

Physically-Based Simulation and Web Visualization of *C. elegans* Behavioural Experiments

Andoni Mujika ^{1,*}

Email amujika@vicomtech.org

Gorka Epelde ¹

Email gepelde@vicomtech.org

Peter Leškovský ¹

Email pleskovsky@vicomtech.org

David Oyarzun ¹

Email doyarzun@vicomtech.org

¹ Vicomtech-IK4, Mikeletegi Pasealekua, 57, 20009 Donostia-San Sebastian, Spain

Abstract

This paper presents the work done in the framework of the *Si elegans* project to develop the physics engine for the simulation of the roundworm *Caenorhabditis elegans* and the interfaces to define and visualize behavioural experiments of the worm. The physically-based simulation of the locomotion of the worm is guided by a biomechanical model, based on anatomically matched biphasic springs. The simulation is presented via an experiment visualization web, using a 3D motion reproduction obtained through animation bones. This web also displays information about the activation in muscles and neurons, on additional information panels. Finally, an experiment definition portal has been developed where, by means of a timeline, the user can easily design complex experimental assays.

Keywords

WebGl visualization

Physically-based simulation

Caenorhabditis elegans

1. Introduction

Caenorhabditis elegans (*C. elegans*) is one of the best known organisms in the world [1], widely used for different biological or medical assays. This is because, on the one hand, *C. elegans* is very useful for genetic studies, considering that its genome is completely known and is up to 35 % similar to the human one. The functions of most of its genes are known and its manipulation for chemical and genetic tests is relatively easy. On the other hand, the relative simplicity of its neuronal system (the hermaphrodite *C. elegans* has 302 neurons) permits to study how its nervous system works. Moreover, it shows a rich behavioural repertoire (e.g. locomotion, feeding and even certain social behaviours).

All these aspects make *C. elegans* a perfect organism for study in the way of understanding how the neural processing in more complex organisms, such as our organism, work. Nevertheless, even if the connectivity of the 302 neurons is known, the knowledge of the scientific community does not explain all the behaviours generated by its nervous system.

The *Si elegans* project aims to emulate the neuronal system of *C. elegans* and show the behaviour that emerges from this emulation in a virtual environment in three dimensions. This way, it will provide the scientific community with **an alternative and complementary tool** ~~tool-alternative and complementary~~ to the laboratory experiments. The specifics of the *Si elegans* project are the hardware-based emulation of the neural network and the parallelism of the neuron to neuron communication.

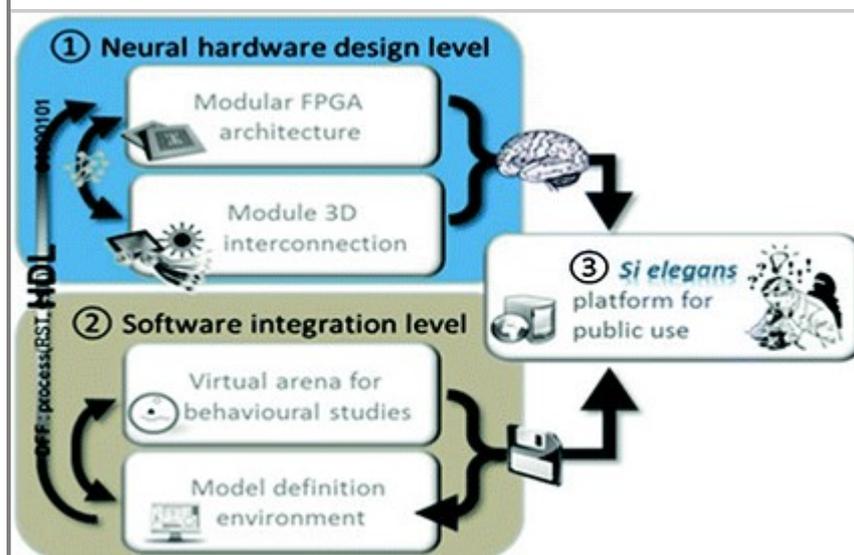
In the following section, we will introduce the *Si elegans* platform. In Sect. 2 we review the state of the art on the *C. elegans* locomotion simulation. Sections 3, 4 and 5 describe all the aspects regarding physical simulation of the worm and its web visualization (and interaction). Finally, Sect. 6 draws conclusions and discusses the future work.

1.1. The *Si elegans* Platform

The *Si elegans* platform is conceptually composed of two main blocks responsible for the emulation of *C. elegans* and a third block responsible for making the platform available for public use (see Fig. 1).

Fig. 1.

The *Si elegans* conceptual platform



The first block (identified as Neural hardware design level in Fig. 1) is a hardware-based computation framework that emulates the neural system of the nematode in real time. The nervous system of *C. elegans* is being replicated on a field-programmable gate array (FPGA) hardware architecture. Each of the neurons is represented in a unique FPGA and will be configured and parameterised by the user. This configuration will be carried out via the third block (identified as the platform for public use in Fig. 1). Regarding the inter-neuron connectivity, we focus on true reproduction of the parallel information flow between neurons; implementing the inter-neuron connectome using optical off-chip interconnects. Intensity-tuneable laser diode emitters will be triggered by the FPGA neuron modules and the light will be distributed through micro optical light elements to specific pixels of light-receptive arrays.

The second block (identified as Software integration level in Fig. 1) provides the tools to convert the high level neuronal system's configuration (provided by the end-users through the third block) to the configuration required by the neural hardware implementation. It also has a virtual arena that simulates the physics of the worm and its environment.

The third block (identified as *Si elegans* platform for public use in Fig. 1) is responsible for providing the end-users with a web interface to define the neuronal system's configuration and the behavioural experiment, as well as for visualizing the results of the experiment. The user will (i) set all the necessary parameters for defining an experiment (e.g. position of the worm or temperature at different positions of the environment), (ii) watch the resulting motion of the virtual nematode, (iii) monitor the activity of the neuron network, (iv) define new neuronal models, and neuronal network configurations, and (v) share neuron models, experiment configurations and results with the scientific community.

This paper details the solution adopted for the physics-simulation of the worm (identified as the virtual arena for behavioural studies) and the approach of making such simulation available to the end-user (identified as *Si elegans* platform for public use in Fig. 1). Additionally, we will describe the web-based public access part used to define and monitor all the aspects related to the behavioural experiment.

2. Related Work

The pioneering work on the physics simulation of the *C. elegans* was done by Niebur and Erdős [2], in the early nineties. They designed a two-dimensional worm with muscles represented by straight segments. Different forces were applied to the muscles in order to make the body move in a way similar to the real worm.

Thereafter, most models that have been presented [3–5] adopted similar musculo-skeletal anatomy of the worm and force systems that actuate on the worm and contribute to the generation of the worm's typical motion: internal pressure of the worm, elasticity of the cuticle ("skin" of the worm), muscle forces and environmental forces.

Considering a 2D simulation of locomotion, Boyle et al. [6] developed one of the most realistic models that is suitable for different environments such as water, gelatine or agar. They suggest that the environment does not affect the muscle function because all the different behaviours of the worm, observable in different environments, emerge from the same model. The authors presented a neuronal model that makes the worm adapt to the environment with minimal sensory input.

In recent years, 3D simulations of the physics of the worm have appeared. They aim to obtain a more realistic representation of the worm, allowing for moves in the vertical axis. The model presented by Mailler et al. [7], uses a limited neural model and a worm made of 25 rigid cylindrical sections, which is very similar to those described above, but implemented for physics in three dimensions.

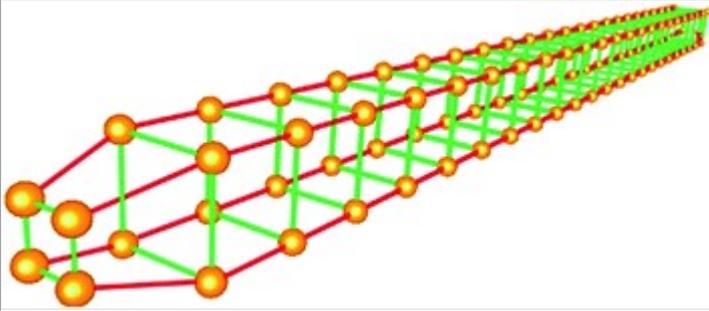
The main active project in the field of simulation of *C. elegans*, the OpenWorm Project [8], is also working with 3D simulation of the physics of the worm. It uses Predictive-Corrective Incompressible Smoothed Particle Hydrodynamics (PCI-SPH) [9] to model the behaviour of all the particles that take part in the simulation (including fluids). The realism they want to achieve is very high, although at the moment the performance is too slow for running in a closed-loop with a hardware-based emulation of the neural system and achieve a close to real-time performance (*Si elegans* project's objective).

3. Physical Model

The hermaphrodite *C. elegans* has 95 muscles [10], divided in 4 quadrants of 24 muscles, except for the ventral left quadrant which has 23 muscles. Although a more realistic structure is being developed, our first prototype has 96 symmetrically distributed muscles, as most of the state of art works do. Figure 2 shows the structure constructed to simulate the worm. In each row (quadrant), there are 24 red segments that represent the muscles. These are connected between the quadrants by rings compound of 4 green segments. The first and the last ring are smaller in order to obtain a more similar shape to the shape of the worm.

Fig. 2.

Structure of our virtual *C. elegans*



The basis of our model is a classical mass-spring model, but in this case biphasic springs have been used. A biphasic spring is a linear spring that changes its spring constant at a certain length. This way, we stop the spring from elongating more than desired. The formulas describing the kinematics of the biphasic spring are as follows:

$$f_s = -k_s(L_c - L_r) \left(\frac{\vec{p}_2 - \vec{p}_1}{\|\vec{p}_2 - \vec{p}_1\|} \right)$$

$$f_d = -k_d \left(\vec{v}_2 - \vec{v}_1 \right) \cdot \left(\frac{\vec{p}_2 - \vec{p}_1}{\|\vec{p}_2 - \vec{p}_1\|} \right) \left(\frac{\vec{p}_2 - \vec{p}_1}{\|\vec{p}_2 - \vec{p}_1\|} \right)$$

where f_s and f_d are the spring and damping forces, respectively, that will be applied to the end-points of the springs (in opposite direction), k_s is the variable spring stiffness dependent on the elongation of the current spring, k_d is the damping constant (not depending on the length of the spring), L_c is the current and L_r is the rest length the spring, p_1 and p_2 are the end-points of the spring and v_1 and v_2 are the velocities of the points p_1 and p_2 , respectively.

In order to generate the locomotion of the worm, these springs need to be activated, i.e. they must contract and relax without any external force. For that, a **modulating** parameter is inserted in the first equation which is modified directly by the motor neurons associated to the corresponding muscle spring. In the case of the rings that maintain the shape of the body, they are springs with higher stiffness and do not activate themselves.

The physical model must define the forces that will be applied to the virtual worm within the simulated environment in order to obtain a coherent locomotion of the whole worm. The forces implemented in our physical model are:

- Muscle contraction: implemented by active springs.
- Cuticle elasticity: implemented by passive springs.
- Gravitation force.
- Internal pressure: as gravity would collapse the spring structure, internal forces are simulated. We use a force field that models behaviour similar to a cube made out of silicon, similar to the one described in [11]. The body of the worm is divided into a grid of hexahedra, supposed to behave isotropically, and then a linear Finite Element Method (FEM) is used to simulate the internal forces, as opposition to the collapsing tendency of the spring structure.
- Friction: is the crucial force which ensures that the worm will move forwards. In our model it is based on Signorini's law and Coulomb's friction law. In the simulation loop, the positions of the mass-points defining the worm are computed taking into account collisions with the floor. Then, based on the considered physical laws, a Non-linear Complementary Problem is obtained and solved for with numerical strategies. Finally, the previous position is corrected in order to encounter for the computed contact forces.
- Other collisions: (e.g. with obstacles) are detected and the resultant forces are applied to the nematode.
- User-defined mechanosensory interaction: for example, the user will be able to touch the worm with a laboratory instrument at a certain time of the simulation. Furthermore, experiments with changes in temperature or chemical substances will be implemented.

All the physics simulation is run using the Simulation Open Framework Architecture (SOFA) [12]. This tool facilitates the creation of complex environments using a wide range of components (solvers, objects, collision detection algorithms, etc.). In our case, these components have been used:

- A Constraint Solver implementing the Non-linear Complementarity Problem formulation described above.
- An Euler Implicit Solver for Ordinary Differential Equations.
- A Conjugate Gradient Linear Solver for systems of linear equations.
- Bounding box hierarchies and a simple triangle/point intersection algorithm for collision detection.

4. Visualization

This section presents how the simulation of the locomotion of *C. elegans* is converted from the simulation calculated in the physics engine to the visualization displayed to the user in a web browser. Since most users of the *Si elegans* framework will be scientists, they might be interested in observing either only the general behaviour of the worm or in analysing the behaviour of a specific neuron or a muscle, and more specifically they would prefer to have access to the values of its inner parameters at each time and not only its position in the simulated environment. Thus, the positions of the whole spring structure are not transferred to the web client.

Only the minimum required information is transferred to the web and there, a realistic 3D reproduction of

the nematode (see Fig. 3) is generated according to the simulation computed in the physics engine. To render the 3D environment without any plug-in installation, the Three.js library [13], that makes use of WebGL [14], has been used.

Fig. 3.

3D reproduction of *C. elegans*, showing its 302 neurons



As shown in Fig. 3, the degree of transparency of the skin of the animal can be changed in order to make the neurons visible. In the next stages of the project, these reproductions of neurons and muscles will be used to show additional information about them, e.g. the use of different colours to represent different degrees of excitation of a neuron.

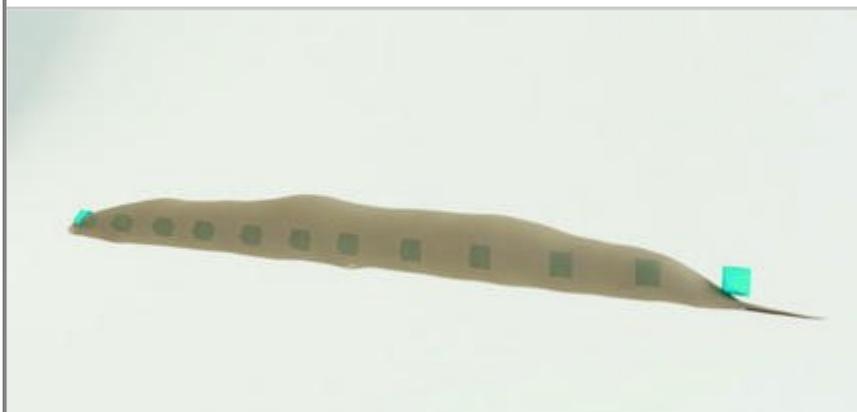
To synchronize the motion of the 3D reproduction in the web with the worm in the physics engine, animation bones have been set along the body of the virtual worm (see Fig. 4). The triangle mesh that represents the body of the worm is attached to these bones, and thus the mesh moves accordingly to the transformations of the bones. The bones are transformed according to the following formula:

$$q_i = \frac{p_{vl,2i} + p_{cr,2i+1}}{2} \quad i = 0, \dots, 11$$

where q_i is the position of the animation bone in the web site; $p_{vl,j}$ is the j th mass point in the ventral, left row and $p_{cr,j}$ is the j th mass point in the dorsal, right row of the server side worm.

Fig. 4

Animation bones of the worm



5. Web Interaction

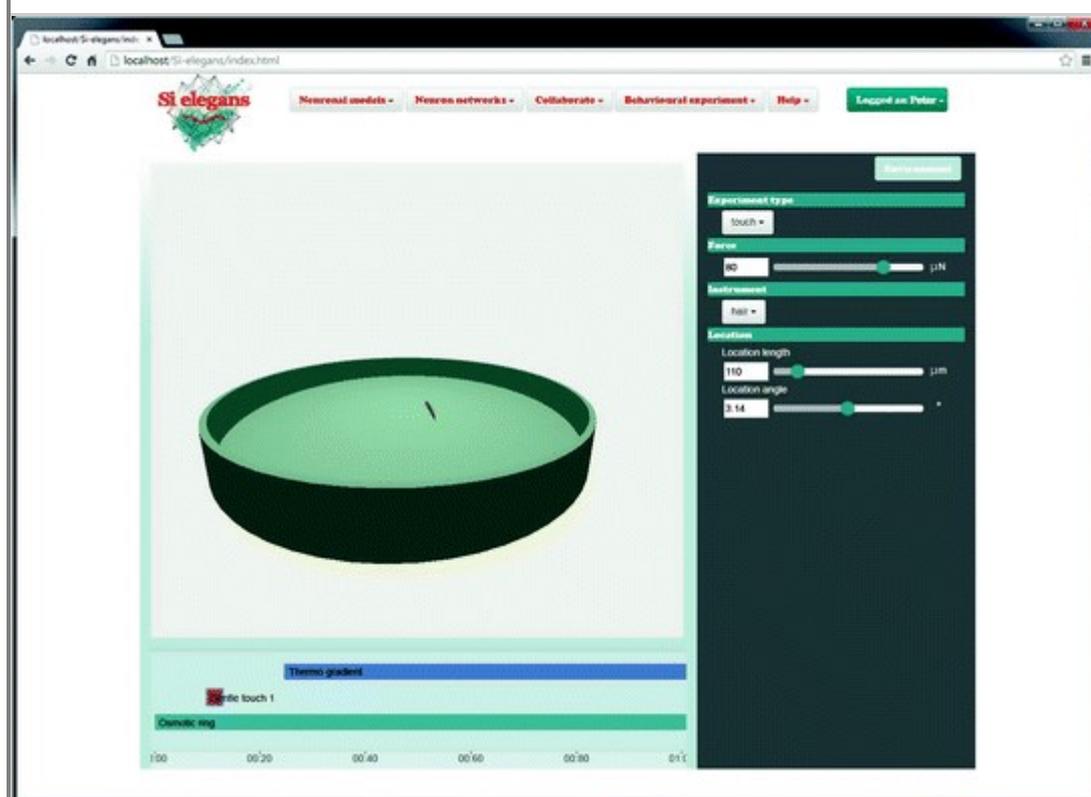
The *Si elegans* web portal will be used to define all the aspects of the *Si elegans* emulation system (neuron models, connectivity, behavioural experiments...) and to visualize the results obtained from the system. In this section, we describe the interfaces developed for the definition of the behavioural experiment and the visualization of its simulation results.

5.1. Defining the Experiment

The experiment definition web portal of *Si elegans* will have three main windows: the experiment definition window, the properties window and the 3D visualization window. Figure 5 shows these three parts of the web-page.

Fig. 5.

Experiment definition web-page



A timeline editor (at the bottom of the web-page) is the interface that controls the experiment creation. All the events that will happen during the assay are placed in the timeline, marked with a colour specific to the given type of event. This way, the configuration of the experiment is clearly recognized, at a glance.

In the properties windows, all the properties of the selected event are shown (on the right side of the website). For instance, in a temperature gradient experiment, the user will set where the heat source will be placed, which temperature it will have and at which moment of the experiment it will be activated.

Besides, in the properties window, the parameters of the environment can be defined: shape of the plate, environmental substance (water, air, silicon, etc.), the initial position and the orientation of the animal, obstacles, etc.

Finally, the 3D visualization window (in the middle) shows the initial configuration of the experiment

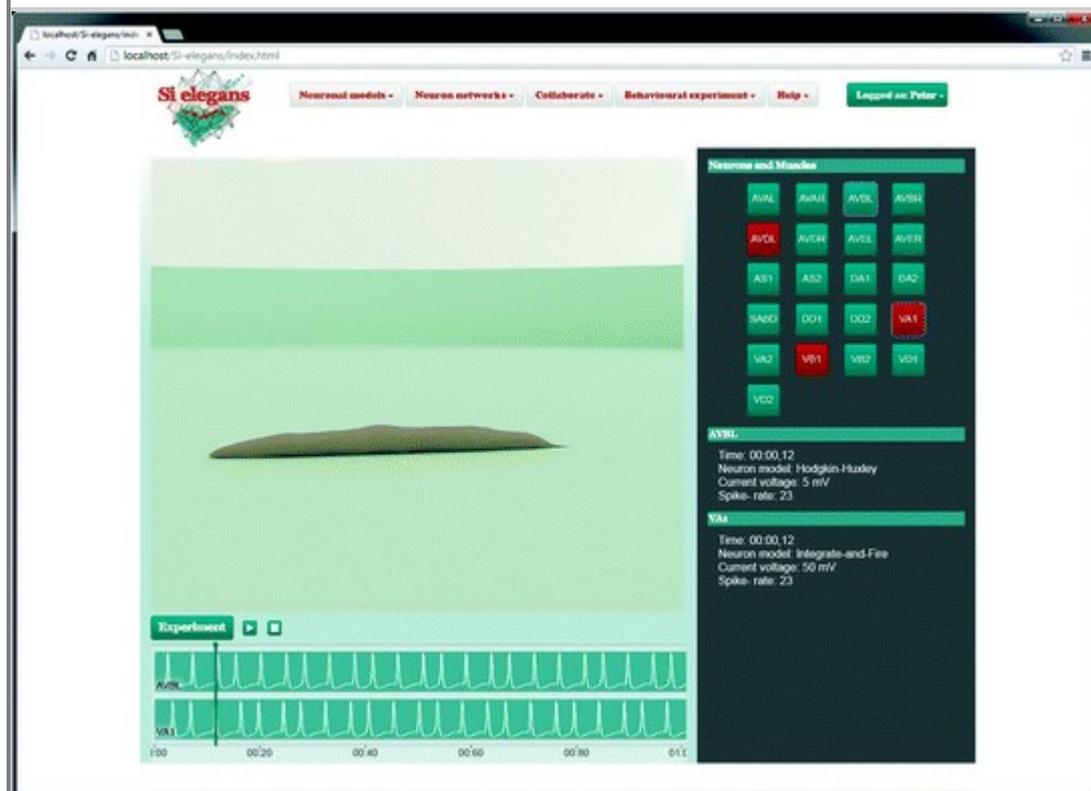
(without animating anything). This window can be used to insert obstacles in the plate, to choose the place where the toxin drop will be released during the experiment, or to choose at which part of the body the worm will be touched, either by clicking on the 3D plate or on the 3D reproduction of the worm, to mark the desired location.

5.2. Observing and Analysing the Results

The web-page for the visualization of the results of the defined experiment consists of three windows: the selection window, the timeline window and the 3D graphics visualization window. The appearance of this web-page is shown in Fig. 6.

Fig. 6.

Visualization web-page



The selection window (at the right side of the web-page) is a grid of squares representing all the neurons and muscles that have been selected to retrieve data. The colour of the squares represents the voltage of each neuron; a spike of a neuron is represented with red colour. This way, the user can know what is happening in all the neurons and muscles he is interested, at a glance.

Furthermore, the user can explore the behaviour of a specific neuron in more detail by clicking one cell of the grid. In the timeline window (at the bottom of the page), a graph representing the voltage of those selected neurons will appear aligned with the experiment timeline and in the selection window, additional information, such as the neuron model that has been chosen for that neuron, is shown.

The timeline window is also used to control the timing of the simulation or to show the previously defined experiment and all the parameters (temperature, chemical concentration...) that take part in the assay. Thus, the user can check how each different stimulus influences the function of the neurons.

The 3D graphics window (in the middle) is where the behaviour of the worm will be displayed, as

calculated by the physics engine and transformed for web representation using the animation bones technique, as described in previous sections. Similar to the experiment definition tool, the user will be able to explore the area with the mouse. At this moment, internal parts of the worm are shown, but they are not interactive. In further stages of the project, the 3D window will be used to select cells of the worm or to show advanced/additional information.

6. Conclusions and Future Work

We have described the tools that allow users to define, simulate and follow experiments with the *C. elegans* nematode in the frame of the *Si elegans* project.

Firstly, we have presented the physically-based simulation of the locomotion of the worm. For that, we have created a structure made of biphasic springs, modelling the basic anatomy and the muscular system of the worm. The muscles contract themselves following the signals that come from the neurons. Currently, a simplification that models four equal muscular quadrants has been used. Nevertheless, in further stages of the project, the muscular structure will be changed to make it more similar to the one of the real *C. elegans*.

After the simulation, the movements of the muscles are transformed to a realistic 3D visualisation of the animal, applying the animation bones technique. The visualization web will help the user to understand what is going on in the physiology of the nematode. At the moment, the information about the muscles and the neurons is displayed on information panels, but in the future, the 3D reproduction will be used to show this information.

Finally, we developed a web portal that uses a timeline to design the experiment that will be simulated by the *Si elegans* platform. The user will be able to easily design assays as complex as desired. The challenge of this part of the project will be to implement more types of possible biological assays and allow the user to specify the corresponding experiment details in a user-friendly way.

Acknowledgments

The *Si elegans* project is funded by the 7th Framework Programme (FP7) of the European Union under FET Proactive, call ICT-2011.9.11: Neuro-BioInspired Systems (NBIS).

References

1. Altun, Z., Hall, D.: Introduction. In: [WormAtlaswormatlas](#) (2009)
2. Niebur, E., Erdos, P.: Theory of the locomotion of nematodes. *Biophys. J.* **60**, 1132–1146 (1991)
3. Suzuki, M., Goto, T., Tsuji, T., Ohtake, H.: A dynamic body model of the nematode *C. elegans* with neural oscillators. *J. Robot. Mechatron.* **17**(3), 318–326 (2005)
4. Bryden, A., Cohen, N.: Neural control of caenorhabditis *elegans* forward locomotion: the role of sensory feedback. *Biol. Cybern.* **98**(4), 339–351 (2008)
5. Voegtlin, T.: Clones: a closed-loop simulation framework for body, muscles and neurons. *BMC Neurosci.* **12**(1) (2011)
6. Boyle, J.H., Berri, S., Cohen, N.: Gait modulation in *C. elegans*: an integrated neuromechanical model.

- Front. Comput. Neurosci. **6**(10) (2012)
7. Mailler, R., Avery, J., Graves, J., Willy, N.: A biologically accurate 3D model of the locomotion of *caenorhabditis elegans*. In: Proceedings of the 2010 International Conference on Biosciences (BIOSCIENCESWORLD'10), pp. 84–90 (2010)
 8. Vella, M., Palyanov, A. Khayrulin, S.: Integration of predictive-corrective incompressible SPH and Hodgkin-Huxley based models in the OpenWorm in silico model of *C. elegans*. BMC Neurosci. 8 July 2013, 14 (Suppl 1):P209 (2013). doi: 10.1186/1471-2202-14-S1-P209
 9. Solenthaler, B., Pajarola, R.: Predictive Corrective Incompressible SPH. ACM Trans. Graph. **28**, 3, pp. 1-6 (2009)
 10. Altun, Z.F., Hall, D.H.: Muscle system, somatic muscle. In: WormAtlas (2009)
 11. Nesme, M., Marchal, M., Promayon, E., Chabanas, M., Payan, Y., Faure, F.: Physically realistic interactive simulation for biological soft tissues. Recent Res. Dev. Biomech. **2**, pp.1-22 (2005)
 12. Allard, J., Cotin, S., Faure, F., Bensoussan, P.-J., Poyer, F., Duriez, C. Delingette, H., Grisoni L.: SOFA—an open source framework for medical simulation. In: Medicine Meets Virtual Reality (MMVR'15), pp. 13–18 (2007)
 13. Three.js (2014). <http://threejs.org/> [Last access: October 2015]
 14. WebGL (2014). <http://www.khronos.org/webgl/> [Last access: October 2015]