Information and Computing Sciences
Faculty of Science, Utrecht University

Experimentation Project

Simple Pose Controller for Muscle Actuated Characters

Author:

Stavros Tsikinas (3938131)

Supervisor:

Dr. Nicolas Pronost

July 2014

Abstract

This report presents the experimentation project that was developed, in order to create a framework for muscle based animation for the lower body of a musculoskeletal human character. The character consists of muscles based upon the *Hill's Muscle Model*. This report presents the procedure of developing this framework and the GUI of the framework. It is demonstrated in a scenario to assist the character reach a certain pose. The full control of the character is outside the scope of this project.

Table of Contents

1. Introduction

This experimentation project deals with a category of physics-based animation, musculoskeletal animation. In this project, skeletal muscles are producing force, depending on the contraction of those muscles, determined by the user. These forces are used, in order to change the pose of a human character. It is a challenging field, because it introduces the ability of the user to manipulate the character, by activating individual body muscles to reach a certain pose. The purpose of this project is not to create a controller in which the character is moving, but to establish the fundamentals of building a framework, where the user is able to change the actuation value of each muscle individually and visualize the change in pose for a character. The project is focusing on the 48 muscles of the lower body of the character, due to complexity of scaling the muscles of the whole body of the human character.

The past few years there have been studies dealing with musculoskeletal animation. There has been research dealing with specific parts of the body, such as hand animation [1] and research dealing with human locomotion, achieved by muscle reflexes [2]. Also, there is research conducted for full 3D bipedal characters control, which controls the characters using muscles and doing specific tasks [3], with the use of the *Hill-Type muscle model* [ref]. However, this research focuses on the muscle model, rather than interactively handle the muscle actuation values. Furthermore, there has been research made concerning to what extend does a muscle injury effects the movement of a virtual character [4]. The last project is taken into great consideration for the fulfillment of this experimentation project, because the work of scaling the *OpenSimSet* muscle data to the character used for this project, is determined. Also, in [9] the muscle model is being adopted, in order to determine the forces applied to the character.

This report starts with the *Method Description*. This chapter, firstly, presents the character, ragdoll, on which the muscle model will be applied to. Also, the muscle model is explained and analyzed, by determining the mathematical formulas and the method that has been used in order to create the musculoskeletal 3D human character and the user's real-time interaction. In the next chapter, *Resources*, all the tools that have been used to accomplish the project are presented. In the chapter *Experiment/Results*, the model and the user-interface are evaluated, by running an experiment, which is setting a target pose for the character to reach. In *Discussion* section, the potential future work suggestions are stated and the limitations of the project, regarding method and experiment, are presented.

2. Method Description

2.1 Ragdoll

The project focuses on creating a muscle model for a human character. So, a human ragdoll is used to apply the muscle forces. This ragdoll consists of a set of connected primitive shapes and has applied physics to it. On the ragdoll exist 12 joints, which represent the human joints and 13 rigid bodies that are aimed to create the mass properties and the body parts of the specific character. Due to the fact that the upper body of the character is not in the scope of the project, the lower body muscles have been constructed. The properties of the rigid bodies and the joints of the upper body are determined, in order to have the rigid bodies' appropriate mass, compared to the lower body. Also, the rigid bodies and joints are placed in the right mesh position. This assists the character to have realistic physics behavior.

2.2 Model Construction

In order to create a character that changes his pose, realistically, by altering the values of the muscle actuation, the creation of a muscle model close to the real human muscles is of great importance. The *Hill's-Type Muscle Model* is the most approved for this purpose, as mentioned in [2], [3], [5] and [9]. This model consists of three elements; a *contractile element*, a *parallel elastic element* and a *serial element*, as depicted in Figure 1.

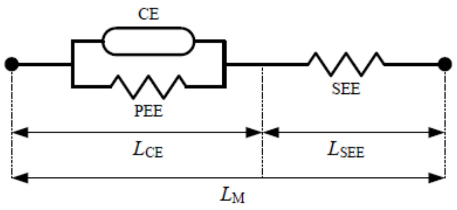


Fig. 1: Hill's-Type Muscle Model
Adapted from *"Flexible muscle-based locomotion for bipedal creatures." [3]*

The contractile and the parallel elastic elements produce the total force the muscle has, as shown in the following equation:

$$F_M = F_{CE} + F_{PEE} = F_{SEE} \qquad (1)$$

From Figure 1 it can be confirmed that the total force F_M is equal to F_{SE}, i.e. the total force produced by the serial elastic element. This is the case, because $F_{CE} + F_{PEE}$ and F_{SEE} forces are arranged in series [6]. Also, the total muscle length $L_M = (L_{CE} * \cos\theta) + L_{SEE}$, where θ is the pennation angle of the muscle. However, in this project all muscles appear to have pennation angle equal to 0, such as $\cos\theta = 1$.

The parallel elastic element is also producing force, called passive force. This is the force responsible for not letting the muscle overstretch, due to the force produced by the CE element F_{CE}. The equation of force produced by the parallel elastic element is [9]:

$$F_{PEE} = \frac{e^{k^{PEE}*(L_{CE}-1)/\varepsilon_0^M} - 1}{e^{k^{PEE}} - 1} \qquad (2)$$

where ε_0^M is the muscle strain produced by the PE element, due to F_{max} and k^{PE} is a dimensionless shape factor.

The serial elastic element (SEE) is described as a relationship between force and the length of the muscle [7]:

$$F_{SEE} = \begin{cases} \frac{\overline{F_{SEE}^{toe}}}{e^{k_{toe}} - 1}\left(e^{\frac{k_{toe}\varepsilon}{\varepsilon_{toe}}} - 1\right), & \varepsilon \leq \varepsilon_{toe} \\ k_{lin}(\varepsilon - \varepsilon_{toe}) + \overline{F_{SEE}^{toe}}, & \varepsilon > \varepsilon_{toe} \end{cases} \qquad (3)$$

The force of this element depends on the strain $\varepsilon = (L_{SEE} - L_{REST})/L_{REST}$, where L_{SEE} is the length of the SE element and L_{REST} is the rest length of the tendon. The normalized tendon force $\overline{F_{SEE}^{toe}}$ of the muscle is determined from literature to be 0.33 [11]. toe stands for the region of the muscle, where there is low stiffness. k_{toe} is a dimensionless shape factor, k_{lin} and ε_{toe} are constants dependent of the reference muscle strain ε_{ref}.

The contractile element is representing the fibers that a muscle has. It is the most important element of the model, because this is the part of the muscle that changes the muscle behavior, taking into consideration the actuation values given for the muscle. This force is also called “Active Force”. The following equation gives the amount of force produced by the contractile element (CE):

$$F_{CE}(a, L_{CE}, v_{CE}) = \alpha F_{max} f_L(L_{CE}) f_V(v_{CE}) \qquad (4)$$

In equation (4) α is the muscle actuation value, determined by the user interface and is always $0 \leq \alpha \leq 1$ and F_{max} is the maximum force a muscle can produce. The functions $f_L(L_{CE}), f_V(v_{CE})$ are the force-length and the force–velocity functions. Those functions are given in equations (3) and (4) respectively:

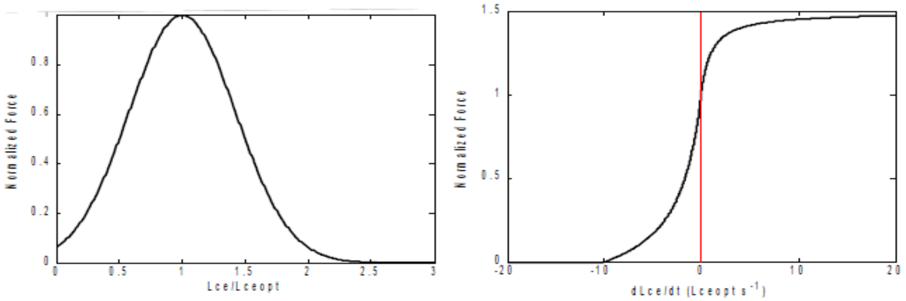


Fig. 2: Left: the Force-Length function of CE. Right: the Force-Velocity function of CE. Both functions are normalized. Adopted from *"Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults." [9]*

$$f_L(L_{CE}) = exp[-(\overline{L_{CE}} - 1)^2/\gamma] \qquad (5)$$

$$f_V(v_{CE}) = \frac{F_{CE}}{a * f_L(L_{CE})} \tag{6}$$

The previous equations, (5) and (6), are representing curved functions, depicted in Figure 2. In the force-length function, γ is a shape factor. Further information regarding this shape factor can be obtained from [10]. In the force-velocity function, v_{CE} is the current contraction velocity of the CE element.

One important aspect of this model is the calculation of the length L_{CE} and the velocity v_{CE}. This is the case, because both values are crucial in the performance of the muscle model, because these values are being used in $f_L(L_{CE})$ and $f_V(v_{CE})$. Therefore, the length L_{CE} and velocity v_{CE} are important for calculating in force F_{CE}, as shown in equation (4). The contraction velocity v_{CE} displays the rate of change of the length L_{CE}. Thus, the following derivative is being calculated, in order to obtain the contraction velocity v_{CE} :

$$v_{CE} = \frac{\partial\, L_{CE}}{\partial t} = (0.25 + 0.75a)v_{max}\frac{F_{CE} - af_L}{b} \tag{7}$$

$$b = \begin{cases} af_L + \dfrac{F_{CE}}{A_f}, & F_{CE} \leq af_L \\ \dfrac{(2 + \frac{2}{A_f})(af_L F_{CE}^{MAX} - F_{CE})}{F_{CE}^{MAX} - 1}, & F_{CE} > af_L \end{cases} \tag{8}$$

At equation (7) v_{max} is the maximum contraction velocity, and b is a parameter computed in equation (8). In this equation, A_f is the dimensionless shape factor of the force-velocity function and $F_{CE}^{MAX}$is the maximum force the CE element can produce, when the muscle is lengthening.

After the contraction velocity v_{CE} is calculated, the CE length L_{CE} is being updated and re-calculated. In order to achieve this, the contraction velocity is integrated, by using the Euler forward integration method, as presented in equation (9):

$$L_{CE} = \int v_{CE}\, dt = L'_{CE} + v_{CE} * dt \tag{9}$$

At the previous equation, the CE length of the previous timestep, L'_{CE} needs to be obtained.

2.3 Muscles

As soon as the muscle model is constructed, the muscles and their properties need to be determined. This is made, by using the *OpenSimSet*, a set of body muscles developed to represent a musculoskeletal 3D human character. This set has been scaled in [4], to represent the lower-body muscles of the ragdoll properties in the experimentation project. There are 46 lower-body muscles constructed, depicting the muscles of a human character. An example of a muscle can be seen in Figure 3 and the full body musculoskeletal system in Figure 4.

The muscle properties obtained from the *OpenSimSet* are:

- *Muscle attachment points*, points attached to specific bones. These points consist of the origin, starting point and the insertion, ending point of the muscle. However, in some muscles there are more than 2 muscle points, called via points.

- *Muscle maximum force* (F_{max}), the maximum force the muscle can produce, while contracted.
- *Optimal CE length* (L_{OPT}), the length of the CE element, where the maximum force can be produced.
- *Optimal SEE length* (L_{SLACK}), the slack length of the SEE, where the maximum force can be produced.

Apart from these properties, each muscle should contain one more property, which is the *maximum contraction velocity* (v_{max}).

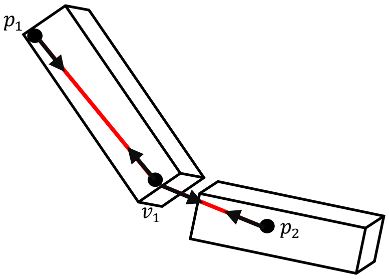


Fig. 3: Example of a muscle with 3 attachment points; origin, insertion and via point. The force is applied in all points, as shown on the black arrows.

Muscles are visualized as lines and are updated in every timestep. When the muscle is completely inactive, the muscle is red, whereas when the muscle is in full activity, its color is green. Depending on the actuation value, the muscle is changing color, as presented in the following formula:

$$C_M = C_{MI} - (C_{MI} - C_{MA}) * a \qquad (10)$$

In the previous equation (10), C_M is the color of the muscle, C_{MI} is the inactivity color and C_{MA} is the fully activity color.

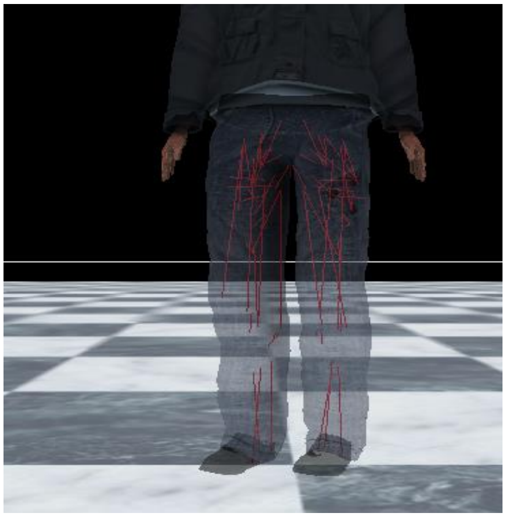
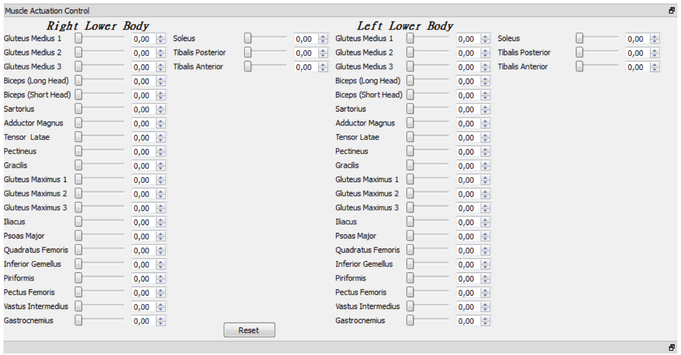


Fig. 4: Full system of lower-body muscles in complete deactivation.

2.4 Graphical User Interface

The GUI lets the user activate specific muscles, by setting their actuation values, as shown in Figure 5 by either:

- Moving the muscle’s slider, and
- Setting a value in the muscle’s spin box.



Right Lower Body

Gluteus Medius 1 0,00
Gluteus Medius 2 0,00
Gluteus Medius 3 0,00
Biceps (Long Head) 0,00
Biceps (Short Head) 0,00
Sartorius 0,00
Adductor Magnus 0,00
Tensor Latae 0,00
Pectineus 0,00
Gracilis 0,00
Gluteus Maximus 1 0,00
Gluteus Maximus 2 0,00
Gluteus Maximus 3 0,00
Iliacus 0,00
Psoas Major 0,00
Quadratus Femoris 0,00
Inferior Gemellus 0,00
Piriformis 0,00
Pectus Femoris 0,00
Vastus Intermedius 0,00
Gastrocnemius 0,00
Soleus 0,00
Tibalis Posterior 0,00
Tibalis Anterior 0,00

Left Lower Body

Gluteus Medius 1 0,00
Gluteus Medius 2 0,00
Gluteus Medius 3 0,00
Biceps (Long Head) 0,00
Biceps (Short Head) 0,00
Sartorius 0,00
Adductor Magnus 0,00
Tensor Latae 0,00
Pectineus 0,00
Gracilis 0,00
Gluteus Maximus 1 0,00
Gluteus Maximus 2 0,00
Gluteus Maximus 3 0,00
Iliacus 0,00
Psoas Major 0,00
Quadratus Femoris 0,00
Inferior Gemellus 0,00
Piriformis 0,00
Pectus Femoris 0,00
Vastus Intermedius 0,00
Gastrocnemius 0,00
Soleus 0,00
Tibalis Posterior 0,00
Tibalis Anterior 0,00

Fig. 5: The GUI of the muscle actuation control, in which the user can interact with the human character.

Apart from these two options, the user is able to reset the values of all the muscles, by pressing a corresponding button. This gives the user an opportunity to test many different values for the desired muscle, without the need of resetting every muscle independently.

3. Resources

This chapter deals with the technology been used, such as development environments and graphics engines. The selection of the following tools is based upon the related work that has been made that assisted in order to complete the project.

3.1 Programming Languages

The main programming language that is used for the project is C++ and the development environment in which the code is programmed is *Microsoft Visual Studio 2010*. However, for the GUI and the main function of the project python language is used and the *Python's Integrated DeveLopment Environment* (IDLE).

3.2 Additional Middleware

The first and most important tool that has been used is the *RAGE real-time animation and game engine*. It has been developed by the Virtual Worlds division of the Utrecht University. In addition, it has been used a GUI designer provided by python, *QT Designer*, in order to create the GUI that the user interacts with. Furthermore, in order to simulate physics phenomena, such as gravity and forces, in the project there had been used a physics engine, *Bullet Physics Engine*. *Bullet* is a very popular and successful physics engine, which has been used in many applications, such as videogames and movies. Finally, the tool to illustrate every visual results of the project, namely character, muscles and scene, is *OGRE Graphics Rendering Engine*.

4. Experiment/Results

4.1 Experiment

In order for the muscle model to behave realistic, the parameter values had to be determined. Table 1 presents these values, which are acquired from the literature [2] and [6] and [9].

The simulation works real-time. During the development of the project, some assumptions needed to be made. Firstly, in order to have initial values for the length of the CE and SE elements [3], the muscles were determined as in rest, following the equation:

$$L_{MUSCLE}^{rest} = L_{CE}^{optimal} + L_{SEE}^{slack} \qquad (10)$$

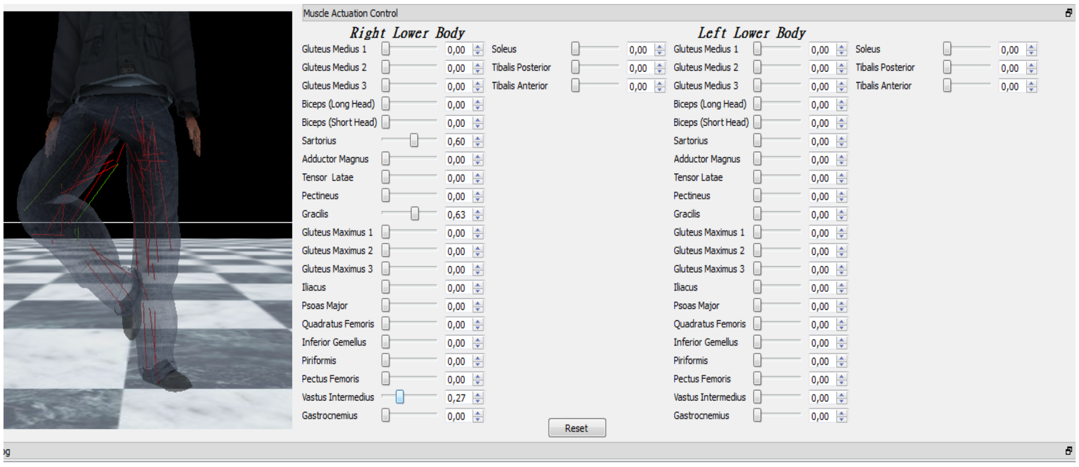
Parameter	**Value**
Ragdoll weight	$86\ kg$
Gravity	$9.81\ m/s^2$
Maximum shortening velocity $\boldsymbol{v_{max}}$	$-12\ L_{opt}/s$
Maximum normalized lengthening force $\overline{\boldsymbol{F_{CE}^{MAX}}}$	$1.4\ N$
Shape factor $\boldsymbol{\gamma}$	0.45
Shape factor $\boldsymbol{k^{PE}}$	4
Reference strain $\boldsymbol{\varepsilon_{ref}}$	$0.04\ L_{slack}$
Passive muscle strain $\boldsymbol{\varepsilon_0^M}$	0.6
Shape factor $\boldsymbol{k_{toe}}$	3
Shape factor $\boldsymbol{A_f}$	0.3
$\boldsymbol{k_{lin}}$	$1.712\varepsilon_{ref}$
$\boldsymbol{\varepsilon_{toe}}$	$0.609\varepsilon_{ref}$
Normalized tendon force $\overline{\boldsymbol{F_{SEE}^{toe}}}$	$0.33\ N$
Fully active muscle color $\boldsymbol{C_{MA}}$	RGB: $(0, 0.8, 0)$
Fully inactive muscle color $\boldsymbol{C_{MI}}$	RGB: $(1, 0, 0)$

Table 1: The model parameters.

Another assumption that is being used in the project is the fact that if the muscle length is beyond the 150% or less than 50% of their rest length, they don't produce any force [8]. Furthermore, when the fiber length L_{CE} of the muscle is less than 50% of the optimal fiber length, $L_{CE}^{optimal}$ the muscle cannot contract any longer, in a way that the realism of the muscles behavior is kept. Also, in order to create a system that has no singularities, another assumption needed to be handled. In equation (7) there is a case where f_L is 0, when the L_{CE} length becomes much smaller than the $L_{CE}^{optimal}$ length. However, in order to avoid this singularity, the value of this variable is no smaller than 0.01.

4.2 Results

The musculoskeletal model is constructed and all the assumptions are taken into consideration. The user is now able of activating certain muscles and observing, or not, the character movement. It appeared that certain muscles, the ones close to the pelvis of the body, are producing force, but the movement produced by these muscles are not having a great impact in the overall character movement. All the muscles have been tested, in order to evaluate if the GUI interacts properly with the corresponding muscles. The color scheme is also performing as expected. To verify the usability of the framework and the GUI, we used it in order to assist the character achieving a certain pose. A pose that has been selected to show clearly whether the system performs realistically or not, is the knee flexion. Figure 6 shows the simulation results.



Right Lower Body

Gluteus Medius 1	0,00	Soleus	0,00
Gluteus Medius 2	0,00	Tibalis Posterior	0,00
Gluteus Medius 3	0,00	Tibalis Anterior	0,00
Biceps (Long Head)	0,00		
Biceps (Short Head)	0,00		
Sartorius	0,60		
Adductor Magnus	0,00		
Tensor Latae	0,00		
Pectineus	0,00		
Gracilis	0,63		
Gluteus Maximus 1	0,00		
Gluteus Maximus 2	0,00		
Gluteus Maximus 3	0,00		
Iliacus	0,00		
Psoas Major	0,00		
Quadratus Femoris	0,00		
Inferior Gemellus	0,00		
Piriformis	0,00		
Pectus Femoris	0,00		
Vastus Intermedius	0,27		
Gastrocnemius	0,00		

Left Lower Body

Gluteus Medius 1	0,00	Soleus	0,00
Gluteus Medius 2	0,00	Tibalis Posterior	0,00
Gluteus Medius 3	0,00	Tibalis Anterior	0,00
Biceps (Long Head)	0,00		
Biceps (Short Head)	0,00		
Sartorius	0,00		
Adductor Magnus	0,00		
Tensor Latae	0,00		
Pectineus	0,00		
Gracilis	0,00		
Gluteus Maximus 1	0,00		
Gluteus Maximus 2	0,00		
Gluteus Maximus 3	0,00		
Iliacus	0,00		
Psoas Major	0,00		
Quadratus Femoris	0,00		
Inferior Gemellus	0,00		
Piriformis	0,00		
Pectus Femoris	0,00		
Vastus Intermedius	0,00		
Gastrocnemius	0,00		

Fig. 6: By activating *Sartorius, Gracilis and Vastus Intermedius* muscles the knee flexion is achieved.

5. Discussion

5.1 Conclusion

In this experimentation project, we managed to construct a framework, where the user is able to activate lower body muscles of a 3D human character. The *Hill's type muscle model* was implemented, in order to build a musculoskeletal model. As illustrated in the results section, the human character, is able to reach a target pose, by contracting certain muscles, determined by the user.

5.2 Future Work

As presented in the method description section, the user is able to contract the lower body muscles. An interesting extension to this project would be to create a muscle model for the whole human body. Also, an interesting part of musculoskeletal animation, which is not covered in this experimentation project, is to create different muscle models and try to evaluate the most suited for a human character. A variation of the *Hill's type muscle model* is presented in [REF] and it could be used in order to compare with the one implemented in the experimentation project. The force applied to the muscles could be a useful value of comparison, between the different muscle models. Another subject regarding the muscle models would be to construct GUIs for different characters, such as animals or fictional characters. Lastly, a very interesting and promising topic regarding the interaction of user and musculoskeletal model, which is out of scope at this experimentation project, would be to develop a library, where the user assigns tasks to the 3D character. Then, the character performs the tasks, by activating specific muscles. However, this case involves joint control as well, so this is the reason that the project seems out of scope.

5.3 Limitations

Although the experimentation project is fulfilling the goal determined in the introduction, some limitations have to be stated. Firstly, in some cases the muscles are out of the mesh character, having some visualization issues. This is the case, because the muscle positions in the physics engine (Bullet) is different from the one in the graphics engine (Ogre). Also, this is not adjustable, because the *OpenSim Skeleton* is determined and scaled by [4] to have the muscle attributes in the specific human character that this experimentation project is using. Another drawback is the assumption regarding the rest length of the muscle. Even though the model assumes that the muscle initially has optimal lengths for both fiber and tendon, the actual muscle length is different from the assumption. This is the case, because in reality some muscles need to be actuated, in order to let the human character standing and not falling. If there were initial fiber and tendon lengths different, the muscle length would be more precise.

6. References

1. Sueda, Shinjiro, Andrew Kaufman, and Dinesh K. Pai. "Musculotendon simulation for hand animation." *ACM Transactions on Graphics (TOG)*. Vol. 27. No. 3. ACM, 2008.
2. Geyer, Hartmut, and Hugh Herr. "A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities."*Neural Systems and Rehabilitation Engineering, IEEE Transactions on* 18.3 (2010): 263-273.
3. Geijtenbeek, Thomas, Michiel van de Panne, and A. Frank van der Stappen. "Flexible muscle-based locomotion for bipedal creatures." *ACM Transactions on Graphics (TOG)* 32.6 (2013): 206.
4. Laclé, Francis. "A Literature Review for the Master Project: Real-Time Musculoskeletal Model for Injury Simulation on 3D Human Characters." *Literature Review Msc Project*, 2013.
5. Rosen, Jacob, Moshe B. Fuchs, and Mircea Arcan. "Performances of Hill-type and neural network muscle models—Toward a myosignal-based exoskeleton."*Computers and Biomedical Research* 32.5 (1999): 415-439.
6. Geyer, Hartmut, Andre Seyfarth, and Reinhard Blickhan. "Positive force feedback in bouncing gaits?." *Proceedings of the Royal Society of London. Series B: Biological Sciences* 270.1529 (2003): 2173-2183.
7. van Ingen Schenau, G. J. "An alternative view of the concept of utilisation of elastic energy in human movement." *Human Movement Science* 3.4 (1984): 301-336.
8. Buchanan, Thomas S., et al. "Neuromusculoskeletal modeling: estimation of muscle forces and joint moments and movements from measurements of neural command." *Journal of applied biomechanics* 20.4 (2004): 367.
9. Thelen, Darryl G. "Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults." *Journal of biomechanical engineering* 125.1 (2003): 70-77.
10. Gordon, A. M., Huxley, A. F., and Julian, F. J., “The Variation in Isometric Tension with Sarcomere Length in Vertebrate Muscle Fibres." *J. Physiol*, 184 (1966): 170–192.
11. Proske, U., and Morgan, D. L., 1987, "Tendon Stiffness: Methods of Measurement and Significance for the Control of Movement. A Review. “*J. Biomech* 20 (1987): 75–82.