Real time modeling of human body dynamics

Ilya Phenichniy[*], Andrey Morozov[*], Victor Sholukha[*,%]
[*]Department of Applied Mathematics, State Polytechnic University of Saint Petersburg, Russia.
[%]Communicating author: vcholouk@ulb.ac.be

***Abstract*—3D real time human body dynamics modeling allows fast and correct assessment of internal forces distribution, providing measured by motion analysis kinematics. In this paper we touch upon a subject of unsupported human body motion modeling with real-time or interactive rate. Speed is very important when someone examines the reaction of human body on some external influence or change of configuration of system containing human body. This paper discusses two approaches for real-time modeling of body dynamics and presents their programming realization.**

Keywords-human body; dynamics; real-time

I. INTRODUCTION

Despite of being relatively easy type of motion for modeling unsupported motion is nevertheless very important field of study in terms of practical application in biomechanics. The samples of such motion include movements of astronauts in a state of weightlessness or movements of athletes during jumps in various kinds of sports. The purpose of the present work was creating a program that allows simulating an anthropomorphic model and controlling its behavior. Controlling behavior means to apply internal forces and torques to body, change its configuration. The importance of such a simulator for practice and behavior modeling is obvious.

This paper describes two approaches to human dynamics modeling.

First is based on Featherstone's [4] Articulated Body Algorithm (ABA). It is well known among robotics society as fast (linear time complexity) and accurate. This approach uses reduced coordinates formulation and is suitable for fixed set of joints (constraints).

The second uses Baraff [2] and Faure [3] algorithm based on Lagrange multipliers approach. It is suitable for general type of constraints even nonstationary (e.g. direct 3D kinematics measurement). It works in linear time for tree-like structures (e.g. human body model).

An attempt to combine both approaches in one so-called engine makes it flexible enough to cover wide range of application, including unsupported motion.

II. REQUIREMENTS

Undoubtedly a physical accuracy is among the main features of physically based modeling. But often the price for good accuracy is performance lose. Unfortunately, most of existing algorithms in robotics and mechanical engineering are oriented towards physical accuracy, but not performance. That conflicts with real-time algorithms demand, which offer an ability to tune the configuration and observe a feedback immediately.

A. Biomechanical conformity

Biomechanical conformity of obtained results is based on two states. The first one is using real anthropometric input data (linear and mass-inertial), interpolated with special regression formulas and parameters. Correct numerical integration accuracy is the second one.

The difference in anthropometric data caused by gender and race is well known. Storage and reuse of all combinations of gender, race, age, weight, height and other parameters is impossible. Thus, an equation for solving this problem was introduced in [8], which allow calculating mass-inertial and linear characteristics of wide range of human bodies, based on the list of input parameters such as age, gender, race, weight and height. Using body-profile method, introduced in [9], a set of regression parameters was founded for approximation of anthropometrical values for a specific character.

B. Performance

Real-time program execution requirement cause a limitation on time complexity of algorithm. Best of presently known algorithms have $O(n)$ time complexity, n is a number of acyclic constraints. Human skeleton model has a tree-like topology (i.e. no cycles), so we will not consider algorithms working with kinematics cycles.

It's also desirable that the method developed is numerically stable to allow the use of simple integration method.

III. ALGORITHMS

Constrained body consists of a number of rigid bodies – *links* – connected by *joints*. For N free (not constrained) rigid bodies there are $6N$ degrees of freedom (DOFs). In our case we have joints (connections), so common number of DOFs will be $M < 6N$.

State of articulated body at any moment can be defined with a set of parameters (e.g. joints angles and velocities). This set is named *state vector*.

There are two main principles of simulation of constrained rigid body systems. The first one is named *Lagrange multipliers method*. This method has become popular due to its simplicity and intuitive intelligibility, as it is a direct extension of well-known rigid body techniques. In this method constrained system of N rigid bodies is considered to be a system of N independent bodies with correspondent equations of motion. Connections between bodies (joints) are considered to be separate constraints, belonging to a common set of system constraints. Dimension of state vector, used in this method is *6N.* Most detailed description of this method can be found in Baraff's works [2].

Second method – *Articulated body algorithm (ABA)* - considers constraints in explicit form, so they are parts of equations of motion for system. Thus it only uses *M* state variables in state vector. To use this method, all joints configurations must be known, so that motion equations can be written. In our case, when we investigate anthropomorphic systems, we can consider all connections as joints with 1, 2 or 3 free DOFs. This method is based on principles of dynamic programming and first was introduced by Vereshchagin [10] but has only become widely popular after works of Featherstone [4].

For both methods realizations with *O(N)* (where *N* is a number of bodies in system) time complexity exists, but multiplicative constant for *ABA* method is smaller, than corresponding one for *Lagrange multipliers* method, because of smaller number of arithmetic operation needed. Furthermore *ABA*, in distinct from *Lagrange multipliers* method, performs calculations in local joints coordinates, not in Cartesian coordinates, that improves calculation accuracy of algorithm. As shown in [1] *ABA* is more physically accurate which is very important for our work.

To make developed engine more flexible and universal both approaches were implemented. Another reason was necessity of fast comparison both approaches for accuracy evaluation (e.g. constraints stability). An accuracy of both approaches was also compared with SimMechanics results modeling for a set of tests.

A. Articulated Body Algorithm (ABA)

Articulated body algorithm is based on dynamic programming principle – each next step uses data obtained on a previous step. Main concepts of *ABA* algorithm are articulated body – sub-tree of initial set of bodies torn off link with index *i* and *i*th link itself L_i named a handle. Main result of Featherstone's work [4] is the theorem stating that the following equation is correct for any articulated body in system:

$$f_i^I = I_i^A a_i + p_i^A \qquad (1)$$

where f_i^I - spatial force applied by inboard joint to L_i link; I_i^A - spatial articulated inertia of L_i link; a_i - acceleration of L_i link, p_i^A - spatial articulated zero-acceleration force of L_i link. For I_i^A and p_i^A corresponding recursive formulas are presented in [4].

Using these formulas we may calculate acceleration of any L_i link at any moment. Then by numeric integration we obtain full state vector of constrained rigid body system.

B. Baraff- Faure algorithm

As mentioned above this approach is based on Lagrange multipliers and consider articulated body as system of not connected bodies, which are influenced by constrained forces in addition to external. Baraff [2] uses widely known sparse formulation of Newton's 2nd law with constrained forces:

$$\begin{pmatrix} M & -J^T \\ -J & 0 \end{pmatrix} \begin{pmatrix} \delta \dot{v} \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ -b \end{pmatrix}.$$

It allows finding constraints induced acceleration correction $\delta \dot{v}$ in linear time.

Faure suggested dividing joints in two sets: acyclic and cyclic. For the 1st set Baraff's technique is used and 2nd is solved in time $O(al + l^2)$ using iterative algorithm, where a and l are amount of acyclic and cyclic constraints sets correspondingly.

The quadratic addition to complexity will not appear in unsupported human body motion until model remains tree-like structured.

IV. CONSTRAINTS MODELING

For solving problem of using joints constraints and also give end-user an ability to control and change system configuration in the context of using Featherstone algorithm we will use so called *test forces* method. This method uses discrete structure of numerical integrator - values of state vector from previous step are used to calculate new values on next simulation step.

Using this discrete structure and also *acceleration constraint* statement – limits for value of acceleration of a given joint – a method was introduced in [5] and [6] that allows to calculate needed value of torque(force) for setting articulated body in desired state. This method is based on theorem stating that there is a linear relationship between the magnitude of a generalized force applied to the body and the magnitude of the acceleration of the body's joints or links.

Formal notation is as follows:

$$a^f = kf + a^0 \qquad (2)$$

where a^f - observed acceleration magnitude after the force is applied; a^0 - observed acceleration magnitude before the force is applied; k - scalar constant; f - magnitude of the force applied to the body.

Furthermore, the following generalized statement is true: for any linear functions of joint acceleration $h(\ddot{q})$ and for some constant *k* the following equation is true:

$$h(\ddot{q}^f) - h(\ddot{q}^0) = kf \qquad (3)$$

Using this formula we can implement the so-called *test force* method: first, we calculate acceleration magnitude without applying any additional forces. Next we apply some known test force to a joint and calculate new acceleration magnitude in a joint. Then, using formula *(3)*, we can calculate the value of coefficient *k*. After that, knowing the value of *k* and using linear relationship between magnitude of applied force and magnitude of resulting acceleration, we can calculate the needed value of torque (force) to give the joint the desired acceleration.

For calculating acceleration after test force applying both Featherstone's and any other algorithms allow calculation of acceleration, like it's done in [6]. Thus, common time complexity of algorithm for calculation new state vector with one acceleration constraint is $O(2*N) = O(N)$.

This method can be generalized for the case of *m acceleration constraints.* In this case we need to solve a system of linear equations of k_i^j - proportionality factor between force applied to *i*-th joint and acceleration appearing in *j*-th joint. Using *ABA* algorithm we can obtain all k_i^j factors performing *(m+1)* calls to the procedure. Taking into account the complexity of algorithm for finding solution of system of linear equations - $O(m^3)$ - total algorithm complexity is $O(mn+m^3)$.

V. RESULTS

Main result of this work is developing a program for simulation of unsupported anthropomorphic motion in real-time. Developed program allows not only simulation of free motion of model with given starting condition, but also motion control, setting desired parameters values (angles and angular velocities) of specified joints.

Performance of developed algorithms is illustrated by the following diagram (Fig. 1). It shows functional dependence between one step simulation time and number of simultaneously presented human models (39 DOFs per model). As the diagram shows even a model with 390 DOFs allows simulation in real-time.

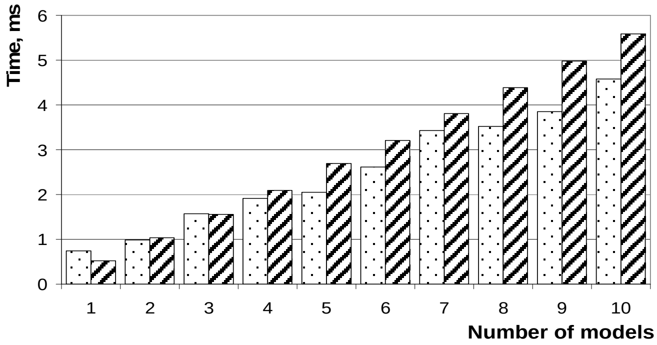

Figure 1. Performance of developed algorithms.

░ - ABA, ▨ - Baraff-Faure

This test was carried out on P4 3.06 HT, with 2 GB RAM.

According to figure 1 both methods show identical linear-time performance as derived theoretically. In figures 2-6 are presented several frames of stick animation of 3D controllable human body motion in real time. User has an ability to control each not blocked joint's DOF (coordinate, velocity or acceleration) during simulation.

VI. DISCUSSION AND FUTURE WORK

The main purpose of this work was implementation of combined approach for developing flexible algorithm and program, which allows physically accurate and realistic simulation of anthropomorphic system behavior. Approximate code capacity is ~600 Kb (~17000 lines). Capability of configuration (state) of this system control – i.e. capability of behavior control assumes user friendly interface and visual feedback. From mechanical point of view the offered algorithm should solve both forward and inverse (mixed) dynamics problem. In case of multipliers approach the 3D kinematics implementation (e.g. motion analysis data) looks more natural, but for predefined total body topology Featherstone's approach less time consuming.

Using Featherstone's ABA algorithm as forward dynamic algorithm with linear time complexity allowed us to simulate model behavior for models with large number DOFs (up to 400 DOFs) in real-time. Anthropomorphic model from [9] with 39 DOFs were chosen as test model.

Using *test force* method in our work allowed us to realize joint constraints, which exist in any real anthropomorphic system.

Since end-users are interesting in not only free motion of model with given starting condition, but to a greater extent with model behavior under motion control, the given algorithm should allow to perform this control. Desired configuration is a state vector with specific values of angles and angular joint velocities. This desired configuration can be obtained using acceleration constraints for joint acceleration. Thus, test force method allows us to solve the second problem – motion control – as well.

An important part of presented program is implementation of regression equations for human body 3D mass-inertia characteristic evaluation. More over, using bilateral constraints in case of motionless joint and substituting these joints by welding like constraint we are reducing total DOFs to make simulation less time consuming.

Further we will add some contact model to our simulator to enlarge number of possible simulated situations. Today we see large variety of contact modeling approaches: from impulse based [7] to linear complementary problem (LCP) or penalty.

REFERENCES

[1] U. M. Asher, D. K. Pai and B. P. Cloutier. Forward Dynamics, Elimination Methods and Formulation Stiffness in Robot Simulation. Int. J. of Rob. Research, vol.16, no.6, pp. 749-758, 1997.

[2] D. Baraff. Linear-time dynamics using Lagrange multipliers. In *Computer Graphics (Proc. SIGGRAPH '96)*, Computer Graphics Proceeding, Annual Conference Series, pages 137-146. ACM SIGGRAPH, Addison Wesley, August 1996. ISBN 0-201-94800-1.

[3] F. Faure. Fast refinable equation solution for articulated solid dynamics. IEEE Transactions on visualisation and Computer Graphics, Volume 5, Number 3, page 268-276. July 1999.

[4] R. Featherstone. The calculation of robot dynamics using articulated-body inertias. International J. of Robotics, 1983, 2(1), 13-29.

[5] E. Kokkeyis, D. Metaxas. Efficient dynamics constraint for animated articulated figures. Multibidy System Dynamics, 1998, 2, 89-114.

[6] E. Kokkeyis. Practical physics for articulated Characters. Game Developers Conference, 2004.

[7] B. Mirtich. Impulse based dynamics simulation of rigid body systems – PhD Thesis, 1996.

[8] G. B. Shan, K. Nicol. Method for obtaining anthropometrical data. Proceedings of the 15th Congress of the International Society of Biomechanics.

[9] G. B. Shan, C. Bohn. Antropometrical data and coefficients of regression related of gender and race. Applied Ergonomics, 34 (2003), 327-337. A.F.

[10] Vereshchagin. Computer simulation of the dynamics of complicated mechansisms of robot manipulators. *Engineering Cybernetics*, 6:65–70, 1974.

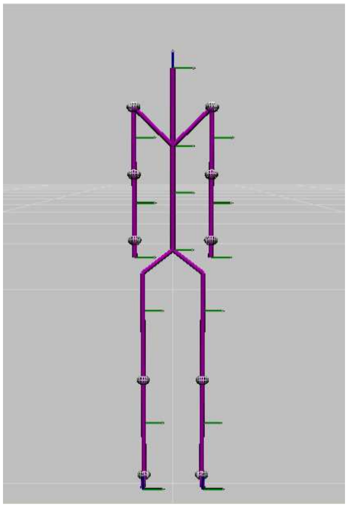

Figure 2.

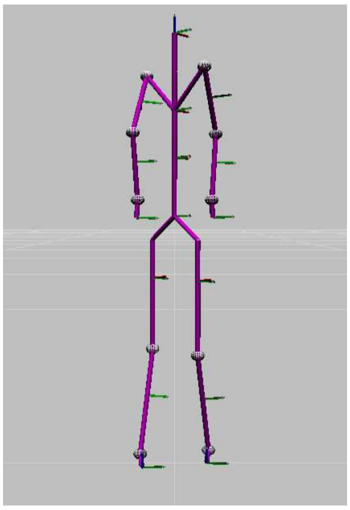

Figure 3.

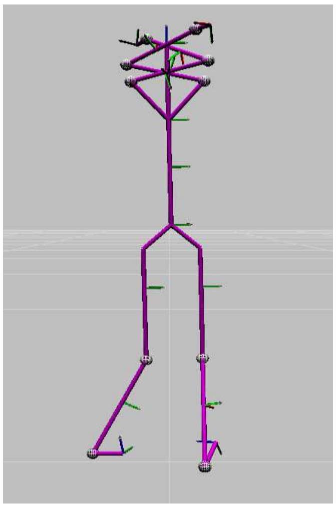

Figure 4.

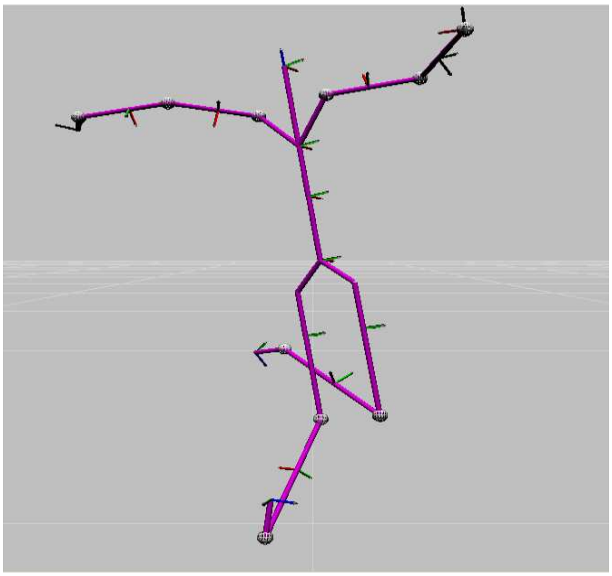

Figure 5.

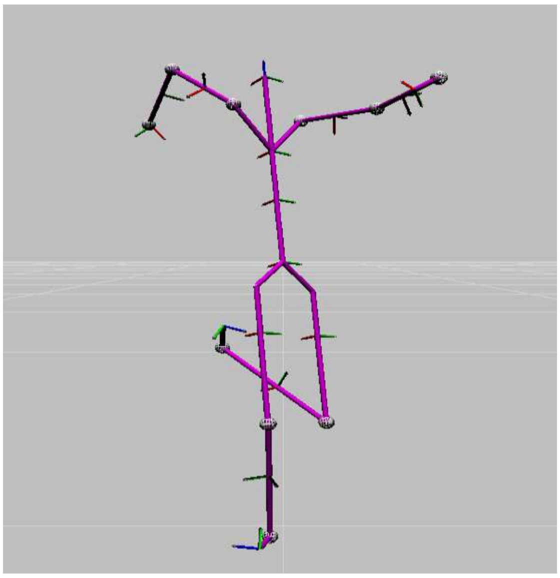

Figure 6.