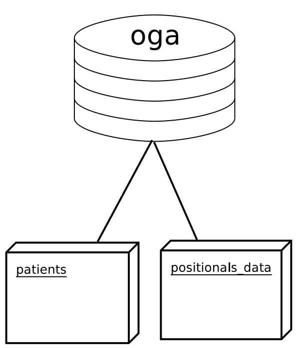


Fig. 6. Application document base. The data base name is ”oga”. This database has two collections: patients and positionals_data.

components are components developed to the application, they are responsible to coordinate data access to the model and to update the view. AngularJS applications uses a famous design pattern called Model-View-Controller [23]. In practice, this means that controllers update the application state, so the AngularJS framework can update the screen to users. If new data or business logic execution are necessary the controller request to a facade component, to communicate with the backend. Facade is a design pattern which hides complex logic necessary to perform some action [23], in this case is the communication over HTTP protocol to the backend. After the backend receives the request, it process him and send a HTTP response with a JSON object inside. JavaScript Object Notation (JSON) are a specification to represent and exchange data.

2) REST Web API Layer: This layer is responsible to execute business logic, for example, calculate angles, extract data from QTM software, among others. The layer was implemented using the architectural style Representational State Transfer (REST) [24]. The main advantage of this technology is the high decoupling with the web application layer. This decoupling promotes concepts separation between these layers, improving development process and application maintainability. To implement the REST style, was chosen the framework Flask [25]. Flask is implemented in Python language, it is known by his minimalist philosophy, allowing to create a simple project, evolving him to more complex models,

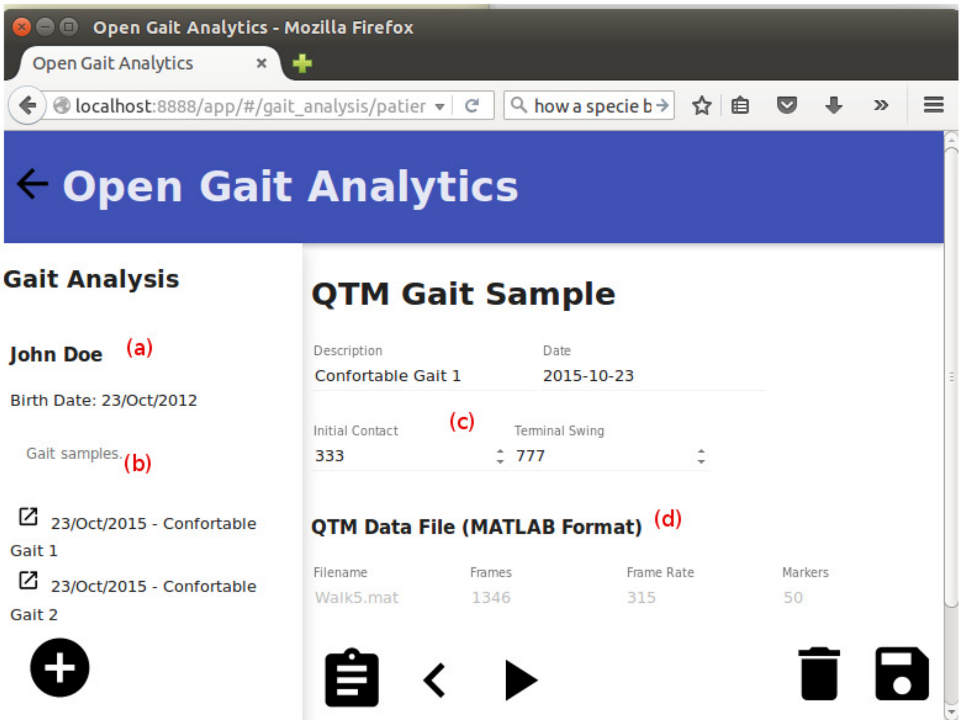


Fig. 7. Gait analysis screen. (a) Patient name; (b) Gait samples acquired from the patient and imported to the software; (c) Initial contact and terminal swing frames. Must be informed by the user, which can use the animation feature to know the correct frames; (d) QTM imported file summary.

according with requisites at moment.

Fig. 5 shows the interaction between main components in REST web API layer. This representation is only a logic view. After a user intact with web application running at web browser and a action from the REST web API is needed, a HTTP request is sent to the webserver. This request is a REST style request. The webserver receives the request, verifies if is a request to the web API, and if yes the request is dispatched to the web API implementation, this is written using the framework Flask. The framework Flask identifies the request and selects the appropriate handling function. This function is written in Python language. The request handling function can them access a lot of libraries, like NumPy for number crunching, Matplotlib for generate scientific graphics that can be sent to the user. Business logic application specific libraries can also be used, like libraries to calculate cinematic information; A very important library that can be used by the handling function too, is the PyMongo library. This component is used to access the document base manager MongoDB, see Section II-C3, so is possible manipulate and persist the application data. The request handling also can implement business logic, but is advise implement a specific library for complex logic. When all needed processing is done a response is generated as a JSON object and sent back to the web application running at the user browser.

3) Document Base Layer: The data persistence and basic manipulation is done by the document base layer. This layer comprises a server running the document base manager MongoDB [26], [26]. This technology was chosen instead of a relational database management system, because the application multidimensional data nature more easily modeled with a document base, and so because the facilities for MongoDB scalability. MongoDB don't use tables like relational databases, it uses the collection concept. A collection is a set of objects. Objects of a collection can have different structures or types, although this is a bad practice. A MongoDB database can have multiple collections. Objects are expressed using de BSON notation, that is very similar to JSON.

The Fig. 6 shows the application data base. It have two data collections: patients and positionals_data. The patients collections has basic data from the patient. The positionals_data collection has spatial and markers data imported from a QTM/MATLAB file.

III. RESULTS

The developed software permits to register patients. After the patient is registered, the user can include gait samples to these patient This is done importing a MATLAB formatted file exported from QTM. After the file is imported the screen described in Fig. 7 is shown. From this screen is possible to run a 3D animation including the markers acquired (Fig. 8). Here is possible accelerate or slowdown the animation, to apply zoom in or Zion out, to controller pan and to see a specific frame in the animation. To see a frame is am important feature, because it can be used to visualize the initial contact and terminal swing frames, so is possible to update these fields in the screen shown at Fig. 7(c). These data fields must be correctly filled or the graphics will be incorrect.

The software has an option to name markers. After a marker is named, a graphic showing the marker's spatial progression can be plotted (Fig. 9).

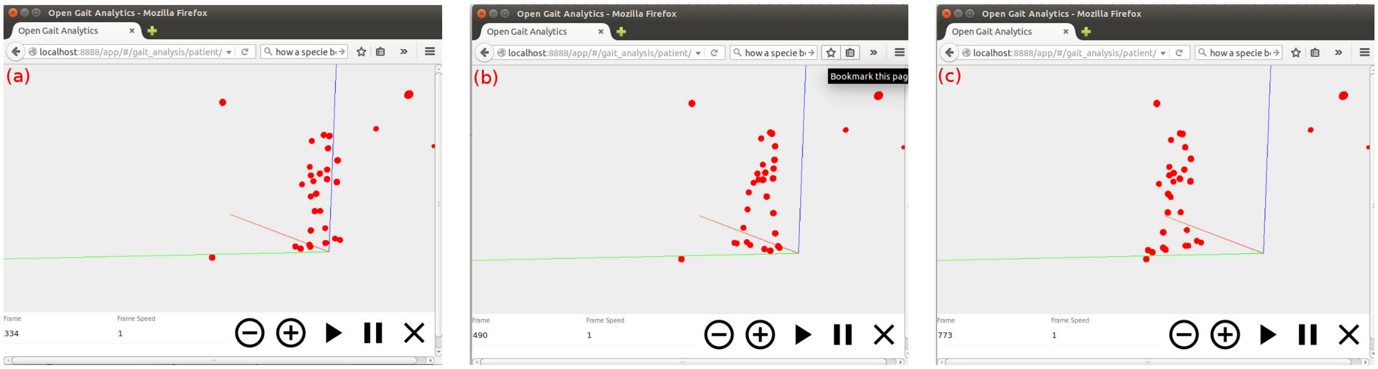


Fig. 8. Markers 3D animation. (a) Left inferior limb initial contact; (b) Left inferior limb terminal instance; (c) Left inferior limb terminal swing.

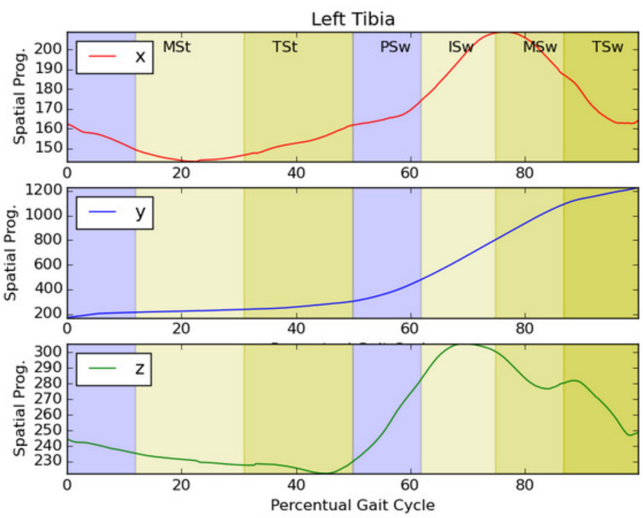


Fig. 9. Left tibial spatial progression.

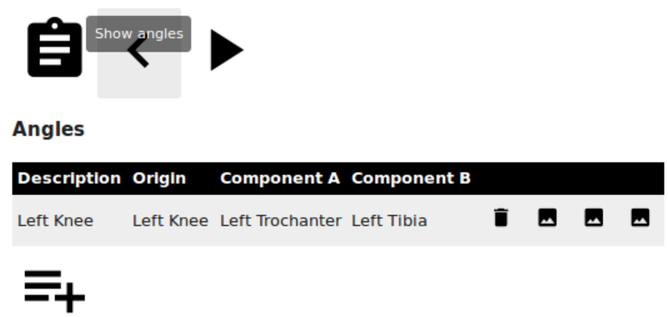


Fig. 10. Angle registering tool.

Another important feature is the angle registering tool (Fig. 10). After a angle is registered, angles (Fig. 11), angular velocities (Fig. 12) and angular accelerations (Fig. 13) can be plotted.

IV. Discussion

A software for gait analysis fully available at web is a innovation in the field. The bigger advantage which this approach does is the easy access to the software, because after the application is implanted in an internet webserver, any user with a recent HTML5 browser can use the application, independent from his location.

Another theme that raises from this project is the creation of a centralized base, with data collected around world. Such endeavor would be priceless to gait analysis researchers. This would require a more profound care with information security.

The software is available as free software. The intention is to attract developers and health professionals to contribute and to use the software. These profiles already exists in the project, but to fully reach the project potential, some more people will be necessary.

As future works, new data acquisition methods cold be implemented, as plate force, electromyography (EMG) and inertial measurement unit (IMU). The inclusion of automatic detection events at gait cycle could be a plus, for example, detect the initial contact and terminal swing. Furthermore, a simulation module is being developed. It will support machine learning methods to regression and classification.

V. Conclusion

This paper presented a gait analysis SaaS. His source code is available as free software under the MIT license [27] ate web address http://github.com/rob-nn/open_gait_analytics.

The software offers data importation from the third party software QTM from Qualisys. These data are about surface markers fixed in a patient body. The software permits name markers, plot a spatial progression from them. Also is possible to register angles and show graphics with angular velocities and accelerations. A 3D animation showing the markers is available. All graphics are visualized according the gait cycle phase.

The software will continue receive more features, and gradually will be available for more users. Anyway is possible to other people implant him by themselves.

Acknowledgment

The first author would like to thank the financial support granted by CAPES. The authors wold like to thank LIS UnB at Gama and LPH UnB at Ceilandia, by use of its resources.

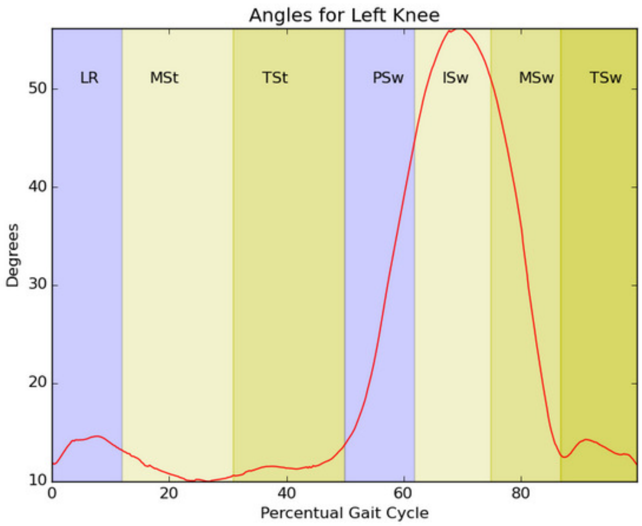


Fig. 11. Angles.

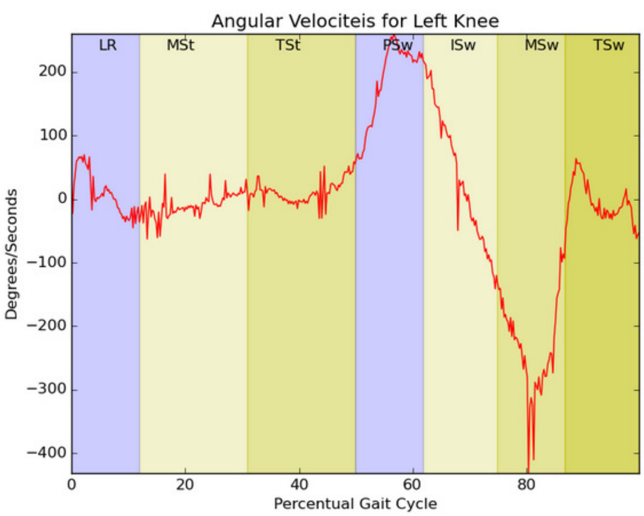


Fig. 12. Angular velocities.

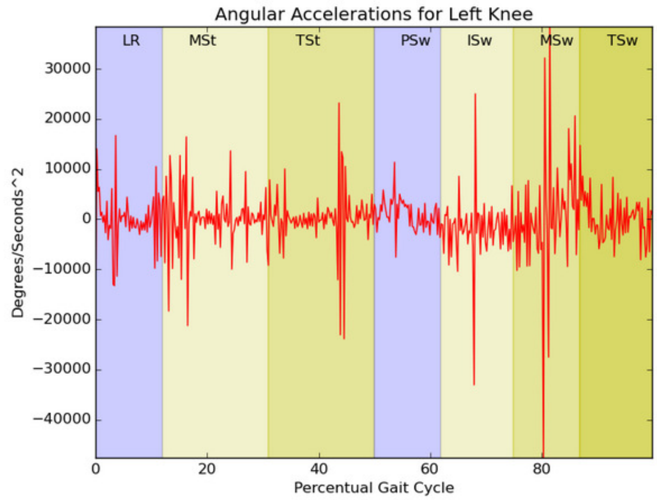


Fig. 13. Angular accelerations.

REFERENCES

[1] A. Fox and D. Patterson, *Engineering Long-Lasting Software: An Agile Approach Using SaaS and Cloud Computing*. Strawberry Canyon LLC, 2012.

[2] R. Baker, "The history of gait analysis before the advent of modern computers," *Gait and Posture*, vol. 26, no. 3, pp. 331–342, 2007.

[3] J. Moraes, S. Silva, and L. Battistela, "Comparison of two software packages for data analysis at gait laboratories," *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No.03CH37439)*, vol. 2, pp. 1780–1783, 2003.

[4] J. Perry and J. M. BurnField, *Gait Analysis Normal and Pathological Function*, 2nd ed. SLACK Inc., 2010.

[5] M. W. Whittle and D. Levine, *Whittle's Gait Analysis*, 5th ed. Churchill Livingstone, 2012.

[6] J. P. Ferreira, M. M. Crisostomo, and a. P. Coimbra, "Human gait acquisition and characterization," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 9, pp. 2979–2988, 2009.

[7] A. Vieira, H. Sobral, J. P. Ferreira, P. Ferreira, S. Cruz, M. Crisóstomo, and A. P. Coimbra, "Software for human gait analysis and classification," in *4th Portuguese BioEngineering Meeting*, no. February, Porto, 2015.

[8] a. Duhamel, J. L. Bourriez, P. Devos, P. Krystkowiak, a. Destée, P. Derambure, and L. Defebvre, "Statistical tools for clinical gait analysis," *Gait and Posture*, vol. 20, no. 2, pp. 204–212, 2004.

[9] S. Ghoussayni, C. Stevens, S. Durham, and D. Ewins, "Assessment and validation of a simple automated method for the detection of gait events and intervals," *Gait and Posture*, vol. 20, no. 3, pp. 266–272, 2004.

[10] A. Moreno, I. Quiñones, G. Rodríguez, L. Núñez, and a. I. Pérez, "Development of the spatio-temporal gait parameters of Mexican children between 6 and 13 years old data base to be included in motion analysis softwares," *2009 Pan American Health Care Exchanges - PAHCE 2009*, no. 2, pp. 90–93, 2009.

[11] S. Beynon, J. L. McGinley, F. Dobson, and R. Baker, "Correlations of the Gait Profile Score and the Movement Analysis Profile relative to clinical judgments," *Gait and Posture*, vol. 32, no. 1, pp. 129–132, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.gaitpost.2010.01.010

[12] R. A. Lima, "Implementando um Software como Serviço para Análise e Simulação de Marcha Humana," 2015.

[13] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*, 1, Ed. Pearson, 2001.

[14] K. Schwaber, *Agile Project Management with Scrum*. Microsoft Press, 2004.

[15] K. S. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley Professional, 2012.

[16] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, "Manifesto para Desenvolvimento Ágil de Software," 2001. [Online]. Available: http://www.agilemanifesto.org/

[17] M. Cohn, *Users Stories Applied*. Crawfordsville: Addison Wesley, 2004.

[18] R. Branas, *AngularJS Essentials*. Birmingham: Packt Publishing Ltd., 2014.

[19] A. Freeman, *Pro AngularJS*. Apress, 2014.

[20] Google, "Material Design," 2015. [Online]. Available: www.google.com.br/design/spec/material-design/introduction.html

[21] J. Dirksen, *Learning Three.js - the JavaScript 3D Library for WebGL*, 2nd ed. Packt Publishing, 2015.

[22] K. Matsuda and R. Lea, *WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL (OpenGL)*. Addison-Wesley Professional, 2013.

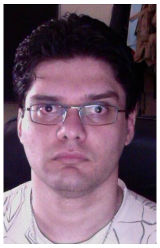
[23] M. Fowler, *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional, 2002.

[24] M. Grinberg, *Flask Web Development*, 1st ed. O'Reilly Media, Inc., 2014.

[25] I. Maia, *Building Web Applications with Flask*, 1st ed. Packt Publishing, 2015.

[26] E. Plugge, P. Membrey, and D. Hows, *MongoDB Basics MongoDB Basics*, 1st ed. Apress, 2014.

[27] MIT, "The MIT License (MIT)," 2015. [Online]. Available: https://opensource.org/licenses/MIT



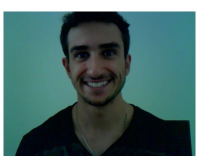
Roberto Aguiar Lima is graduated in computer science from Brasilia Catholic University (2000). He attends the University of Brasilia Biomedical Engineering M.Sc. Program. He has more than twenty years as software developer.

He is envolved in gait analisys research and has interest in machine learning and software engineering.

Vera Regina Da Silva Mares is graduated in physioterapy by Sao Carlos Federal University (1995). She has a M.Sc. degree in celular and structural biology by Campinas State University, and a Ph.D. degree in physioterapy from Sao Carlos Federal University. Actually she is adjunct professor III at: University of Brasilia at Ceilandia; University of Brasilia Biomedical Engineering M.Sc. Program; and University of Brasilia Health Pro-Education.

She has experience in physioterapy, acting mainly in: health teatching, amputee prosthesis adaptation, human functionality, exercice phisiology, cardiac rehabilitation and cardiac frequency variability.

Jairo Simao Santana Melo ?????



Vladimir Franca Nogueira ?????

Lourdes Mattos Brasil .Electrical Engineer, Federal University of Santa Catarina (1984), Master of Science in Electrical Engineering/Biomedical Engineering, Federal University of Santa Catarina (1994), Sandwich Doctorate in Mathematical Applied from Facultes Universitaires Notre Dame de La Paix (FUNDP), Belgium (19971998) and D.Sc. in Electrical Engineering/Biomedical Engineering (1999). She is currently professor/researcher/head of the Lato Sensu in Clinical Engineering Course, besides being professor of the Course of Electronic Engineering, at the University of Brasilia at Gama, Brazil. Her scientific works concern: artificial intelligence, data mining, machine learning, knowledge acquisition, knowledge based systems, hybrid expert systems, virtual reality, intelligent tutoring systems, informatics in health, elearning and nanotechnology.