Dynamic Simulator of Articulated Virtual Avatars

Jaume Gaseni, Antonio Susín
Dept. Matemàtica Aplicada I
Universitat Politècnica de Catalunya
Diagonal 647, 080828-Barcelona

Abstract

A dynamic simulator based on the dynamic robotic equations applied to articulated structures is presented. The correct trajectories are obtained using a motion capture equipment (exported in biovision format). Motion is reproduced from these trajectories with a mixed kinematic and dynamic model. Position and orientation of each link of the virtual skeleton are described using Denavit-Hartenberg coordinates. The inverse and direct dynamic problems are solved with Newton-Euler and generalized D'Alembert formulations. Warping technique is also implemented in order to obtain more complex movements from simple captured ones.

Key words: dynamic simulator, humanoid virtual, warping, articulated structures.

1 Introdution

A dynamic simulator of an articulated structure can be an useful tool for different application fields. On an animation context, it can be used as a generator of motion for a virtual humanoid [12], [19]. It can be used also in the design of actual industrial robots because one can obtain the forces and torques acting in the joints of the structure [7],[13]. This information can be used also in biomedicine to compute the stress at the joints of the legs and arms of a patient [6],[1].

The complexity of the movements of a human being forces to work with trajectories obtained using capture motion equipment and the subsequent reproduction from a dynamic point of view. On carrying out a dynamic simulation of articulated chains, motion is produced as a result of a series of applied forces, among them: Gravity, the pair produced by a motor (at the joints),as well as inertias, Coriolis and friction forces. The assembly of all these forces describes the physical behavior of the motion of the articulated chain in the real space.

Our simulator can be used to animate scenes where there are articulated rigid solids, such as biped skeletons or robotics arms, from trajectories previously captured. Moreover, dynamic information can be obtained and used in case of a deeper study is needed.

2 Mathematical models

In order to make a dynamic simulation of a articulated chain as the one considered in this paper, we need to solve two different dynamic problems, the **inverse** and the **direct** problem. The last one is known as **simulation**. The pairs which are the forces obtained as solutions of the inverse problem, are used to obtain the final motion resulting from the direct problem.

The dynamics of an articulated chain is solved using the mathematical formulations of the equations of the motion of a robotic arm. These are an assembly of second order differentials equations that describe the dynamic behavior. In order to place and to orient each element of the skeleton in the 3D space the representation of Denavit-Hartenberg [9] is used.

For the inverse problem we use the improved Newton-Euler (N-E)formulation ([9]) since it is more efficient computationally speaking. This formulation consists of an assembly of recursive equations divided in two iterations. One ahead iteration, from the first element to the last one of the chain, where the kinematics information is propagated, and one backwards iteration, where forces and pairs for each of the joints of the articulated chain are computed.

The N-E recursive equations are presented below differentiating these two iterations:

Ahead equations : $i = 1, 2..., n$

$$\mathbf{R_0^i}\omega_\mathbf{i} = \mathbf{R_{i-1}^i}(\omega_\mathbf{i-1} + \mathbf{Z_0}\dot{q}_i)$$

$$\mathbf{R_0^i}\dot{\omega}_\mathbf{i} = \mathbf{R_{i-1}^i}(\mathbf{R_0^{i-1}}\dot{\omega}_\mathbf{i-1} + \mathbf{Z_0}\ddot{q}_i +$$

$$+\mathbf{R_0^{i-1}}\omega_\mathbf{i-1} \wedge (\mathbf{Z_0}\dot{q}_i))$$

$$\mathbf{R_0^i}\dot{\mathbf{v}}_\mathbf{i} = \mathbf{R_{i-1}^i}\dot{\omega}_\mathbf{i} \wedge \mathbf{R_0^i}\mathbf{p_i^*} +$$

$$+\mathbf{R_0^i}\omega_\mathbf{i} \wedge (\mathbf{R_0^i}\omega_\mathbf{i} \wedge \mathbf{R_0^i}\mathbf{p_i^*}) + \mathbf{R_{i-1}^i}\mathbf{R_0^{i-1}}\dot{\mathbf{v}}_\mathbf{i-1}$$

$$\mathbf{R_0^i}\mathbf{a_i} = \mathbf{R_0^i}\dot{\omega}_\mathbf{i} \wedge \mathbf{R_0^i}\mathbf{s_i} +$$

$$+\mathbf{R_0^i}\omega_\mathbf{i} \wedge (\mathbf{R_0^i}\omega_\mathbf{i} \wedge \mathbf{R_0^i}\mathbf{s_i}) + \mathbf{R_0^i}\dot{\mathbf{v}}_\mathbf{i}$$

being $\mathbf{R_{i-1}^i}$ the Euler's rotation matrix, ω_i and $\dot{\omega}_i$ are the angular velocity and acceleration respectively, $\dot{\mathbf{v}}_\mathbf{i}$ is the lineal acceleration and $\mathbf{a_i}$ is the lineal acceleration of the center of mass of each link. Variables $\dot{q}_i$ and $\ddot{q}_i$ are velocity and acceleration of the angular variable for each link.

Backwards equations: $i = n, n-1, ..., 1$

$$\mathbf{R_0^i}\mathbf{f_i} = \mathbf{R_{i+1}^i}\mathbf{R_0^{i+1}}\mathbf{f_{i+1}} + m_i\mathbf{R_0^i}\mathbf{a_i}$$

$$\mathbf{n_i} = \mathbf{R_{i+1}^i}(\mathbf{R_0^{i+1}}\mathbf{n_{i+1}} +$$

$$+\mathbf{R_0^{i+1}}\mathbf{p_i^*} \wedge \mathbf{R_0^{i+1}}\mathbf{f_{i+1}}) +$$

$$+(\mathbf{R_0^i}\mathbf{p_i^*} + \mathbf{R_0^i}\mathbf{s_i}) \wedge \mathbf{R_0^i}\mathbf{F_i} + \mathbf{R_0^i}\mathbf{N_i}$$

$$\tau_i = (\mathbf{R_0^i}\mathbf{n_i})^T\mathbf{R_{i-1}^i}\mathbf{Z_{i-1}} + b_i\dot{q}_i$$

being $\mathbf{f_i}$ the force applied over the joint i, $\mathbf{n_i}$ is the moment and τ_i is the associated pair.

The N-E formulation was initially stated to move articulated chains like a robotic arm or a manipulator. In a robotic context it is considered that each single joint has only one degree of freedom. Ball-and-socket joints are not used because of their lack of precision and they are also difficult to control. In order to apply this formulation to a human skeleton which moves in the 3D space fictitious elements have been added [9], that have no mass and null length but maintaining the appropriate rotation axis. For each actual element there will be two fictitious ones with their respective joints. The combination of the three elements will produce an association of three joints with their three different rotation axes allowing a full 3D motion.

On the other hand, to solve the direct dynamic problem the generalized D'Alembert's formulation has been used [9] since it is a closed formulation and a very well structured one. This will facilitate the numerical integration of the differential equations using an explicit method like the usual Runge-Kutta 4. The system of differential equations can be written as

$$\sum_{j=1}^{n} \mathbf{D_{ij}}\ddot{\theta}_j(t) + \mathbf{h_i^{tras}}(\theta, \dot{\theta}) + \mathbf{h_i^{rot}}(\theta, \dot{\theta}) + \mathbf{c_i} = \tau_i \quad (1)$$

for $i = 1, 2, ..., n$

In the above equation, first term is related with the inertia of the elements. Second and third terms depend on the centripetal and Coriolis forces caused by the translation and rotation of the elements. The $\mathbf{c_i}$ term reflects the applied external forces. In our case, the only external force considered is the usual gravity. Finally, term τ_i is the pair at each joint, this term has been previously calculated as a solution of the inverse problem. The results of the numerical integration of these equations are the dynamic trajectories that will be represent on the graphic window of our simulator.

3 Dynamic control

Given the equations of the dynamic motion of an articulated chain, the objective of the control procedure is to maintain the dynamic simulation in agreement with some pre-specified criterium. Although the problem of the control can be stated so easily, its solution becomes complicated when different forces act simultaneously.

Taking into account the characteristics of the articulated chains, their trajectories and their formulations we decided to implement a simple derivative and proportional control, using the technique of the calculated pair. This law of control can be obtained from the recursive equations of N-E replacing $\ddot{q}_i(t)$ to obtain the necessary pair for each joint. The final expression of the control equation can be written as

$$\ddot{q}_i(t) = \ddot{q}_i^d(t) + \sum_{j=1}^{n} \mathbf{K_v^{ij}}[\dot{q}_j^d(t) - \dot{q}_j(t)] + \sum_{j=1}^{n} \mathbf{K_p^{ij}}[q_j^d(t) - q_j(t)] \quad (2)$$

where $\mathbf{K_p^{ij}}$ and $\mathbf{K_v^{ij}}$ are the gains of feedback at joint i for the position and derivative, and $e_i(t) = q_i^d(t) - q_i(t)$ is the position error for this joint. The physical meaning of the previous equation in N-E recursive equations can be understood in the following way:

1. First term will generate the pair that would correspond to each joint if there is no modelation error and if the physical parameters are known for the system. However, errors due to roominess at the joints, lack of precision for some parameters, etc, will give inevitable deviations of the desired trajectory for the joint.

2. The remaining terms in the equation (2) will generate the correction pair to compensate small deviations on the trajectory.

Matrices of feedback gain $\mathbf{K_p}$ and $\mathbf{K_v}$ must be diagonal dominant and positive defined in order to obtain a critically damped system by each one of the subsystem of the joints. With this control the error is not eliminated completely but we can reduce it obtaining very satisfactory results in the final simulations.

4 The problem of trajectories

The problem of defining the trajectories corresponding to a human motion is a complex task with very difficult solution. In order to solve the inverse problem, we needed to know the trajectories, which will be simulated later. To find trajectories for each one of the joints of a skeleton is a problem that cannot be solved by test and error. Our simulator uses as trajectories the values stored in a file, using biovision standard format, which is created from a motion capture equipment *MotionStar* (an Ascent trademark) owned by *Televisió de Catalunya*. (see figure 1).

The values obtained by this equipment when recording a motion are very noisy and as a consequence sudden changes in the acceleration direction of the joints of the skeleton are caused. Therefore, negative effects in the dynamic behavior of the simulation arises and non desired oscillations appear. In order to filter and to smooth the captured values a method based on Kernel

Figure 1: Capturing several motions

functions, has been implemented. We make an average of the values according to a density function associated to each point. If we consider point data as pairs $(t_i, x_i)_i$ where i is the number of captured frames, we define a Kernel function for each point by.

$$f_i(t) = \frac{1}{\sqrt{2\pi}} \exp\left[\frac{-1}{2}\left(\frac{t - t_j}{p}\right)\right]^2,$$

where p is a variable parameter used to weighting the neighboring points and that can be modified according to the target point. The filtered data $(t_i, \bar{x}_i)$ are computed from the weighted function

$$\bar{x}_i = \frac{1}{p} \sum_{j=0}^{nframes} x_j f_j(t_i).$$

This way we are able to filter the trajectories and to improve the behavior of the simulation.

Another feature of the present simulator is the possibility of warping different captured motion. Two independent movements can be connected in a continuous way without any discontinuity. As a result, more complex animations are obtained from the previous ones. This can be done applying the technique of **Warping** between two movements. A warping zone of transition is generated making an interpolation between both movements. The new motion can be obtained as

$$\theta_r(t) = \theta_a(t)(1 - \lambda) + \theta_b(t)\lambda,$$

where θ_r is the resulting movement, θ_a is the first movement, θ_b is the second movement and λ is a parameter that indicates the transition speed of the first movement towards the second.

5 Description of the simulator

The simulator is mainly structured in four modules with different relations between them.

1. **Input/Output Module :** This module manages the input and output files of the different data required by the simulator.
2. **Module of Trajectories:** It is the module in charge of generating all the trajectories. It also filters the captured trajectories and applies warping when the user decides.
3. **Module of the Simulator:** It is the nucleus of the application. This module contains all the dynamics for the animation (inverse and direct problem) and is the one that allows to move the skeletons and solids of the scene.
4. **Graphical Module :** It is the module which contains the computer graphics options of the simulator and manages the user's interface needed to represent and to interact with the simulations.

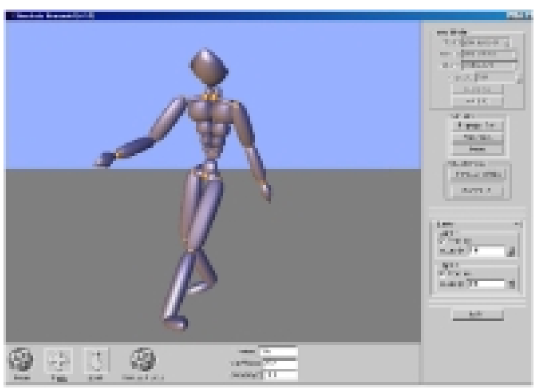

Figure 2: An humanoid frame

The above images are an example of a frame of a simulation of a humanoid which is walking in the first case and making an acrobatic dance in the second. In all the animations the same geometric primitive is used to construct each link of the virtual humanoid. Every link of our humanoid is obtained from the so called *superquadrics* [14], consisting on a quadratic parametric surface that depends on few parameters. Many different shapes can be obtained from the variation of these parameters.

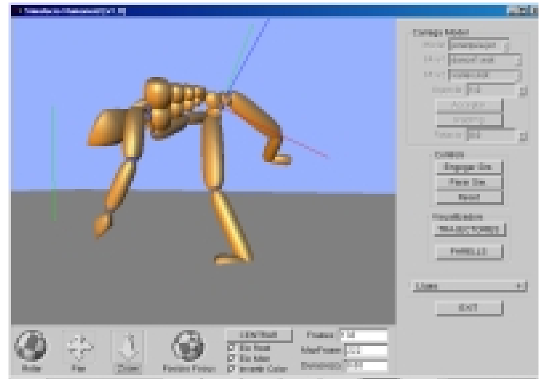

Figure 3: Another simulation view

6 Additional options of the simulator

Our simulator has two additional options that allow the user to visualize both the curve of trajectories and the curve of the pairs for each joint of the skeleton at every moment of time. The information about the maximum and minimum pair until the current moment or the pair in current frame is also facilitated. From the analysis and deep study of these curves very interesting applications and conclusions in the field of robotics and in the field of medicine can be reached.

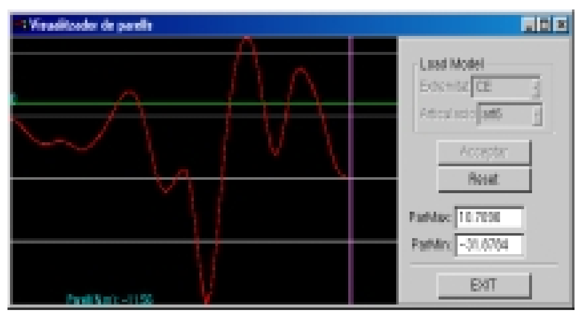

Figure 4: Curve of pair of humanoid's knee

As a possibility, these curves could help to design a robotic arm. Knowing the amount of work, the length and the weight of the arm, the pair curves can be obtained. Therefore, a theoretical computation of the intensity of the electrical motors necessary to produce the prefixed movement can be made. From the relation

$$\tau = K_m * I \tag{3}$$

where τ is the pair calculated using the simulator, K_m an empirical constant that depends on the characteristics of the motor and manufacturer, and finally I is the current intensity. Thus, knowing the necessary current rank to produce the motion one can be able to determine the necessary theoretical power of the motor. The variation on length and weight of the robotic arm will produce a change in its dynamics and, therefore, modifications in the pairs and the power (that we will be able to recalculate with the simulator) will take place.

On the other hand, in a medicine context it can be useful to compare the efforts to which the joints of the body are put under when making a certain movement. The greater the pair, the greater the effort produced by the joint and therefore it will suffer more damage. From the capture of the motion of a patient and its weight, the corresponding curves of pair could be obtained, and thus the ideal weight of this patient can be stated in order to constrain their curves of pair within standards limits considered save for the patient.

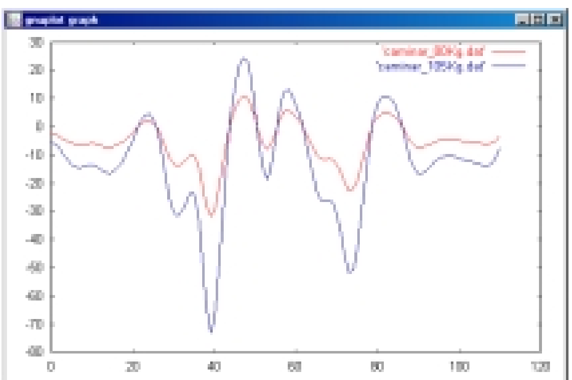

Figure 5: Comparison, with different weight, of the effort of a knee on walking

References

[1] Arnaldi B., Nougaret J.L., Hegron G, Ravazi A., WP6, Human Arm Motion Control Charm Project Report.*1:111 (1995).*

[2] Badler N.I., Virtual Humans for Animation, Ergonomics, and Simulation. *Virtual Humans: Behaviors and Physics, Acting and Reacting*, pp. 1-42. Course notes of ACM SIGGRAPH'97.

[3] Boulic R. and Thalmann D., Combined Direct and Inverse Kinematic Control for Articulated Figure Motion Editing. *Computer Graphics Forum, 11(4):189-202, 1992.*

[4] Boulic R., Fua P., Herda L., Silaghi M., Monzani J., Nedel L. and Thalmann D., An Anatomic Human Body for Motion Capture. *EMMSEC, Bordeaux, September 1998.*

[5] Bruderlin A. and Calvert T.W., Goal-Direct, Dynamic Animation of Human Walking. *In Proceedings of ACM SIGGRAPH'89, pp. 233-242. Addison Wesley, July 1989.*

[6] Faure F., Debunne G., Gascuel MP. and Moulton F., Dynamic Analysis of Human Walk. *In Eurographics Workshop on Computer Animation and Simulation, pp. 95-107. Set. 1997.*

[7] Fu K.S., Gonzalez R.C., Lee C.S., ROBOTICA: Control, Detección, Visión e Inteligencia. *Mc Graw Hill, 1990.*

[8] Fua P., Gruen A., Plänkers R., D'Apuzzo N. and Thalman D., Human Body Modeling And Motion Analysis From Video Sequences. *International Archives of Photogrammetry and Remote sensing,Japan.*

[9] Gaseni J., Susín A., Simulador dinámico de cuerpos articulados. Humanoides Virtuales. *PFC Ingenieria Industrial (ETSEIB), Marzo 2001.*

[10] Herda L., Fua P., Plänkers R., Boulic R. and Thalman D., Skeleton-Based Motion Capture Reconstruction of Human Motion. *Computer Graphics Lab (LIG).*

[11] Hodgins J.K., Three-Dimensional Human Running. *In Proccedings of the IEEE Conference on Robotics and Automation, avril 1996. Minneapolis, Minnesota.*

[12] Ko H. and Badler N.I., Animating Human in Realtime Using Inverse Dynamics, Balance and Comfort Control. *IEEE Computer Graphics 16(2), pp. 50-59, 1996.*

[13] Lilly K.W., Efficient Dynamic Simulation of Robotic Mechanisms. *Kluwer Academic Publishers, 1993.*

[14] Linares P., Rodriguez L. and Montilla G., Genetic Algorithm Fitting of Deformable Superquadrics Applied to Left Ventricle Visualization.

[15] Mas R., Cinética Inversa de segundo orden para animación de estructuras articuladas. *Editores Joan R., Navazo I., Quirós R., CEIG-2000 Col. Treballs d'Informàtica i Tecnologia 3, 283:298 (2000).*

[16] Multon F. Contrôle du Mouvement des Humanoïdes de Synthèse PhD. *Thèse. Univ. de Rennes (1998).*

[17] Phillips C.B. and Badler N.I., Interactive Behaviours for Bipedal Articulated Figures. *Computer Graphics, Volume 25, Number 4, July 1991.*

[18] Wilhelms J., Dynamics for Computer Graphics: A tutorial.*Computing Systems, Vol 1 (1), 63:93 (1988).*

[19] Zordan, V. B., Hodgins, J. K. 1999. Tracking and Modifying Upper-body Human Motion Data with Dynamic Simulation. Computer Animation and Simulation '99, Eurographics Animation Workshop, pp. 13-22, Sept. 1999.