

Study Thesis: Jumping Humanoid Simulink Simulation With Three Hill-Type Muscle Models

C. A. Seidel

Supervisors: A. Jahr, A. Batos

Hochschule Düsseldorf, 40476 Düsseldorf, Germany
anton.seidel@outlook.com

Abstract. This paper investigates the implementation of a human body model for the simulation of a jumping motion in Simulink. It contains details to get an insight into modern biomechanics without previous experience. In the simulation, a Hill-Type muscle is used to generate joint momentum with a phenomenologically realistic relationship between muscle length and force. Three of these biomechanical units are implemented to perform a leg motion, initiated at a *Hip* joint of the body. To build this simulation, explorations on how to dimension body parts, how to derive corresponding centres of mass and inertias, to connect limbs with the used muscles and to stimulate them, are executed. The reason for the publication is to give others with no background in this content an application for trials and a start for further findings in modern biomechanics. Realistic simulations have the potential to support research for intuitively usable and in a wide range of motion applicable prostheses. Injured people who have lost a part of their body may gain better motivation towards their life with a new highly functional prosthesis. The opportunity to help people is a motivation to search for feasible solutions and to excite others. The simulation published with this article is one of many solutions to design a humanoid simulation. The following paper guides through the implemented build and explains its range of application.

Keywords: biomechanics · humanoid Simulation · Hill-Type model · motion study · Simulation of Mechanical Systems · Engineering Conferences.

1 Introduction

The understanding of how the human body moves is increasing towards the possibility of imitating reality. Simulation results of movement sequences with synthetic muscle models come closer to real electric detectable muscle activity [6] [3]. This leads to new research topics and conclusions on how future prosthetics can be improved in their functionality. At the Massachusetts Institute of Technology (MIT), Hugh M. Herr invents humanoid robotic body parts for the lower limbs which are able to be controlled by the users' muscle stimulus [2].

Current findings show how a bidirectional interaction between prostheses and human subjects can be managed [2]. This means, that the user not only sends neural information to the prosthesis but also receives information on its current state (position, speed, torque) [2]. This, for a normal functioning body part natural sensible state, is called proprioception, which is not the present state of the art for commercial prosthetic technology [2]. Findings on a new amputation technique are the basis of this new technology [10]. Merged with new methods to sense muscle activity and new control systems for prosthetics, the technology for an intuitive usable anthropoid robotic part is no longer a futuristic fictionality [7]. Hugh M. Herr calls his findings "Neuro Embodied Design, a methodology for creating cyborg function" [1]. The in this paper regarded simulation has its roots in articles which are the basis for these modern possibilities in prosthetics.

The regarded and with this paper provided simulation model exists of two legs, a pelvis, a torso and a head. This design is inspired by a walking human simulation of Hugh M. Herr from 2010 [3]. The body parts' spatial dimensions and inertias are calculated regarding statistics of summarized cadaver examination data in two dimensions [8]. With three Hill-Type muscles, a two dimensional leg motion is initiated at the implemented hip joint of the body. The muscle models are provided by a publication from 2014 [6]. Here, the design and function are explained in an official research paper. All other leg motions are either calculated dependent on the *Hip* movement or controlled by the joint's internal damping effects. They are adjusted to regulate and perform an observable, realistic looking jump.

2 Model Structure and Initial Properties

The simulation is implemented in the Simulink model *Jump_Simulation*, located inside the downloadable file *Human_model_jumping_hip_init*. The Matlab Version 2019b and a corresponding Simulink multibody library are needed to work with that file. The first overview is pictured in Fig. 1 and shows from left to right the subsystems *World Peripheral*, then *Pelvis* followed by the subsystems *Right Leg*, *Upper Body* and *Left Leg*. Inside the subsystem *World Peripheral*, a *Ground*

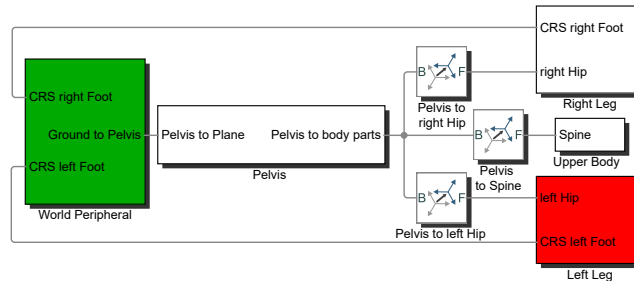


Fig. 1. Implemented model structure in Simulink

is implemented. It spans a solid area that generates a contact force with the soles of the feet to prevent the model from falling into nowhere. This is managed with an already existing contact force library which is included in the Simulink project [9]. The next subsystem is the *Pelvis* (see Fig. 1). Here, a prismatic joint with one degree of freedom (DOF) is implemented and can be exchanged with a two DOF (non-rotational) subsystem or with a six DOF joint. Solely the one and two DOF *Pelvis* were checked in order to simulate natural movement. The *Upper Body* is connected to the *Pelvis* with a revolute joint (one rotational DOF) starting at the middle top of the *Pelvis*. It consists of a torso and a head which are attached through another revolute joint. This subsystem is designed to be stiff and with any movement inhibited through the internal mechanics of the implemented revolute joints. They are only included to examine movement with an active *Upper Body* in the future. The body parts *Right Leg* and *Left Leg* are attached to the lower right or left edge of the *Pelvis*.

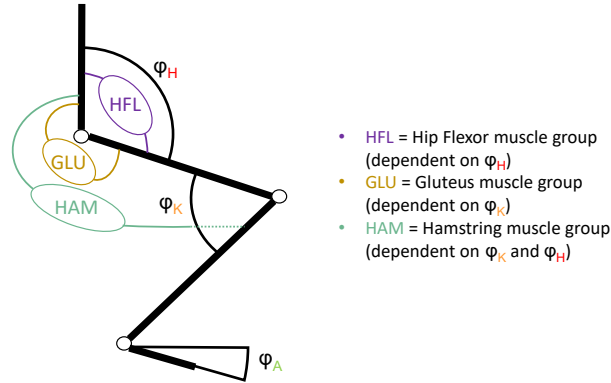


Fig. 2. Human model orientation and muscle locations (in style of [3])

Angle orientations and a reduced number of muscle groups are included on the bases of the article from 2010 (see Fig. 2) [3]. The jumping motion is supposed to be initiated by hip muscle force, thus only the muscle groups HFL, GLU and HAM are regarded. Other official researchers use this simulation built likewise to compare their results [3]. For future comparisons, this similarity was adopted. In addition, the total body length and mass are chosen in regard to the publication from 2010 [3]. The reason for that is to provide feasible first muscle parameters. Total body length and mass can be adjusted inside the Matlab code *LEG_PELVIS_TORSO_mech_parameter*. In regard to these, all specific body lengths, weights, inertias, and centres of mass are calculated by statistics in two dimensions [8]. There are other methods to calculate these initial body properties, which are more complex. To keep the model simple and prevent high computation power, a statistical approach was chosen.

2.1 Leg Structure

Both legs are designed equally. Exemplary, the *Left Leg* is pictured in Fig. 3. It starts with the DOFs of the *Hip* and ends with the *Foot* which is connected to the *World Peripheral* through the system's output *CRS* (coordinate system) *left Foot* (see Fig 2 and Fig. 3). This output is needed to span an area where the *Foot* generates contact force with the *Ground* [9]. The rest of the leg is aligned between the *Hip* DOFs and the *Foot*, beginning with the *Upper Leg*, followed by the *Knee*, the *Lower Leg* and the *Ankle*, coloured in gradation (see Fig. 3).

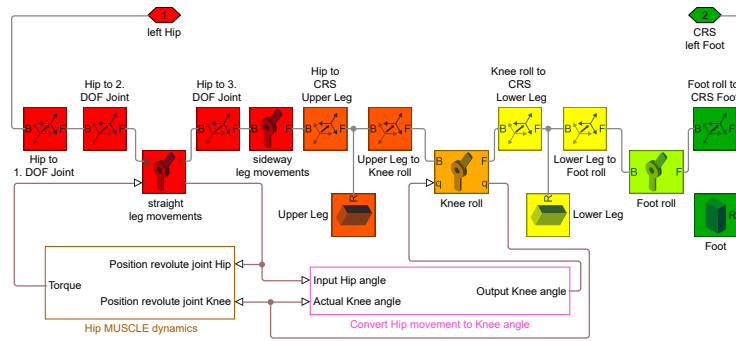


Fig. 3. Implemented leg structure in Simulink (*Left Leg*)

In Fig. 3 inside the brown rectangle all muscles and the required formulations for the muscle force and *Torque* calculation are implemented. The kinematic parameters for that are adopted from the publication of Hugh M. Herr and deposited in the Matlab file *Leg_Dynamics_para* [3]. Position information of the

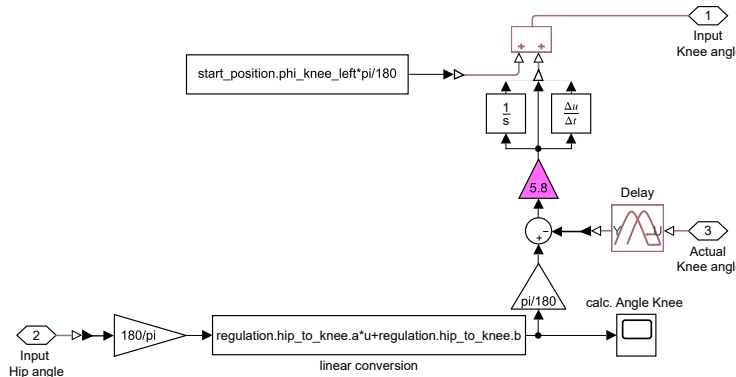


Fig. 4. Implemented *Knee* motion control in Simulink

Hip and the *Knee* joints are used to calculate new muscle length. Then a force is calculated and a *Torque* can be inserted at the joint for *leg sideways movement*. The muscle model usage is explained in the corresponding publication [6].

Inside the magenta rectangle of Fig. 3, the controller is located to convert *Hip* motion into a fitting *Knee* movement (see Fig. 4). Here, the *Hip* motion is first linearly mapped between an expected start and end angle of the *Knee*. Then the difference of this wanted *Knee* angle to the actual one is calculated. This displacement and its two derivatives are passed on to the *Knee* joint to generate the motion. The shown regulator was also used to try and convert the *Hip* motion into a fitting *Foot* movement. This could not be conducted because the *Ankle* needed a damping effect, which is now implemented through the internal properties of the used revolute joint.

2.2 Application and Adjustments

Adjustments concerning starting angles and initial muscle activity are recommended to be implemented into the Matlab code *Main_Init*. This code also initiates all the Matlab scripts necessary for the simulation. The angle of the *Ankle* is calculated here in accordance with the *Hip* and *Knee* angle to achieve an even sole to ground contact. In order to prevent the legs from exceeding the wanted motion, the range of the joint angles' agitation is fixed to a certain degree. These bounding angles are noted in Table 1 in regard to the orientation of Fig. 2. No further limitations are used to achieve boundaries for the leg movement. All initial angles should be chosen in between the boundaries given in Table 1.

Table 1. Internal boundaries of the leg joints (Symbols in respect to Fig. 2)

Symbol	Upper limiting angle [deg]	Lower limiting Angle [deg]	Description
φ_H	180	90	<i>Hip</i> angle
φ_K	-	-	<i>Knee</i> angle
φ_A	40 (start position)	-40	<i>Ankle</i> angle

If the total body mass or height needs to be changed, it is necessary to speed up or slow down the *Knee* motion. The easiest way to do so is to adjust the gain element of the *Knee* controller to a higher or lower number. It can be seen in magenta in Fig. 4. If a couple of simulations with the model have been executed, a user may change the contractile element optimum length and tendon initial length. These parameters can regulate the muscles further to the wanted motion.

2.3 Post Processing

All post processing data concerning the used muscle is automatically compiled inside the Matlab program *Plot_Data*. Here, the muscle force, length, and activation pattern of the *Right Leg* are exemplarily plotted in diagrams. This

is achieved by defining a data log name in the corresponding scope of the model. Data of the wanted simulated parameter is exported automatically to the Matlab workspace. The Matlab script starts on it's own after running the Simulation.

3 Model Review

Proper validation of the actual closeness to reality cannot be done. This is due to the lack of comparable official work examining a jumping motion. Also, no tests with a real human are executed during this work to derive realistic muscle force, length, and activation data. Nevertheless, some verifications observing phenomenological plausibility of the muscle length variation and applied muscle force during movement have been done. These are derived by comparing the Simulink visualization of the simulation with the muscle data plotted in diagrams during the post processing. Thus, the verification of the model is conducted by observing phenomenological plausibility regarding the following points:

- The muscles are not extending or shrinking unrealistically
- Force variation of the muscle groups is plausible
- The visualized motion looks natural

These points were examined and accomplished. For compartments with realistic data in the future, it may be necessary to adjust the parameters of *HFL*, *GLU* and *HAM*. This is because the used muscle settings are derived for walking and not for jumping [3]. Still, the goal simulating a jumping motion is reached.

4 Conclusion and Outlook

The closeness to reality is not determined during the tests of the simulation. This is due to the lack of other comparable official work and proper empirical data with a corresponding human body. But phenomenological correctness of the implemented structure is accomplished. The simulation of jump locomotion is possible in two directions. Correct muscle connection and a natural-looking movement were achieved. The result of this work is the successful implementation of a phenomenologically accurate biomechanical simulation.

During the work, further ideas could be derived to extend the implemented model. They concern the properties of the skeleton and the muscle to obtain a simulation that is adjustable to different body types. Locomotion studies can lead to a better understanding of how to build future prosthetics [3]. Thus, these modifications are noted to develop the presented simulation and enhance the possibility to derive feasible data of a real jump.

To broaden the skeleton properties different joint locations should be adjustable in Matlab code. To improve the foot regulation, its shapes can be adjusted and another segmenting of the foot can lead to better regulation and motion control [4]. But these modifications make the model more complex and an evaluation on the necessary effects should be executed. Also, the implementation of another rotational degree of freedom inside the two DOF Pelvis is needed

to give the body model the possibility to turn and fall during the simulation. Then the rotation induced by the leg motion needs to be damped or regulated. Otherwise, the Upper Body would rotate during the jump. This can be managed by a regulative system for movement of the Upper Body that stabilizes the gyration.

To accomplish adjustable muscle properties, the calculation of the initial muscle length and its maximum isometric force could potentially be achieved by statistics. These parameters should be implemented dependent on the total body height and weight. Also, the adopted kinematic formulations for muscle force calculation should be dependent on the length of the single body segments. At last, a periodic square wave signal can replace the current linear ramp used for muscle stimulation. Properly adjusted, this leads to periodical locomotion which allows checking the stability of the simulated jump [5].

References

1. How we'll become cyborgs and extend human potential. Online (April 2018), https://www.ted.com/talks/hugh_herr_how_we_ll_become_cyborgs_and_extend_human_potential
2. Clites, T.R., Carty, M.J., Ullauri, J.B., Carney, M.E., Mooney, L.M., Duval, J.F., Srinivasan, S.S., Herr, H.M.: Proprioception from a neurally controlled lower-extremity prosthesis. *Science Translational Medicine* **10**(443) (2018)
3. Geyer, H., Herr, H.M.: A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE Transactions on neural systems and rehabilitation engineering* **18**(3), 263–273 (2010)
4. Günther, M., Ruder, H.: Synthesis of two-dimensional human walking: a test of the λ -model. *Biological Cybernetics* **89**(2), 89–106 (2003)
5. Haeufle, D., Grimmer, S., Seyfarth, A.: The role of intrinsic muscle properties for stable hopping—stability is achieved by the force–velocity relation. *Bioinspiration & Biomimetics* **5**(1), 016004 (2010)
6. Haeufle, D., Günther, M., Bayer, A., Schmitt, S.: Hill-type muscle model with serial damping and eccentric force–velocity relation. *Journal of Biomechanics* **47**(6), 1531–1536 (2014)
7. Herr, H.M., Clites, T., Maimon, B., Zorzos, A., Carty, J.M., Duval, J.F.: Method and system for providing proprioceptive feedback and functionality mitigating limb pathology. WO/2017/120484 (July 2017), <https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2017120484>
8. Krishnan, R.H., Devanandh, V., Brahma, A.K., Pugazhenth, S.: Estimation of mass moment of inertia of human body, when bending forward, for the design of a self-transfer robotic facility. *Journal of Engineering Science and Technology* **11**(5), 166–176 (2016)
9. Miller, S.: Simscape multibody contact forces library. GitHub (September 2021), <https://github.com/mathworks/Simscape-Multibody-Contact-Forces-Library/releases/tag/21.1.5.0>
10. Srinivasan, S.S., Gutierrez-Arango, S., Teng, A.C.E., Israel, E., Song, H., Bailey, Z.K., Carty, M.J., Freed, L.E., Herr, H.M.: Neural interfacing architecture enables enhanced motor control and residual limb functionality postamputation. *Proceedings of the National Academy of Sciences* **118**(9) (2021)