The Visual Computer

Interactive control of biomechanical animation

T. Ertl, H. Ruder, R. Allrutz,
K. Gruber, M. Günther,
F. Hospach, M. Ruder, J. Subke,
and K. Widmayer

Lehr- und Forschungsbereich
Theoretische Astrophysik, Universität Tübingen,
Auf der Morgenstelle 10, D-72076 Tübingen,
Germany

Physics-based animation can be generated by performing a complete dynamical simulation of multibody systems. This leads to the solving of a complex system of differential equations in which biomechanical results for the physics of impacts are incorporated. Motion control is achieved by interactively modifying the internal torques. Realtime response requires the distribution of the workload of the computation between a high-speed compute server and the graphics workstation by means of a remote-procedure call mechanism.

Key words: Animation – Biomechanics – Distributed Computing

Correspondence to: T. Ertl

1 Introduction

In the first phase of computer animation the traditional techniques of animation were brought to the computer. This resulted in computer-animated films where the keyframes were linked by image-based and parametric interpolation. It soon became obvious, especially when trying to compute aesthetic human movement, that a more realistic computer animation had to take into account the basic physical properties of the objects and the fundamental principles of physics that govern their movement (Wilhelms and Barsky 1985; Armstrong and Green 1985; Zeltzer 1982; Cohen 1989). In algorithmic animation, the evolution of the state of a system of objects is not determined by interpolation, but by physical laws given either as algebraic formulas in the simple case or as a more complicated set of coupled nonlinear differential equations. The most general approach for generating physically correct animation sequences is to perform a complete dynamical simulation of the given model that takes into account all external and internal forces and torques. However, even rigid objects with only a few degrees of freedom or very simple elastic models require supercomputer performance, and the design of a desired motion by controlling the internal torques is a tedious, highly interactive process.

2 The motion of a rigid body

Although there are several computer animation techniques involving interacting point-masses, most of the objects of interest are rigid bodies. A rigid body is defined by its total mass M and its tensor of inertia $\underline{\underline{\Theta}}$. Its position and orientation in space are determined by three Cartesian coordinates for the position of the center of mass $\vec{r}_c$ and by three Euler angles α, β, γ for the orientation of a body-fixed coordinate system $\xi\eta\zeta$, the origin of which coincides with the center of mass (Fig. 1). The equations of motion decouple into the Newton equation for the motion of the center of mass under the influence of external forces:

$$M\ddot{\boldsymbol{r}}_c=\dot{\boldsymbol{P}}=\sum_i \boldsymbol{F}_i \tag{1}$$

and Euler equations for angular velocities of the rotation relative to the center of mass described by the body-fixed axes under the influence of external torques:

$$\dot{\boldsymbol{L}}=\frac{d}{dt}(\underline{\underline{\Theta}}\,\omega)=\sum_i(\boldsymbol{r}_i'\times\boldsymbol{F}_i)+\sum_i \boldsymbol{T}_i \tag{2}$$

The Visual Computer (1993) 9:459–465

Here, $\boldsymbol{P}$ denotes the total momentum of the rigid body; $\boldsymbol{F}_i$, the external forces; $\boldsymbol{L}$, the angular momentum; and $\boldsymbol{T}_i$, the external torques relative to the center of mass. If the torques are known, Eq. (2) can be integrated so that they yield the components ω_ξ, ω_η, ω_ζ of the angular velocity projected on the body-fixed axes as functions of time. A further integration of the first-order differential equation system leads to the motion of the body-fixed system connecting the components ω_ξ, ω_η, ω_ζ with the Eulerian angles α, β, γ and their time derivatives:

$$\dot{\alpha} = -\frac{1}{\sin\beta}(\omega_\xi \cos\gamma + \omega_\eta \sin\gamma) \tag{3a}$$

$$\dot{\beta} = \omega_\xi \sin\gamma + \omega_\eta \cos\gamma \tag{3b}$$

$$\dot{\gamma} = \omega_\xi \cot\beta \cos\gamma - \omega_\eta \cot\beta \sin\gamma + \omega_\zeta . \tag{3c}$$

These equations can easily be derived and integrated to give the physically correct motion for a given rigid body. For the case of a freely falling rod (mass M, length l, center of mass at x_c, y_c, z_c, no rotation around its main axis, which means $\omega_\zeta = 0$) the only external force is gravity (M_g), which acts on the center of mass, and the external torques vanish:

$$M\ddot{x}_c = 0, \quad M\ddot{y}_c = 0, \quad M\ddot{z}_c = -Mg \tag{4a}$$

$$\dot{\omega}_\xi = 0, \quad \dot{\omega}_\eta = 0. \tag{4b}$$

If we further assume that the rod rotates in the same plane in which its center of mass moves ($\alpha = 0$), and if we introduce the angle $\varphi = 90° - \beta$ for the orientation of the rod, the equations of motion can be solved analytically:

$$x_c = v_{cx0}\, t + x_{c0} \tag{5a}$$

$$z_c = -\tfrac{1}{2} g t^2 + v_{cz0}\, t + z_{c0} \tag{5b}$$

$$\varphi = \omega_y\, t + \varphi_0 . \tag{5c}$$

Nevertheless, even for the simple case of a falling rod, the correct treatment of the impact of the rod hitting the ground is a nontrivial problem. Since there is only a very limited number of interesting animations involving noninteracting rigid bodies, it is worthwhile to investigate the physical aspects of the impact. Assuming the rod and the ground to be infinitely rigid, the duration of the impact is infinitely short and the ground reaction force $\boldsymbol{F}_G$ is a δ-peak.

The integration of the equations of motion

$$M\ddot{x}_c = F_{Gx}(t) = c_x\, \delta(t^{\mathrm{impact}}) \tag{6a}$$

$$M\ddot{z}_c = -Mg + F_{Gz}(t) = -Mg + c_z\, \delta(t^{\mathrm{impact}}) \tag{6b}$$

$$\Theta\ddot{\varphi} = (x_e^{\mathrm{impact}} - x_c)\, F_{Gz} - (z_e^{\mathrm{impact}} - z_c)\, F_{Gx} \tag{6c}$$

over the infinitesimally short impact time leads to the relations between the changes of the linear and angular momenta and the integral of the impact forces and torques

$$M(\dot{x}_c^{\mathrm{after}} - \dot{x}_c^{\mathrm{before}}) = M\Delta\dot{x}_c = \int_{t^{\mathrm{impact}}-\varepsilon}^{t^{\mathrm{impact}}+\varepsilon} F_{Gx}(t')\, dt' = c_x \tag{7a}$$

$$M(\dot{z}_c^{\mathrm{after}} - \dot{z}_c^{\mathrm{before}}) = M\Delta\dot{z}_c = \int_{t^{\mathrm{impact}}-\varepsilon}^{t^{\mathrm{impact}}+\varepsilon} [-Mg + F_{Gz}(t')]\, dt' = c_z \tag{7b}$$

$$\Theta(\dot{\varphi}^{\mathrm{after}} - \dot{\varphi}^{\mathrm{before}}) = \Theta\Delta\dot{\varphi} = (x_e^{\mathrm{impact}} - x_c)\, c_z - (z_e^{\mathrm{impact}} - z_c)\, c_x . \tag{7c}$$

More information on the mechanical properties of the ground is necessary to calculate these changes explicitly. Two limiting cases are on the one hand, a totally elastic behaviour and, on the other, a totally inelastic behaviour. In the first case the sign of the z-component of the velocity of the impacting end $\dot{z}_e$ of the rod is altered. The second equation for determining the constants c_x and c_z describes the conservation of the total energy during the impact. In the second case the end of the rod comes to rest immediately after the impact, which yields the two additional equations desired. The reaction force of the ground, which depends on the amount of deformation and on the instantaneous velocity, has to be taken into account for a more realistic model. As an example of a specific ground behaviour, the following equations can be used as components of the ground reaction force:

$$F_{Gz} = a(z_{\mathrm{deform}})^b + d\,\dot{z}_{\mathrm{deform}}; \qquad F_{Gx} = \mu F_{Gz} . \tag{8}$$

The vertical component of the ground reaction force depends on the deformation and on the deformation velocity of the ground; a, b and d are material constants. The horizontal component of the ground reaction force is usually determined by friction and therefore proportional to F_{Gz}. The initial effect on the free parameters of a ground reaction-force ansatz can be visualized by interactively modifying them during the animation (Fig. 2).

3 The mechanics of multilinked models

Multilinked systems of extended bodies connected by joints are necessary for the modelling of human beings or animals with legs and arms. Developing a satisfactory model is by no means a trivial problem. The joints and their constraints, as well as the mechanical properties of the body segments, must be correctly described. As an example of this approach we discuss the plane two-linked rigid model (Fig. 3). Here, we have eight unknowns, the six coordinates x_{c1}, z_{c1}, φ_1, x_{c2}, z_{c2}, φ_2 and the two constraint forces F_{12x}, F_{12z}. The equations of motion can be derived easily:

$$m_1 \ddot{x}_{c1} = F_{Gx} + F_{12x} \tag{9a}$$

$$m_1 \ddot{z}_{c1} = -m_1 g + F_{Gz} + F_{12z} \tag{9b}$$

$$\Theta_1 \ddot{\varphi}_1 = (\boldsymbol{r}_{11} \times \boldsymbol{F}_G)_y + (\boldsymbol{r}_{12} \times \boldsymbol{F}_{12})_y + T_{12} \tag{9c}$$

$$m_2 \ddot{x}_{c2} = -F_{12x} \tag{10a}$$

$$m_2 \ddot{z}_{c2} = -m_2 g - F_{12z} \tag{10b}$$

$$\Theta_2 \ddot{\varphi}_2 = (\boldsymbol{r}_{21} \times \boldsymbol{F}_{12})_y - T_{12}. \tag{10c}$$

In addition, we have the joint conditions, which require that the upper end of the lower segment is connected to the lower end of the upper segment:

$$x_{c1} + \frac{l_1}{2} \cos \varphi_1 = x_{c2} - \frac{l_2}{2} \cos \varphi_2 \tag{11a}$$

$$z_{c1} + \frac{l_1}{2} \sin \varphi_1 = z_{c2} - \frac{l_2}{2} \sin \varphi_2. \tag{11b}$$

Now we can introduce four independent coordinates x_h, z_h, φ_1, φ_2 and eliminate the two unknown constraint forces and we get the minimum set of differential equations for four independent coordinates:

$$\begin{aligned} &(m_1+m_2)\,\ddot{x}_h - (m_1+2m_2)\frac{l_1}{2} \sin \varphi_1\, \ddot{\varphi}_1 \\ &\quad - m_2 \frac{l_2}{2} \sin \varphi_2\, \ddot{\varphi}_2 \\ &= (m_1+2m_2)\frac{l_1}{2} \cos \varphi_1\, \dot{\varphi}_1^2 + m_2 \frac{l_2}{2} \cos \varphi_2\, \dot{\varphi}_2^2 + F_{Gx} \end{aligned} \tag{12a}$$

$$\begin{aligned} &m \ddot{z}_h + (m_1+2m_2)\frac{l_1}{2} \cos \varphi_1\, \ddot{\varphi}_1 + m_2 \frac{l_2}{2} \cos \varphi_2\, \ddot{\varphi}_2 \\ &= (m_1+2m_2)\frac{l_2}{2} \sin \varphi_1\, \dot{\varphi}_1^2 + m_2 \frac{l_2}{2} \sin \varphi_2\, \dot{\varphi}_2^2 \\ &\quad + F_{Gz} - mg \end{aligned} \tag{12b}$$

$$-\Theta_1 \ddot{\varphi}_1 = T_G - \frac{l_1}{2} \sin \varphi_1\, F_{Gx} + \frac{l_1}{2} \cos \varphi_1\, F_{Gz} + T_{12} \tag{13a}$$

$$-\Theta_2 \ddot{\varphi}_2 = -l_2 \sin \varphi_2\, F_{Gx} + l_2 \cos \varphi_2\, F_{Gz} - T_{12} \tag{13b}$$

However, serious problems occur when the corresponding equations resulting from larger models are manipulated in the same way by a symbolic-algebra program. Huge algebraic terms are produced when the symbols representing the forces and torques of constraint are eliminated and the segmental coordinates are replaced by generalized coordinates. The size of these terms exceeds the capabilities of symbol manipulation programs, though such programs are being given an increasing number of degrees of freedom. Also, the generated source code may reach dimensions that are very hard to handle. These problems can be avoided by using a method that requires the solution of a system of more equations, but whose terms are of a much simpler structure. If we use this approach for the plane two-linked rigid model, we first have to differentiate the constraint conditions twice:

$$\begin{aligned} &\ddot{x}_{c1} - \frac{l_1}{2}(\cos \varphi_1\, \dot{\varphi}_1^2 + \sin \varphi_1\, \ddot{\varphi}_1) \\ &\quad = \ddot{x}_{c2} - \frac{l_2}{2}(\cos \varphi_2\, \dot{\varphi}_2^2 + \sin \varphi_2\, \ddot{\varphi}_2) \end{aligned} \tag{14a}$$

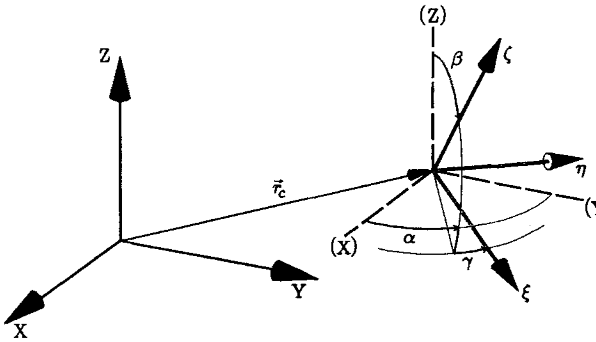
$$\begin{aligned} &\ddot{z}_{c1} + \frac{l_1}{2}(-\sin \varphi_1\, \dot{\varphi}_1^2 + \cos \varphi_1\, \ddot{\varphi}_1) \\ &\quad = \ddot{x}_{c2} + \frac{l_2}{2}(-\sin \varphi_2\, \dot{\varphi}_2^2 + \cos \varphi_2\, \ddot{\varphi}_2). \end{aligned} \tag{14b}$$

Together with Eqs. 9 and 10, we now have eight linear equations for six second derivatives of the coordinates $\ddot{x}_{c1}$, $\ddot{z}_{c1}$, $\ddot{\varphi}_1$, $\ddot{x}_{c2}$, $\ddot{z}_{c2}$, $\ddot{\varphi}_2$ and two constraint forces F_{12x} and F_{12z} that show a much simpler structure than the minimum number Eqs. 12 and 13. Even in three dimensions and for an arbitrary number of segments the structure of the equations of motion can be generally written in the form

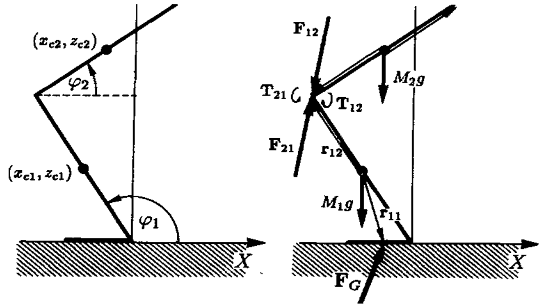
$$\underline{M}(\boldsymbol{x}) \cdot \ddot{\boldsymbol{x}} + \boldsymbol{C}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \boldsymbol{F}(\boldsymbol{x}, \dot{\boldsymbol{x}}) \tag{15}$$

where the vector of independent coordinates is denoted by $\boldsymbol{x}$; the mass matrix by $\underline{M}$: the vector of the generalized Coriolis and centrifugal forces, by $\boldsymbol{C}$; and the vector of the generalized forces and torques acting on each segment, by $\boldsymbol{F}$.

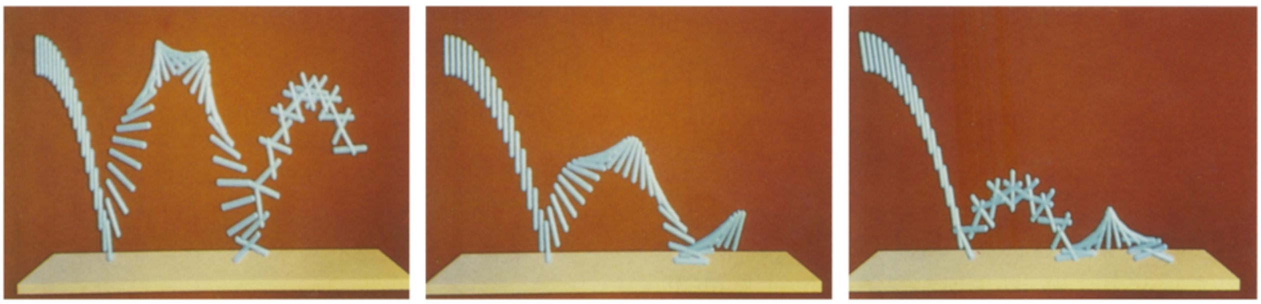
After each of the second order equations has been split into two first-order equations by introducing a new independent variable, the resulting initial-value problem for a system of coupled ordinary differential equations can be integrated numerically. A variable-order, variable-stepsize method should be chosen for a good trade-off between accuracy and efficiency. We successfully used a



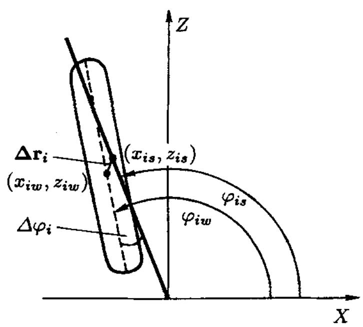
1



3



2



4

Fig. 1. Coordinates for the determination of position and orientation of a rigid body in space. The Euler angle β is the angle between the (Z)- and ζ-axis, the Euler angle α is defined as the angle between the line of intersection of the $(Z)\,\zeta$-plane with the $(X)(Y)$-plane and the (X)-axis, and the Euler angle γ is the angle between the line of intersection of the $(Z)\,\zeta$-plane with the $\xi\eta$-plane and the ξ-axis

Fig. 2. Three stroboscopic time series of a falling rod for an increasing (left to right) damping component in the ground reaction force

Fig. 3. The plane two-linked rigid model

Fig. 4. Plane model of one body segment consisting of a skeletal part and a coupled wobbling mass, which is movable relative to the skeletal part. Displacement of the centers of mass ΔT; relative rotation $\Delta\varphi_i$

code called DE (Shampine and Gordon 1975), which is based on an Adams-Bashford predictor-corrector method. A routine from Linpack or any other matrix solver can be used to invert the mass matrix for each evaluation of the right-hand side. However, the limits of the model based on a multi-linked system consisting of only rigid body segments soon become obvious when one tries to simulate motion with high accelerations. The reasons for this shortcoming are easily recognized by analysing a high-speed movie of a jump or of an impact. The reaction of a segment of the human body, trunk, thigh, lower leg, or arm is very unlike that of a rigid body during an impact. Therefore, it is necessary to take into account the different composition of the body, namely the rigid skeletal part and the soft components like tendons, muscles, organs, and, last but not least, fat.

A correct finite element modelling of the human body, is extremely expensive, and an animation based on this level is nearly hopeless. A simple, practicable and very successful method of modelling the essential properties is to introduce a *wobbling mass* (Gruber et al. 1987, 1991), which summarizes all the soft parts of a segment. The wobbling mass is coupled quasi-elastically to the skeletal part and is strongly damped to it. This wobbling mass can be moved and rotated relative to the skeleton, as shown in Fig. 4. The additional coordinates needed to describe the wobbling mass for each segment in the plane case are: (1) two Cartesian coordinates Δx_i, Δz_i for the displacement of

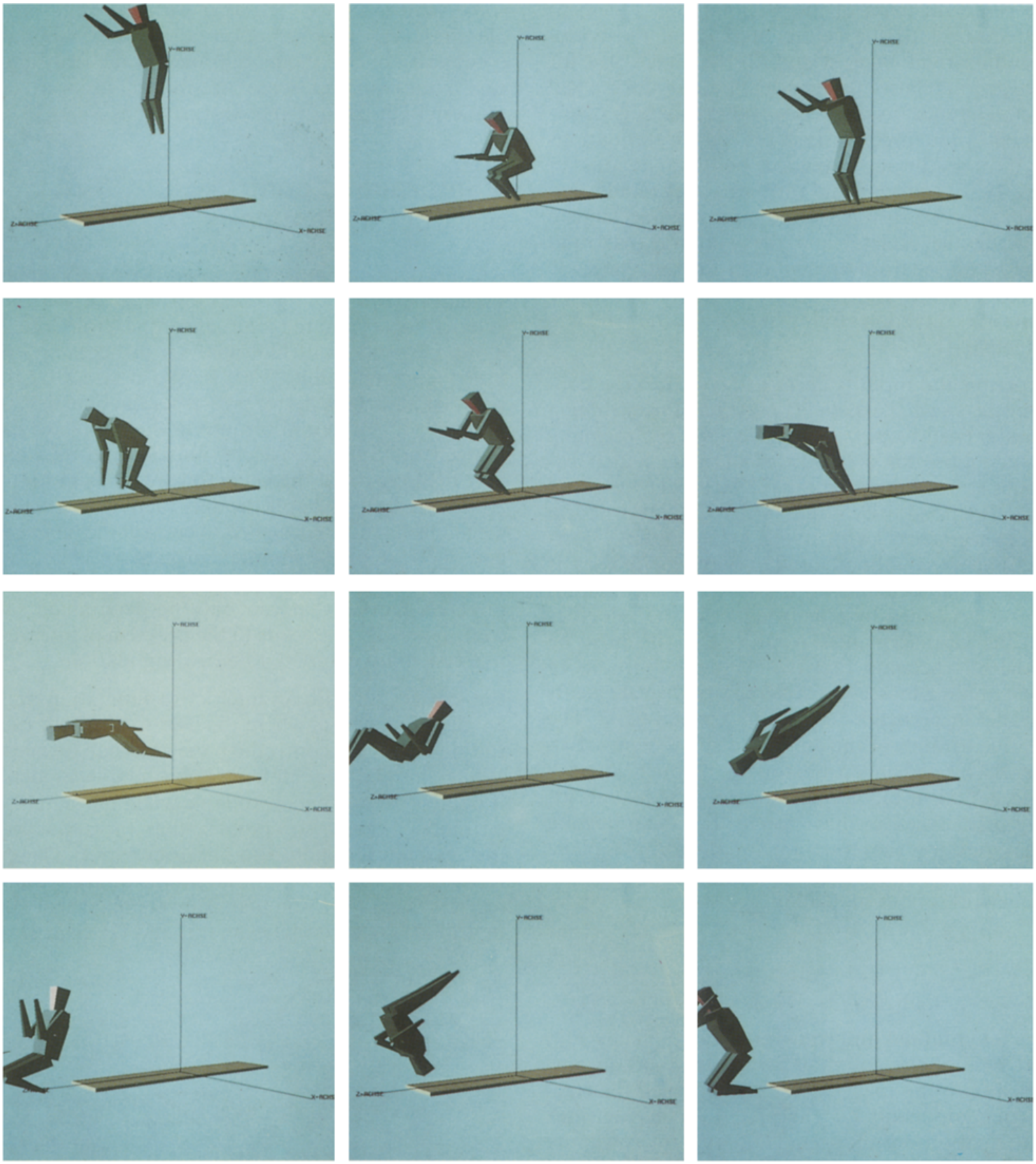


Fig. 5. Snapshots of a down jump of a five-linked model with landing on a stiff plank. The internal torques are controlled in such a way that the model jumps off again and performs a somersault

the center of mass of the wobbling-mass element with respect to the centre of mass of the corresponding rigid element and (2) the angles $\Delta\varphi_i$ for the torsion relative to the orientation of the skeletal part. The motion of each wobbling mass is determined by the three (or six in the 3 D-case) equations of motion for an extended body analogous to Eqs. 1–3. The forces and torques acting in addition to gravity are given by the coupling mechanism between the skeletal and the wobbling part and depend on the displacement coordinates. The same forces and torques act on the skeletal part in the opposite direction following the *actio = reactio* principle.

Experimental input is necessary to adjust the coupling between the skeletal part and the wobbling mass as well as possible. In general, the coupling is very loose for a small displacement and it becomes stiff in a narrow range. Such behaviour can be described by a term in the form "(displacement)^m" with an exponent $m \approx 3 \ldots 4$. Furthermore, the motion of the wobbling masses is strongly damped and stops after a few oscillations. This is described by a dependence of the velocities on the displacements. Additionally, it must be taken into account that the coupling constants of the wobbling masses are different for displacements parallel or perpendicular to the skeletal parts. The relative torques of the angular displacements $\Delta\varphi_i$, can be treated in an analogous manner. After a long period of biomechanical experiments and fitting procedures, we have found that the following equations for the coupling forces (for the plane case) seem to be the best approximations in the framework of our modelling.

$$T_{wi} = a_{wi}\,\Delta\varphi_i + b_{wi}\,\Delta\dot{\varphi}_i \tag{16a}$$

$$F_{wi,l} = c_{wi,l}\,\mathrm{sign}(\Delta r_{il})|\Delta r_{il}|^3 + d_{wi,l}\,\Delta\dot{r}_{il} \tag{16b}$$

$$F_{wi,t} = c_{wi,t}\,\mathrm{sign}(\Delta r_{it})|\Delta r_{it}|^3 + d_{wi,t}\,\Delta\dot{r}_{it}. \tag{16c}$$

The longitudinal and transversal components $F_{wi,l}$ and $F_{wi,t}$ of the coupling forces are then decomposed onto the space-fixed axes and yield the Cartesian components $F_{wi,x}$ and $F_{wi,z}$, which enter into the equations of motion. The coupling constants a_{wi}, b_{wi}, $c_{wi,l}$, $d_{wi,l}$, $c_{wi,t}$ and $d_{wi,t}$ must be adjusted individually.

Even for the simplest case of the simulation of human motion, the plane three-linked wobbling-mass model results in a system of 14 coupled second-order differential equations. In Fig. 5 a down jump of an extended five-linked model performing a somersault is shown. Together with the quite simple graphical representation, the computational complexity of this model approaches the real-time capabilities (in the sense of smooth motion) of modern high-performance workstations.

4 Interactive control of direct dynamics

An approach still quite often used to do a dynamic simulation of a complex motion is to derive the external forces and the internal torques from a real motion by means of film analysis or acceleration measurements (Selbie 1989; Boulic et al. 1990). In contrast, direct dynamics prescribes the time development of the internal torques in the joints, which are generated by the skeletal muscles and thus reflect the free will of the being to control its motion. Although while sophisticated active feedback mechanisms can be designed to control the increase and decrease of the internal torques, for example, during a down jump (Ruder et al. 1991), many parameters remain that can only be determined by trial and error. This leads to the problem of interactive control of compute-intensive simulations.

We started out with an interactive simulation and animation program where the user was able to use the dial box for controlling the internal torques. While this worked quite well for simple models, it became almost uncontrollable when the simulated model became so complex that there was no instantaneous response to the user input. Therefore, we tried to partition the problem into (1) numerical integration and matrix inversion, which are performed by a compute server, and (2) visualiza-

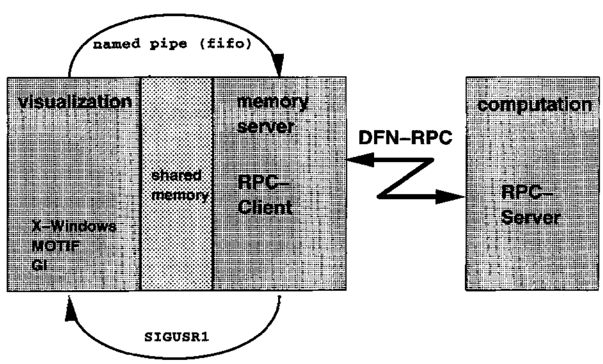


Fig. 6. Remote-procedure call setup with a special shared memory server, which allows for user interaction even while the compute server is performing the next simulation step

tion of the motion, which is performed on the graphics workstation. We used a remote-procedure call mechanism developed for the DFN (German Research Network), which offers special support for FORTRAN programmers and great flexibility in the underlying synchronization mechanisms (synchronous, asynchronous, buffered, unbuffered) (Rabenseifner 1992). In order to allow for user interaction at the workstation even during the remote-procedure call being executed on the compute server, a special shared-memory server process has been designed that is controlled from the graphics application via named pipes and signals (Allrutz and Rabenseifner 1992). Thus, not only can geometric attributes local to the graphics program (such as viewpoint), but also simulation variables (such as internal torques) be modified with the attached dials during the simulation, and the compute server can also use these updated values in the next time step.

5 Conclusion

The ultimate goal for the physically correct motion of multibody systems is a task-level control that derives the necessary internal torques from a data base that was established by learning (e.g. under the control of a neural network) and by reacting to external feedback (e.g. vision). However, until then, a lot of trial and error processing will be necessary, demanding interactive control and real-time visualization of the dynamic simulation.

References

Allrutz R, Rabenseifner R (1992) Der DFN-RPC, ein Remote-Procedure-Call Tool. Proc DECUS München Symposium 1992, pp 523–532

Armstrong WW, Green MW (1985) The dynamics of articulated rigid bodies for purposes of animation. Visual Comput 1:231–240

Boulic R, Magnenat-Thalman N, Thalmann D (1990) Human free-walking model for a real-time interactive design of gaits. Proc Comput Animat 1990, pp 61–80

Cohen MF (1989) Gracefulness and style in motion control. Proc Mechanics, Control and Animation of Articulated Figures. Morgan Publisher

Gruber K, Denoth J, Stüssi E, Ruder H (1987) pp. In: International series on biomechanics 6 B. The Wobbling Mass Model Human Kinetics Publishers, Inc. Champaign, Illinois, pp 1095–1099

Gruber K, Denoth J, Ruder H, Stüssi E (1991) Zur Mechanik der Gelenkbelastung. Grundlagen für die theoretische Modellierung. Z Orthop, pp 260–267

Rabenseifner R (1992) Proc GI Workshop Visualisierung – Rolle von Interaktivität und Echtzeit 1992, pp 198–203

Ruder H, Ertl T, Gruber K, Günther M, Hospach F, Ruder M, Subke J, Widmayer K (1991) Kinematics and dynamics for computer animation. Eurographics technical report series EG. 1991 Turorial. Note 12

Shampine LF, Gordon MK (1975) Computer solutions of ordinary differential equations. Freeman, San Francisco

Selbie S (1989) An introduction to the use of dynamic simulation for the animation of human movement. Proc Comput Animat 1989, pp 33–45

Wilhelms JP, Barsky BA (1985) Proc Graph Interface 1985, pp 197–204

Zeltzer D (1982) Motor control technique for figure animation. IEEE Comput Graph Appl 2:53–59

THOMAS ERTL studied physics and computer science at the University of Erlangen/Nürnberg and at the University of Colorado at Boulder, from where he received a M.S. degree in computer science in 1982. Working on the numerical modelling of pulsar magnetospheres he received a Ph.D. in theoretical astrophysics in 1988 form the University of Tübingen. His current research interests are visualization in scientific computing, parallel processing and workstation-supercomputer-integration.