XVII International Symposium on Computer Simulation in Biomechanics
July 28th – 30th 2019, Alberta, Canada

A THREE-DIMENSIONAL MUSCULOSKELETAL MODEL GENERATING HUMAN GAIT PATTERNS: A LIGHTWEIGHT CUSTOMIZABLE OPEN-SOURCE SYSTEM

Makoto Yoshida and Kazunori Hase

Department of Mechanical Systems Engineering, Tokyo Metropolitan University, Japan
Email: yoshida-makoto@tmu.ac.jp

INTRODUCTION

Walking is a fundamental everyday movement of humans, whose bipedal nature distinguishes human from any other extant mammals. Computer simulation of human gait is useful as a platform of virtual experiments, where even dangerous situations such as those including falling are tolerated without ethical consideration. It may also be used to simulate the process of human gait evolution, which is impossible to reproduce experimentally due to the enormous time required.

Hase and Yamazaki developed a forward-dynamics simulation system that generates three-dimensional human gait patterns [1]. This system does not require any information about actual human movement, which are only available through time-consuming measurements, and is able to generate stable gait patterns for as long as three gait cycles or longer. This system was subsequently used to study the causal relationship between gait patterns and the physical malfunctions of elderly people [2].

In spite of the usefulness and the success of the system, it had only been used by a limited number of researchers, who have had personal contact with the authors. We are currently refining the source code of the system in a preparation to provide this system as open-source software.

METHODS

This system will be distributed as a set of C++ source codes, which may not be friendly to those who are not familiar with programming languages, but offers maximal flexibility; it can be customized to do anything a programming language can do, even beyond the imagination of the authors of the system. The platform of the system has migrated from C with X Window System to C++ with Qt, which increased the number of potential users because Qt is a cross-platform application framework.

The model reproduces the whole body of a human as a rigid-link system with 14 links. Each degree of freedom of a joint is modeled as a hinge with viscoelastic properties. The torso is divided into four parts: the pelvis, the lower part of the lumber, the upper part of the lumber and the thorax, which is combined with the neck and the head. Each joint on the torso has a single degree of freedom, that for flexion and extension, except for that between the pelvis and the lower part of the lumber, which has three degrees of freedom. Each leg is divided into three parts: the thigh, the calf and the foot. Each hip joint has three degrees of freedom. Each knee has one degree of freedom. Each ankle has two degrees of freedom; one for eversion and inversion, and the other for plantar flexion and dorsiflexion. Each arm is divided into two parts: the upper arm and the forearm, which is combined with the hand. Each shoulder has two degrees of freedom; one for abduction and adduction, and the other for flexion and extension. Each elbow has one degree of freedom. Class SkeletalSystem handles the dynamics of the rigid-link system, including the coordinate conversions.

This skeletal system is equipped with 70 muscle models. The geometry of each muscle, used to calculate the length and the moment arm, is modeled as a series of line segments, the vertices of which are defined as fixed points

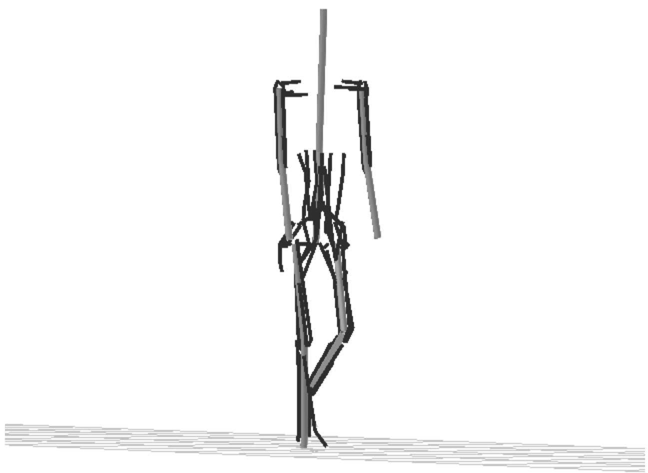

Fig 1: Three-dimensional representation of the musculoskeletal model. Gray cylinders represent the rigid links. Black cylinders represent the geometry of the muscle models.

with respect to the nearest body segment. Class MuscularSystem handles the dynamics of the muscular system.

This musculoskeletal system (Fig. 1) is controlled by a model of nervous system constructed according to that of Taga [3]. The nervous system has a set of 23 pairs of neural oscillators as the rhythm generation system, which emulates the central pattern generator. Each pair of neural oscillators is assigned to a joint and controls the muscles relating to the joint, hence the name. The muscles are stimulated proportionally to the output from the pair of neural oscillators, while the stimuli are distributed among the muscles by static optimization so as to equalize the load. Class NervousSystem handles the dynamics of the set of nervous system. The neural oscillators are modulated by the sensory feedback system simulating the spinal reflex, employing the somatic senses, e.g. the joint angles, the ground reaction forces and the positions and the angles of the body segments with respect to the global coordinate system.

The dynamics of the whole model, including the skeletal, the muscular and the nervous systems, is expressed as simultaneous differential equations, which are numerically solved with the time step of the integration of 0.4 ms. Most part of this model is described in a human-readable data file in the JSON format, and can be modified manually as required. Time-series data of the generated movements, i.e. that of the joint angles, muscular tensions and the ground reaction forces, are saved as a CSV file. The system is designed adequate to reproduce human gait patterns, while being lightweight for the sake of the computation speed.

The parameters of the nervous system are optimized by genetic algorithm (GA) with the criteria of maximizing the locomotive energy efficiency and the smoothness of the muscular tension change. The process of GA includes mutation, crossover and selection. The initial set of state variables of a successful individual is inherited to the descendant.

The migration to one of the object-oriented programming languages enables the intuitive description of the overall workflow of the system as follows. An instance of class Individual represents a whole-body human model. This class has an instance of class SkeletalSystem, that of class MuscularSystem and that of class NervousSystem. An instance of class Individual is provided as an argument to the constructor of class Solver, which solves the differential equations to generate the gait pattern. An instance of class TimeSeries, which is to record the movement data, is also provided to the constructor of class Solver. The generated movement is displayed three-dimensionally by an instance of class View3D, using OpenGL technology. After the generation of the gait pattern, its performance is evaluated and recorded to an instance of class Fitness, which is then coupled with an instance of class Chromosome that has a copy of the mutable parameters of the instance of class Individual. This couple will then undergo the process of GA.

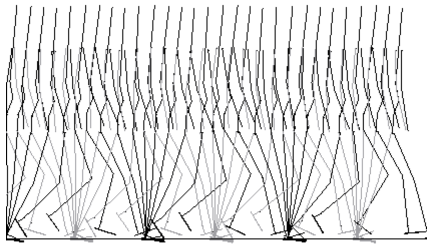

Fig 2: Sequence of a human gait pattern generated by the system. The postures of every 0.1 s are drawn. Three cycles of gait has successfully completed.

RESULTS AND DISCUSSION

The stride, the speed and the period of a gait cycle of an optimized gait pattern were 1.135 m, 1.107 m/s and 1.026 s, respectively, which are comparable to an actual human gait (Fig. 2). More than 10000 individuals were generated and evaluated in the GA for the optimization.

As this system is designed to be lightweight, this system completed a simulation of three gait cycles, which takes 2.809 s, in a 30 s of computation time, when run on a tablet PC (Microsoft Surface Pro 4, Intel Core i7-6650U 2.20 GHz, 16.0 GB, Microsoft Windows 10 Pro). When run on a multi-core hardware, this system is capable of generating gait patterns of multiple individuals parallelly during GA, which accelerates the optimization by approximately a factor of the core number.

CONCLUSIONS

A forward-dynamics simulation system that generates three-dimensional human gait patterns has been ported to C++ with Qt. A draft version of this source code will be provided to the audience via e-mail upon request to the corresponding author.

REFERENCES

1. Hase K and Yamazaki N, *JSME Int J Ser C* **45**: 1040-1050, 2002.
2. Hase K, *Anthropol Sci* **116**: 95-104, 2008.
3. Taga G, *Biol Cybern* **73**: 97-111, 1995.