A novel motion simulator technique for generating a realistic physical human jump movement

Przemysław Prokopow[1]
1) Computational Biomechanics Unit
The Institute of Physical and Chemical Research (RIKEN)
Wako-shi, Japan
przem@post.pl

Krzysztof Pomorski[2,3]
2) Faculty of Physics
University of Warsaw
Warsaw, Poland

3) Institute of Physics
Jagiellonian University
Krakow, Poland
kdvpomorski@gmail.com

Abstract—**The aim of this study was to develop and validate a novel motion simulator technique for generating a realistic physical human jump movement. To achieve this goal, we used a physically modeled computer simulation that employed a 3D neuro-musculo-skeletal model and a forward dynamics approach. The simulation consisted of all the important features of the human body and physics salient for human jump movement. The muscle excitation pattern was calculated using feed-forward artificial neural networks. The only input given to the model was a set of parameters defining a starting position and a jump high. The output of the simulation generated realistic and natural-looking motion pattern. We found that the forward dynamics approach is capable of generating physical natural-looking human jump motion. In the future, these simulations can be extended by defining other motion patterns and can be used in computer animation for generating detailed human movements.**

Keywords: animation; physical simulation; human jump; motion generation

I. Introduction

The modeling and animation of human movement is a complex task due to the complexities of human body structure and dynamics. One example of such a complex movement is a vertical jump motion. This whole-body movement is characterized by high forces acting on all body segments, which are generated by high joint torques and the resulting joint velocities. Consequently, a jump motion is characterized by complex dynamics and some physical details that are difficult to capture using conventional methods.

When it is necessary to generate a realistic motion with all the details (for example, a powerful ankle plantar flexion near take-off), it is necessary to model it using a new approach capable of generating a detailed physical motion. Such details are important because they create a realistic impression of the motion. However, it is difficult to model the details of physical motion with conventional animation methods. The human body has many degrees of freedom and complex non-linear dynamics that are difficult to mimic with conventional animation methods. These limitations in conventional methods result in methods that rely on chance to model physical realism or do not reproduce every detail of a human motion. Therefore, a new technique that could overcome the shortcomings of conventional methods would be advantageous for computer graphics.

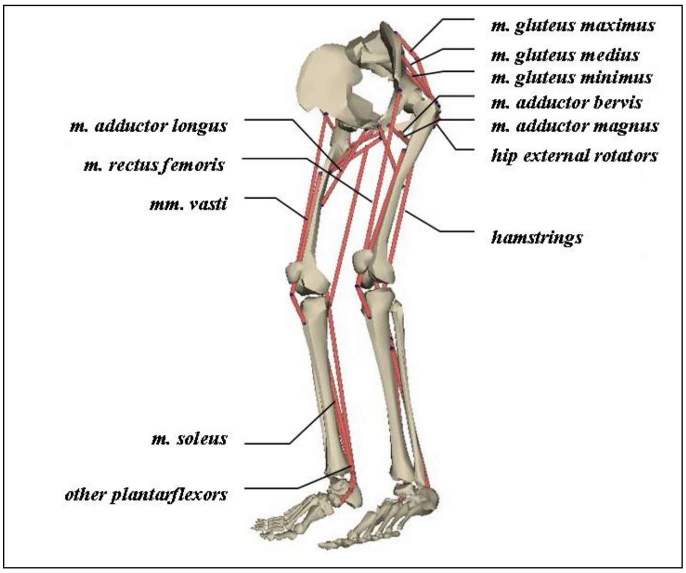

Figure 1. The neuro-musculo-skeletal model used in this study.

A number of techniques have been developed that use physically based simulation to generate motion for animation. Most of the previous research has focused on the issue of designing a simulation method for a particular type of physical phenomenon, such as cloth movement, fluid dynamics, hair, and other passive objects.

Those previous studies [1, 2, 3] used a dynamic physical simulation, and in some of the studies they use a physiological human body model, however instead of neuro-muscular physical model, models used in those previous studies they used a controllers methodology derived from robotics, or the control over the multi-body model was hand-tuned. That approach was proved to be successful in generating realistic physically based animation [1, 2]. However the approach used previously had some important limitations. Due to the fact that in the previous studies control was not physically based the control could be applied only to single movement. Moreover generated in those studies motion, even though physically sound did not account for human neurophysiology and therefore generated movement did not look natural.

In a computer graphics, up to now, a complex and dynamic phenomenon, such as human movement, has not been simulated with a physical equations and a neuro-musculo-skeletal model of a human body.

One technique that can be used to generate a realistic physical motion is a physically modeled computer simulation. Such physically based simulations have been developed and used in research areas of the life science (e.g., biomechanics, medical engineering and cybernetic biology). To date, a few researchers have undertaken a large effort to develop such simulations [4, 5]. However, the usefulness of physically-based models of human movement in computer graphics is very limited. This limited use is because such simulation of human movements is inherently complex [6]. Hence, the technique requires an enormous number of parameters and mathematical functions to be defined in a physical model.

Table 1. The parameters of the musclotondons

Muscle	Peak Isometric Force (N)	Pennation Angle (deg)	Optimal Fiber Length (m)	Tendon Slack Length (m)
GMAXI	1,883.4	5.0	0.1420	0.1250
GMEDI	1,966.2	8.0	0.0535	0.0780
GMINI	848.6	1.0	0.0380	0.0510
ADDLO	716.0	6.0	0.1380	0.1100
ADDMA	1,915.8	5.0	0.0870	0.0600
ADDBR	531.1	0.0	0.1330	0.0200
HEXRO	1,512.0	0.0	0.0540	0.0240
RECTF	1,353.2	5.0	0.0840	0.4320
HAMST	3,053.6	15.0	0.0800	0.3590
VASTI	6,718.3	3.0	0.0870	0.3150
GASTR	2,044.4	17.0	0.0450	0.4080
SOLEU	5,880.7	25.0	0.0300	0.2680
OPFLE	3,137.1	12.0	0.0310	0.3100

The parameters of the musclotondons used in the model are shown here. The abbreviations are GMAXI (m. gluteus maximus), GMEDI (m. gluteus medius), GMINI (m. gluteus minimus), ADDLO (adductor longus), ADDMA (m. adductor magnus), ADDBR (m. adductor bervis), HEXRO (hip external rotators), HAMST (hamstrings), RECTF (m. rectus femoris), VASTI (mm. vasti), GASTR (m. gastrocnemius), SOLEU (m. soleus), and OPFLE (other plantar flexors).

The aim of this study was to develop a motion simulator that would be able to generate a realistic physical human motion and, at the same time, be easy for an animator to use. To achieve the latter we aimed to describe the motion specified in the simulator in a straightforward and high-level manner.

To implement such a physically based motion simulator, we set three goals. The first goal was to develop a physical simulation of human movement based on a neuro-musculo-skeletal model and optimal control theory. The second goal was to develop an easy to use and intuitive user interface in which the motion would be parameterized in straightforward manner and which would calculate and setup all the parameters needed for physical simulation of the motion. The final, third goal was to implement this motion simulator to generate a realistic human jump motion and evaluate its performance for generating realistic physically-modeled human movement.

The reason a vertical jump was generated in this study is that a vertical jump is especially difficult to generate with conventional techniques. This difficulty is due to the complex dynamics of the motion and physiological details, such as powerful segment interactions, which characterize jump motions. This motion, however, is only an example used in the study, and in the future, the motion simulator proposed here might be used to generate virtually any motion.

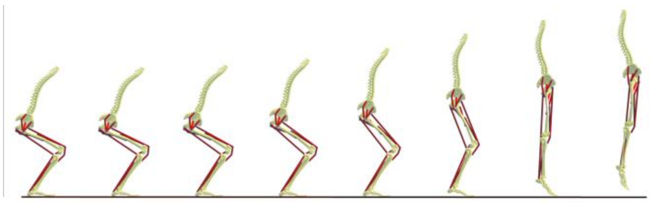

Figure 2. The neuromusculoskeletal model used in this study.

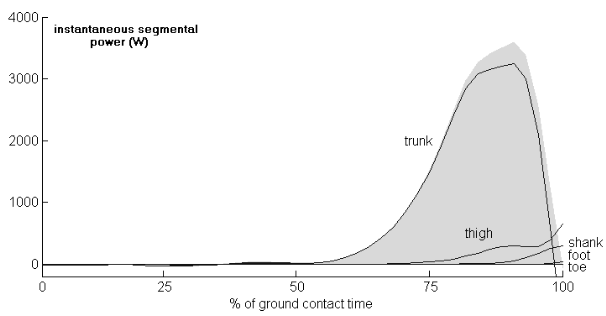

Figure 3. The instantaneous segmental power during propulsion in the optimal jump. The ground contact time is expressed relative to the time interval between the beginning of the simulation and take-off.

II. METHODS

The aim of this study was to develop a new motion simulator capable of generating a realistic motion based on physical equations and a neuro-musculo-skeletal model of human body.

The musculoskeletal model used in this study is presented in Figure 1. It was three-dimensional and consisted of nine rigid body segments (head-arms-trunk, right and left upper legs, right and left lower legs, right and left feet and right and left toes). The model had 20 degrees-of-freedom. It was free to make and break contact with the ground, where the foot-ground interaction was modeled as exponential springs-dumpers that were placed under each foot [4].

For the starting position, the body was assumed to be in a static squat position with the heels flat on the ground. Passive elastic joint moments were applied to each joint. The passive elastic moments of the hip extension/flexion, knee extension/flexion and ankle dorsi/plantar flexion were applied to the model as a set of equations reported by Riener and Edrich [7]. The remaining passive moments were applied as the exponential equations reported by Anderson and Pandy [4].

The model included 26 major muscle groups of lower extremities. These were modeled as Hill-type muscles, which consist of a contractile element and a series elastic element [7]. In this study the formula for the muscle force produced during isometric contraction was adopted from [9 & 10]. The fraction of the force relative to maximal muscle force is given by (1).

$$F_{isom} = c_1 \cdot \left(\frac{L_{CE}}{L_{CEopt}} \right)^2 + c_2 \cdot \frac{L_{CE}}{L_{CEopt}} + c_3 \qquad (1)$$

where

$$c_1 = \frac{-1.0}{width^2} \qquad (2)$$

$$c_2 = -2.0 \cdot c_1 \qquad (3)$$

$$c_3 = 1.0 + c_1 \qquad (4)$$

Muscle paths and tendon slack length were derived from the work of [5]. Muscle parameter values, such as the optimal contractile element length, maximal isometric force of contractile element, and pennation angle, were derived from the data reported in other literature [11]. The values of these parameters are presented in table 1.

Neural control consisted of the activation patterns of 13 muscles. Because the vertical jump has been assumed to be bilaterally symmetric, identical neural control signals were sent to muscles in both legs. Each muscle activation pattern was specified by three variables: onset time, offset time, and magnitude of muscle activation. A first-order differential equation described the delay between a muscle's activation and its active state [12]. The formula of the equation describing activation dynamics is given as follows:

$$\dot{q} = \frac{u - q}{\tau(u, q)} \qquad (5)$$

In this equation above, $u \in [0, 1]$ is the neural excitation and the term $\tau(u, q)$ is a time constant. The time constant consists of two parts: ascending and descending part. The two conditions for calcium release and uptake must be defined separately, one for each (ascending and descending) part:

$$\tau(u,q) = \begin{cases} (\tau_{rise} - \tau_{decl}) \times u + \tau_{decl} & for\ u \geq q \\ \tau_{decl} & otherwise \end{cases} \qquad (6)$$

In this equation above, τ_{rise} and τ_{decl} are the activation and deactivation time constants respectively. In this study I used specific values of τ_{rise} = 55 ms for ascending time and τ_{decl} = 65 for descending time [13].

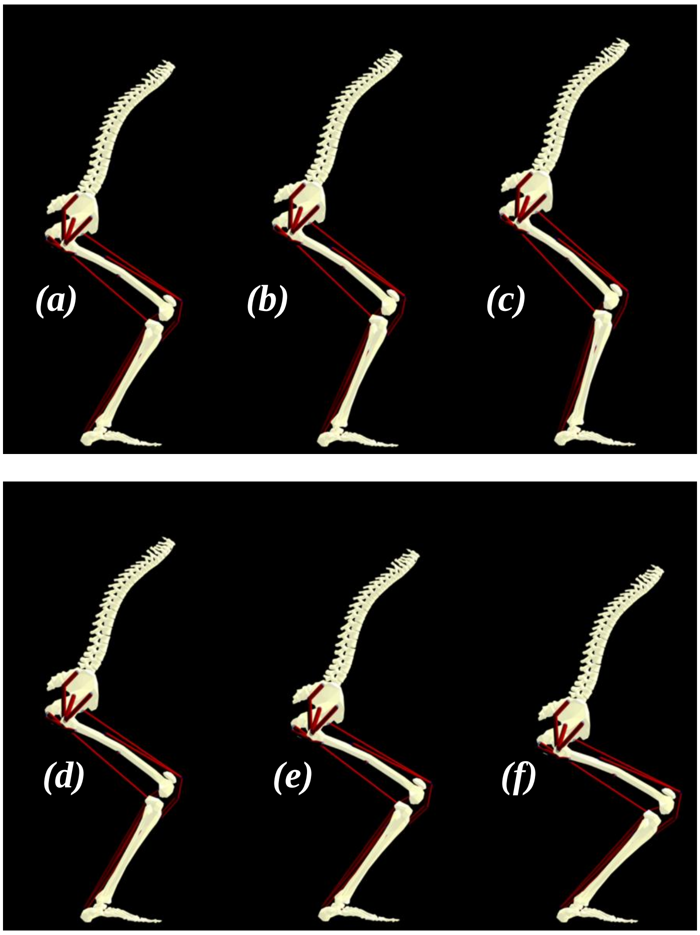

Figure 4. The figure shows the generated starting position where the body's center of gravity was first increased (b & c) and then lowered by 10 and 20% (d & e). The original starting position (a) is depicted as a reference. It can be seen that the generated position looks natural.

The optimal muscle activation pattern was found through Bremermann's optimization method [14], in which a maximum height reached by the body's center of mass (BCM) was used as an objective function that is described by a formula (7) in which *m* is a total mass of musculo-skeletal system and *h* is an instantaneous vertical displacement of the BCM.

$$E_{\mathrm{MAX}} = \max \int_{t=0}^{\lambda} \left(m \cdot g \cdot h + \frac{1}{2} \cdot m \cdot \dot{h}^2 \right) dt \qquad (7)$$

In the optimization process, onset time and offset time of muscle activation ranged between zero (simulation start time) and 0.5 s (simulation end time) and the magnitude of muscle stimulation had a range of zero (no excitation) to one (muscle fully excited). Hence, virtually any value in those ranges could be chosen as a result of the optimization process.

To make a straightforward and easy to use motion simulator, the input to the simulator consists of only two high-level parameters that precisely, but simply, describe a generated jump motion. The first of the inputs describes a starting position and the second one describes a jump height.

The first input parameter describes how deep a squat position from which the jump starts. In the real word, people start jumping from different squat positions, and to implement a realistic and practical motion simulator of a vertical jump, it is important to define different starting positions. To calculate the value of this input parameter, at first, the simulator calculates a vertical position of the BCM, and then, the value is expressed relative to a vertical position of the BCM in a typical squat position, from which people start jumping. The value of the BCM in a typical squat position was calculated based on the experimental data published by Anderson [4]. In the simulator, when this first input parameter has a value of 100%, it means the starting squat position is set to a typical pose, whereas in the case when the value of this parameter is lower than 100% it means that a squat position is deeper than in a typical jump (that is, the BCM is lower relative to a typical jump position). Based on this single scalar input, it is possible to generate the starting position of a physical model and then generate a realistic motion.

The second input parameter to specify a jump motion was jump height. With the goal of making it intuitive for an animator to specify the value of the second input parameter, its value is given as a percentage of an expected jump height relatively to maximum jump height. In this specification, any value between 50% and 100% can be used, where 100% means a jump where jump height is maximal and 50% means a jump where the jump height is a half of the maximal height.

In this model, neural control was represented as a muscle excitation pattern sent to each muscle. The pattern consisted of onset time, offset time of muscle activation and magnitude of muscle stimulation [15]. The excitation pattern was first calculated using feed-forward artificial neural networks (MATLAB, The MathWorks Inc 2007) and then given as an input to the feed-forward physical simulation. Neural networks used in this study had three different layers with a different number of neurons in each. In the network that we used, each neuron received its inputs directly from the previous layer or from the user parameter and sent its output directly to units in the next layer or directly generated the

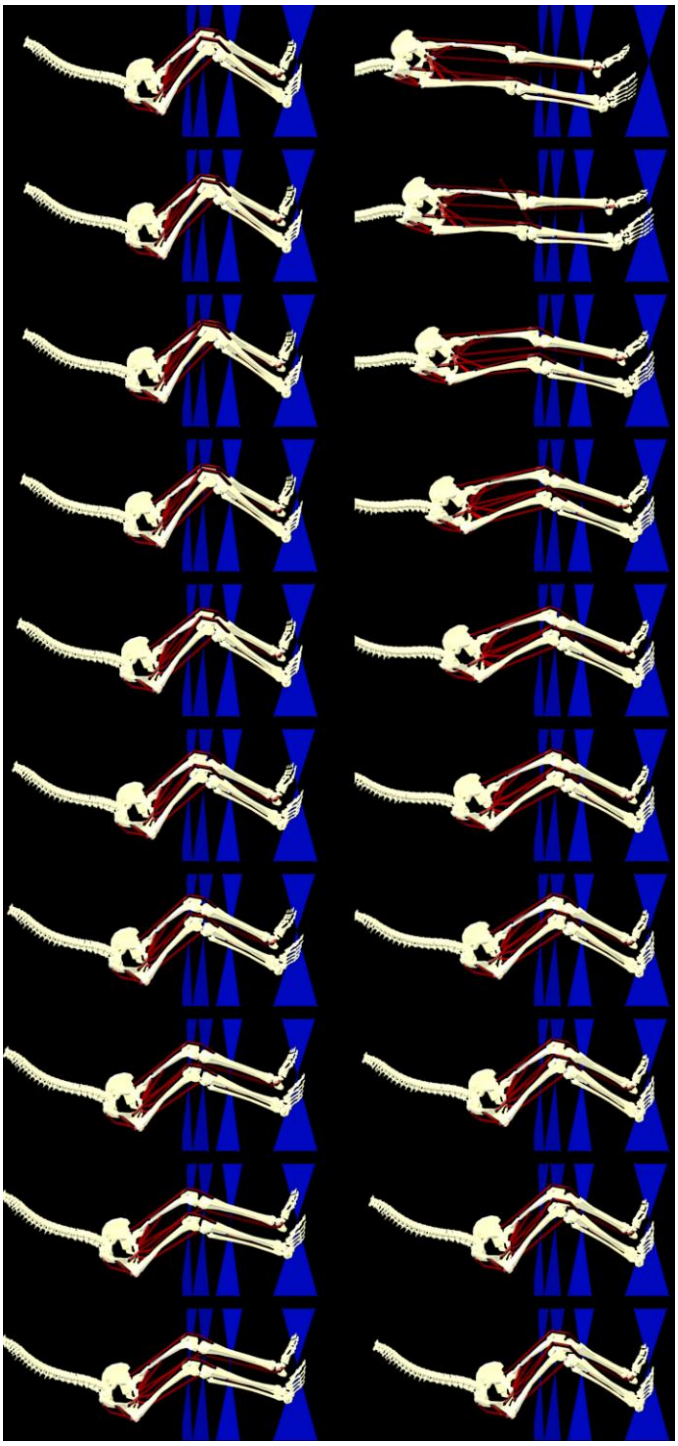

Figure 4. The jump motion generated from the erect starting squat position where the BCM was 20% higher than the original starting squat position. All figures are spaced equally in time (Δt=25 ms). It can be seen here that the physical motion generated in the study was natural-looking and smooth.

muscle stimulation pattern. To train a neural network (i.e. to adjust weights and a bias of each neuron), we used the Levenberg-Marquardt back-propagation training algorithm [16].

III. RESULTS

In this study, a human vertical jump motion was successfully generated with the proposed motion simulator. The natural-looking and smooth human motion presented is as rendered animation frames in Figure 2. In the simulator, a physical simulation and neuro-musculoskeletal model were used to successfully generate the realistic physical vertical jump motion. To quantitatively validate the human vertical jump generated in this study, the motion parameters were compared with experimental data from other literature and we found that neural control, muscle exertion patterns, and jump motion completely reproduce real human motions reported in previous studies.

The duration of the push-off phase in this simulation was found to have approximately the same duration (290 ms) as reported in other literature [15].

Instantaneous segmental power in the optimal vertical jump is presented in Figure 3. It can be seen that the trunk was the main power generator in a jump, but shortly before take-off, the power generated by distal segments was growing up quickly. At the same time, shortly before take-off, the power generated by the trunk decreased rapidly. The reason for this result was that at the instant, ~20 ms, before take-off the hip joint was already fully extended; therefore, the angular inertia at the hip joint could not further contribute to the upward acceleration of the BCM. These results are consistent with other studies describing an optimal control solution in vertical jumps [4, 16, 17].The ground reaction forces (GRF) recorded in the optimal jump are plotted in Figure (5.8). The peak in the GRF was recorded at the BCM position 0.82 m and were equal to 2.8 body weights. The GRF recorded in this study remain in agreement with experimental data reported in the literature [15].

The greatest forces in the optimal jump were developed by the mm. vasti, m. soleus and other plantar flexors. The monoarticular knee extensor muscles (mm. vasti) developed force first and then the biarticular hamstrings and monoarticular hip extensor muscles. Ankle plantar flexors (monoarticular and biarticular) developed force later. The last muscle that developed force in jump was the biarticular m. rectus femoris.

A kinematics of the jump motion is presented in the Figures 4 and 5. As a result of the parametric calculation, the motion simulator was able to calculate any position specified by user input. The generated positions are presented in Figure 4. All starting positions generated in trials reproduced a range of physiologically correct squat position. Figure 5 presents a kinematics of the jump motion, where the objective of jump height was set to 80% of the maximal jump height. It can be seen that the jump motion is smooth and realistic. Similarly, Figure 6 presents a kinematics of the jump motion generated from an erect starting squat position where the BCM was 30% higher than the original starting squat position and the jump height was set to 20% lower than maximal. By comparing the joint angles of the simulated jump motion with experimental results, we concluded that the kinematics pattern completely reproduced the jump motion pattern observed in a human body.

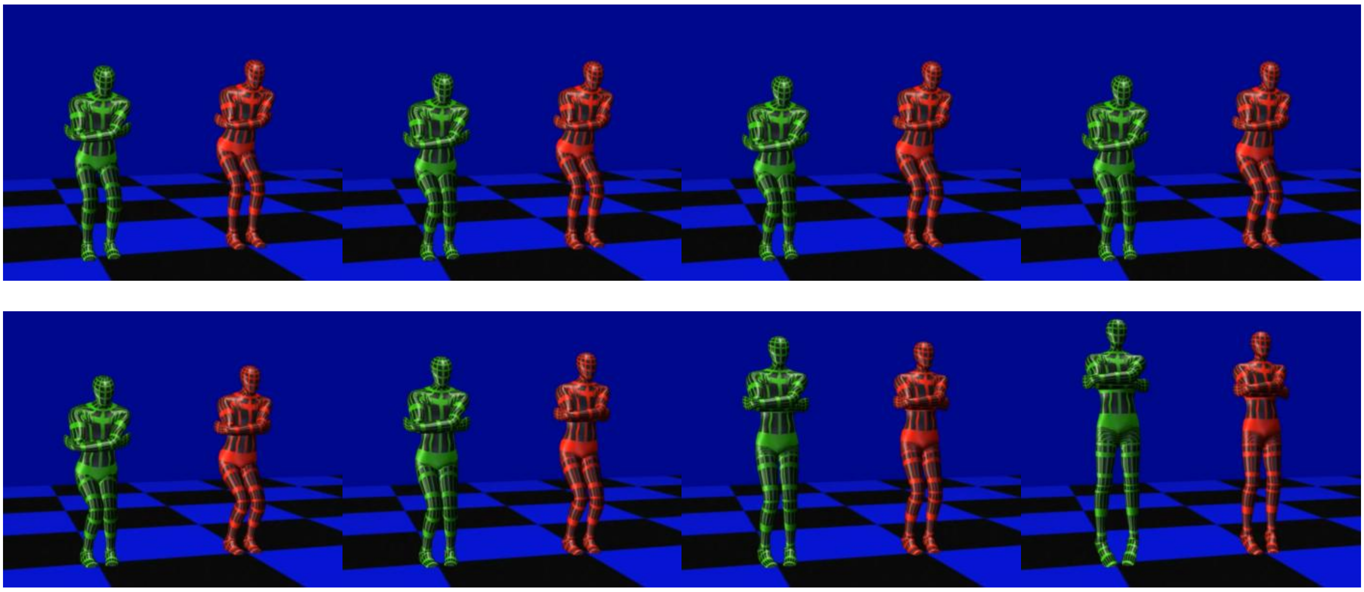

Figure 5. The jump motion generated from an erect starting squat position where the BCM was 30% higher than the original starting squat position and the jump height was set to 20% lower than maximal. The original jump is shown in a green color on the left side and the jump from an erect position is shown in a red color on right side. All the figures are spaced equally in time (Δt=60 ms). It can be seen here that the physical motion generated in the study was natural-looking and smooth.

IV. DISCUSSION

In this paper, we presented a new technique to generate a realistic human movement based on a physically-based model. In the technique presented here, all the parameters of the physical model were simplified by artificial neural networks. Therefore, the technique was intuitive and straightforward to use in computer graphics. As a result, we succeeded in making a physical motion simulator, which generated realistic motion and which was also easy to use even for unskilled animators. Finally, the technique proposed here was found to be computationally inexpensive for a user.

The simulator uses a straightforward interface through which an animator specifies a motion and as output the simulator generates a realistic physically-based motion. In the computations, the motion simulator uses a detailed neuro-musculo-skeletal model, whereas the motion is generated through physical equations in a completely feed-forward manner. The simulation technique introduced in this study is important for computer graphics and can be applied to computer games, computer generated animation, and movie production.

One of the most important problems in computer graphics is the generation of realistic human movement. Even though a number of techniques have been proposed to generate a realistic motion, they can be used only in some animations; therefore, the proposed techniques have no ability to generate a wide range of realistic motions. Most of the methods that have the ability to produce realistic motions are based on motion capture and allow the motion capture to be only slightly modified [2]. However, the simulation technique used in life science and in this study was able to create a realistic and precisely specified motion. Moreover, the technique developed in this study is based completely on a computer simulation and did not require any captured data. Therefore, we conclude here that the motion simulator technique proposed in this study is a very powerful tool to generate realistic physical human motion.

In this study, a generated motion is based on a completely mathematical, physical and biomechanical model, which describes human physiology and physical laws. A Simulation technique based on the same physical modeling technique was also previously used to generate realistic human movement in science [4]. However, those simulations could not be used in computer graphics for three major reasons: the complexity of the models they used, the large computational expense, and the model developed for scientific purposes were inflexible and could not be easily used to generate more than a single movement. In this study, the same scientific approach is used for computer graphics. To implement a physical simulation method in computer graphics, a new motion simulator was proposed, in which all the parameters of a physical model are calculated automatically with artificial neural networks. The network has the capability to find input parameters for the physical simulation, which in practice, allowed this study to make a physical simulation easy and intuitive to use because all the expert knowledge required to find a motion was recorded by a training process used to build the artificial neural network. Moreover, in this study, we propose an easy to use interface, which helps an animator specify a motion and automate processes generating the motion.

The simulator technique presented here demonstrated that our approach to generate human motion based on a complete model of physical and physiological body can be successfully used to generate realistic physical human movement. Moreover, due to the implementation of the interface based on artificial neural networks, it was possible to completely automate the procedure of setting up the parameters of a physical simulation and a neuro-musculo-skeletal model. Therefore, the motion simulator is straightforward and easy to use for an animator.

In future work, the simulator technique developed in this study should be extended by adding some of the most common human movements so that it can be used to generate a wide range of human movements. Consequently, it could be widely used in computer animation tasks, such as computer generated movies and animations, to generate realistic human movement.

ACKNOWLEDGMENT

We would like to thank Professor Ryutaro Himeno at RIKEN & University of Tokyo for his constructive comments on our research.

REFERENCES

[1] J. Hodgins, N. Pollard, 1997. Adapting simulated behaviors for new characters. In proceedings of ACM SIGGRAPH 1997, USA.

[2] N.I. Badler, C..B. Phillips, B.L. Webber, 1993. Simulating Humans: Computer Graphics Animation and Control, Oxford University Press, New York.

[3] D. Pai, 1990. Programming anthropoid walking: Control and simulation", Tech. Rep. 90-1178, Cornell Computer Science, 1990.Alias, 2004, Learning Maya 6 - Foundation. Berkeley, CA: Sybex.

[4] F.C. Anderson, M.G. Pandy, 1999. A dynamic optimization solution for vertical jumping in three dimensions. Computer Methods in Biomechanics and Biomedical Engineering 2, 201 231.

[5] S.L. Delp, J.P. Loan, M.G. Hoy, F.E. Zajac, E.T. Topp, J.M. Rosen, 1990. An interactive graphicsbased model of the lower extremity to study orthopaedic surgical procedures. IEEE Transactions in Biomedical Engineering, 37, 757-767.

[6] F.E.Zajac, 2002. Understanding muscle coordination of the human leg with dynamical simulations. Journal of Biomechanics 35, 1011-1018.

[7] R. Riener, T. Edrich, 1999. Identification of passive elastic joint moments in the lower extremities. Journal of Biomechanics 32, 539-544.

[8] A.V. Hill, 1938. The heat of shortening and the dynamic constants of muscle. Proceedings of the Royal Society of London. Series B, Biological Sciences, 126 (843), 136-195.

[9] A.J. van Soest, M.F. Bobbert, 1993. The contribution of muscle properties in the control of explosive movements. Biological Cybernetics 69(3), 195-204.

[10] G.K. Cole, B.M. Nigg, A.J. van Den Bogert, K.G. Gerritsen, 1996. Lower extremity joint loading during impact in running. Clinical Biomechanics 11, 181-193.

[11] J.A. Friederich, R.A. Brand, 1990. Muscle fiber architecture in the human lower limb. Journal of Biomechanics 23, 91-95.

[12] J.M. Winters, and L. Stark, 1988. Simulated Mechanical Properties of Synergistic Muscles Involved in Movements of a Variety of Human Joints, Journal of Biomechanics, 12, 1027-1042.

[13] M.G.Pandy, F.E.Zajac, 1991. Optimal muscular coordination strategies for jumping. Journal of Biomechanics 24, 1-10.

[14] H. Bremermann, 1970. A method of unconstrained optimization. Mathematical Biosciences, 9, 1-15.

[15] M.F. Bobbert, A.J. van Soest, 1994. Effects of muscle strengthening on vertical jumps: a simulation study. Medical Science Sports Exercise 27, 1012-1020.

[16] M. Hagan, and M. Menhaj, 1994. Training Feedforward Networks with the Marquardt Algorithm, IEEE Transactions on Neural Networks, 5 (6), 989-993.

[17] Pandy, M.G., Zajac, F.E., Sim, E., Levine, W.S., 1990. An optimal control model for maximum height human jumping. Journal of Biomechanics 23, 1185-1198.