EUROGRAPHICS 2001 / A. Chalmers and T.-M. Rhyne
(Guest Editors)

Volume 20 (2001), Number 3

A Dynamic Motion Control Technique for Human-like Articulated Figures

Masaki Oshita and Akifumi Makinouchi

Department of Intelligent Systems, Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan

Abstract

This paper presents a dynamic motion control technique for human-like articulated figures in a physically based character animation system. This method controls a figure such that the figure tracks input motion specified by a user. When environmental physical input such as an external force or a collision impulse are applied to the figure, this method generates dynamically changing motion in response to the physical input. We have introduced comfort and balance control to compute the angular acceleration of the figure's joints. Our algorithm controls the several parts of a human-like articulated figure separetely through the minimum number of degrees-of-freedom. Using this approach, our algorithm simulates realistic human motions at efficient computational cost. Unlike existing dynamic simulation systems, our method assumes that input motion is already realistic, and is aimed at dynamically changing the input motion in real-time only when unexpected physical input is applied to the figure. As such, our method works efficiently in the framework of current computer games.

1. Introduction

Generating realistic character animation is a difficult challenge. Recently, many online applications such as computer games and virtual environments require the generation of realistic and continuous character animation in real-time. Currently, such animations are generated by dynamically composing motion sequences such as motion capture or keyframed motion data. These motion sequences need to be created in advance. Therefore, it is difficult to produce dynamically changing motion that respond to physical input from the environment, such as the gravitational force when carrying a heavy load, an external force, or a collision impulse from other objects. This kind of interaction between a character and the environment are frequent and important events in computer games. Nevertheless, very few methods have been developed for dynamic motion control in such situations. This is one of the most important issues in real-time character animation.

This paper presents a dynamic motion control technique for human-like articulated figures. This method controls a character based on input motion specified by a user, and environmental physical input in a physically based character animation system. In the system, the angular acceleration of character's joints are controlled so as to track the usr-input motion. Dynamic simulation then generates the resulting animation. When environmental physical input is applied to the character, the dynamic motion control computes the angular joint accelerations in order to produce dynamically changing motion in response to the physical input. We introduce two kinds of dynamic control; comfort and balance control. Under comfort control, when a torque on a joint exceeds the available muscle strength of the joint, the angular joint accelerations are controlled so as to reduce the joint stress based on the moment of inertia. Under balance control, when the character is likely to lose balance, the angular joint accelerations are controlled so as to maintain balance. This approach produces human-like dynamic motion control, such as reducing the stress on the back by swing the arms and maintaining balance by moving the pelvis, when the character carries a heavy load or collides with other objects. This dynamic motion control method is specific to human-like articulated figures, controlling the arms, back and legs separetely in order of importance. Each part is controlled through the minimum number of degrees-of-freedom (DOF). A number of minor factors are ignored in this method and the resulting motion is not perfectly physically correct. However, our method makes it possible to simulate realistic human motions at lower computational cost because the method does not include heuristics. The goal of our method was not to establish a stable control method but to produce realistic character reactions in response to physical interactions.

A number of techniques have been developed for generating character animation in real-time using dynamic simulation. However, most of these methods are aimed at generating physically correct motion from unnatural input motion such as specified keyframes and monotonous procedural motion. Because these methods cannot utilize existing realistic motion sequences such as motion capture data, they are not used in many applications. Our method assumes that input motion is already realistic, and is aimed at dynamically changing the input motion only when unexpected physical input is applied to the figure from the environment. As such, our method works efficiently in the framework of current computer games and other online applications.

The reminder of this paper is organized as follows. Section 2 reviews related work and issues relevant to solve our problem. Section 3 describes the structure of proposed system and its components. Section 4 presents a simple tracking control algorithm to track an input motion directly. Based on the tracking controller, section 5 then introduces a dynamic control algorithm for comfort and balance control. In section 6, an experimental result is presented, and section 7 concludes this paper and outlines future research.

2. Related Work

There are two main approaches for generating or editing physically correct motion based on dynamics; spacetime constraints and dynamic simulation. In addition, there are motion control techniques using dynamics for specific task.

2.1. Spacetime Constraints

In the spacetime constraints approach [20], an optimal motion trajectory is automatically computed such that the resulting motion minimizes an objective function based on spacetime constraints specified by the user. Rose et al. [15] adapted this approach to articulated figures and proposed a keyframe interpolation technique in whitch the required torque, calculated using inverse dynamics for each motion segment between specified keyframes is minimezed. Komura et al. [9] introduced a musculoskeletal model and an objective function for minimizing muscle strength, thus allowing input motion to be retargeted to other characters with different musculoskeletal models. Tak et al. [16] proposed a motion balance filtering technique that modifies an input motion sequence such that the balance of the figure is maintained during motion making dynamic adjuctments to the trajectory of the zero moment point (ZMP). These methods are effective for converting input motion to more realistic motion. Recently, Popović and Witkin [14] proposed a transformation technique based on a spacetime constraint approach and dynamics. This method involves extracting the essential physical characteristics from an original motion for the simplified model using the spacetime constraints approach, and then modifing the extracted dynamics and reconstruct the resulting motion for the original articulated figure. By this method, the dynamics of an existing motion sequence can be modified easily. However, this method does not model the character's skeleton or strength. Furthermore, no human-like dynamic control is employed.

Although the spacetime constraint technique makes it possible to edit motion, ensuring both controllability and physical realism, the technique is difficult to apply practically for two reasons. First, animations cannot be produced in real-time because solving an optimal problem requires significant computational time, hence an offline process. Second, because the spacetime constraint technique controls motion in angular space, it is difficult to realize motion that interacts dynamically with the environment. During static motion, joint stress and overall body balance depend on primarily joint angle. However, during dynamic motion, the effect of the moment of inertia due to angular joint acceleration should also be considered. The spacetime constraint technique controls joint angle, and indirectly controls angular joint acceleration. The technique is suitable for planning stable motion by minimizing joint stress and maintaining balance before motion is initiated. However, the technique remains unsuitable for dynamic control in which the figure is required to incidental physical input during motion.

2.2. Dynamic Simulation

Dynamic simulation methods use a dynamic controller to compute joint torques based on the current state and the desired motion. Forward dynamics simulation then generates the resulting motion based on joint torque. A number of researchers have developed dynamic controllers for a specific character skeletons and behavior, such as for walking [3] [18] and athletic movements [6]. These controllers use proportional-derivative (PD) servos to compute joint torque based on the desired and current angle for each joint. The PD controller determines the output torque in proportion to the difference between the desired state θ_d, $\dot{\theta}_d$ and the current state θ, $\dot{\theta}$ (vector of angles and angular velocities, respectively) according to

$$\tau = k_p(\theta_d - \theta) + k_v(\dot{\theta}_d - \dot{\theta}). \quad (1)$$

The PD controller is easy to implement. However, the technique does not account for the dynamic characteristics of the system. Therefore, to produce stable and natural motion, the proportional gains k_p and k_v need to be tuned empirically for both the character and motion. van de Panne [18] developed an optimization technique so as to tune the various parameters of gait motion. Hodgins and Pollard [5] proposed a transformation technique that transforms a successful controller to another character. However, it remains difficult to construct a controller that works successfully. Furthermore, because these systems combine dynamic controllers and reference motion generators that are specific for a particular task, it is difficult to adopt the model for other motions.

Recently, more advanced controllers for tracking kinematically specified general motion sequences have been proposed. Zordan and Hodgins [21] proposed a dynamic controller for general human upper-body motion. They combined the PD controller and optimal control in their system, optimal parameters k_p and k_v are determinded so as to minimize the error between the desired and produced motion sequences. However, because determining the parameters requires an offline process, this method is not suitable for real-time applications. Kokkevis et al. [8] introduced model reference adaptive control (MRAC) as a replacement for PD control. They developed a MRAC controller that takes a convergence speed of the reference model as the parameter instead of gain parameters.

Existing dynamic simulation methods are aimed at generating physically correct motion based on physically unnatural input motion such as manually specified keyframes and monotonous procedural motions. Using dynamic simulation, these methods generate motion that reflects a figure's physical properties and accounts for external input such as external force or impact. However, these methods do not include active control such as the comfort and balance controls presented in this paper. These methods control each DOF separately, failing to consider the effect of joint torque on the angular acceleration of other joints.

2.3. Motion Control using Dynamics

A number of techniques have been developed for generating dynamically controlled motion based on dynamics for a particular kind of task. Lee et al. [11] introduced a muscle strength model into the inverse kinematics method, modifying the trajectory of an end-effecter and motion speed based on the muscle strength of the joints. Boulic et al. [2] developed the inverse kinetics method to control the trajectory of the center of mass of an articulated figure. These methods make it possible to create motion that includes comfort and balance control. However, they are unable to handle the change of velocity of an articulated figure due to a collision impulse, nor can they make use of existing motion data.

Ko and Badler [7] developed system that produces a human walking motion with balance and comfort control using inverse dynamics. They combined a walking motion generator and dynamic modification of the generated walking motion. The system transforms the positions of the pelvis and torso during the generated walking motion, and controls walking speed in response to the joint torques calculated by inverse dynamics in real-time. However, the computation of displacement does not include dynamics and remains dependent on empirically tuned parameters. Furthermore, the method is unable to handle interactions with the environment.

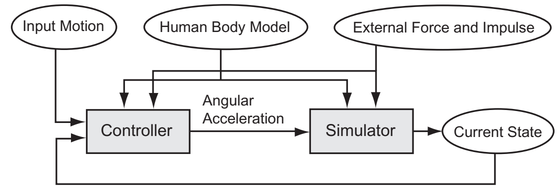

Figure 1: *System structure.*

3. System Description

The structure of the animation system presented in this paper is shown in Figure 1. The system consists of two main module; a controller and a simulator. At each simulation step, the controller computes the angular joint acceleration of the figure, based on the current state of the figure and the motion input by a user. The simulator then updates the state of the figure through dynamic simulation. A human body model and external physical input are considered in both the controller and the simulator. Unlike standard controllers [6, 8, 21] control joint torque, our controller controls angular joint acceleration directly. No forward dynamics are used in our system. Instead, inverse dynamics is used in the controller to take into account the torque required to realize a given angular acceleration. The algorithm for the controller is presented in detail in section 4 and 5. The remainder of this section explains the other components in the system.

3.1. Human Body Model

The human body model considered by this method is as an articulated figure, which is a common representation in character animation. The articulated figure consists of segments and joints; each rigid segment is connected by one, two, or three rotational joints. For example, the shoulder has three joints and the elbow has one. Based on this skeleton model, the configuration of a figure is represented by the set of angles of all joints and the position and orientation of the root segment. In addition, each segment has physical properties relevant to dynamic simulation, such as mass and moment of inertia. These properties are calculated from the polygonal geometry of each segment [6]. The polygonal geometries also are used for collision detection and for computing the contact surface between the segment and the ground. For our experiments, we use a skeleton model that has 18 segments and 39 joints (Figure 2).

The dynamic controller uses the available muscle strength of each joint as the criterion for comfort control. We adopt a simple muscle strength model [7, 11] in whitch two muscle strength functions; the maximum and minimum available torque, are assigned to each joint. Pandya et al. [13] showed by collecting human strength data that these values can be approximated by functions of the joint angle and angular velocity. We assigned approximated strength functions to

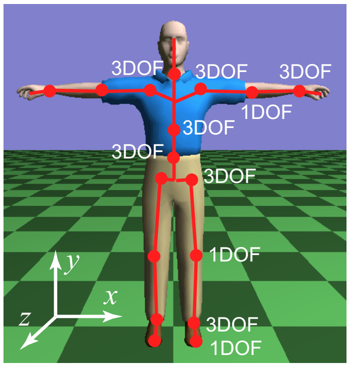

Figure 2: *Human skeleton model.*

each joint, taking into account references including muscle strength data [13] [11].

3.2. Motion Representation

Desired motion is specified in terms of the displacements of the configuration of a figure over time. Therefore, motion data are expressed as the angular trajectories of all joints and the spatial and orientational trajectory of the root segment. In addition, while a foot is in contact with the ground, the joints of the leg is controlled such that the foot is held in the same position (as explained in section 4.2). Therefore, the time when each foot lands on the ground and leaves again should also be indicated. As input motion is represented kinematically, any form of motion capture data or keyframe motion sequence can be used as an input to our system.

3.3. Dynamic Simulation

Given the angular accelerations of all joints, the simulator updates the angles and angular velocities of all figures by Eular integration [?]. In addition to the angular acceleration of the joints, the rotational acceleration of the supporting segment of the figure (e.g. foot) is computed based on the angular joint acceleration, simulating falling motion. The segment upon which the moment of the center of mass of the figure is maximum is chosen as the supporting segment. To compute the rotational acceleration of a supporting segment, we use the zero moment point (ZMP) and minimum moment point (MMP). The details of the concept of the ZMP are explained in [16]. The ZMP is the point where the torque exerted by the figure on the ground is zero. When ZMP is within the support area (Figure 3(a)), the figure is balanced and there is no rotational acceleration of supporting segment. Otherwise, rotational acceleration occurs around the MMP where the exerted torque is minimum. The MMP is the closest point from the ZMP within the support area (Figure 3(b)). The support area is the convex hull of contact surfaces between the foot segments and the ground. The rotational acceleration of the supporting segment is computed from the torque exerted on the MMP and the moment of inertia of the whole body in that configuration. After the integration, collision detection and response are performed. When two figures collide, an impact force is imparted on each and their velocities change. The velocity changes are computed by solving the linear equation [12] [8]. If the figures remain in contact, a reaction force acts between them. Reaction forces and other external forces are considered in the inverse dynamics component of the dynamic controller.

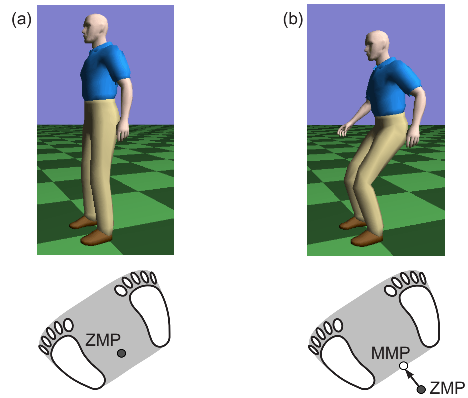

Figure 3: *ZMP in (a) balanced and (b) unbalanced state.*

4. Tracking Control

This section presents the algorithm used to compute the angular acceleration of all joints so as to track the desired motion, based on the current state of the figure and the desired motion. This algorithm controls joint angular acceleration directly rather than via joint torque, witch is the case in standard dynamic simulation systems. As result, the desired motion is almost exactly tracked. However, unlike standard animation and game systems in which the joint angles of a figure are directly controlled according to a desired motion trajectory, our tracking controller produces continuous motion that approaches the desired motion even when the velocity of the figure is changed through a collision. In addition, when a figure loses its balance, a falling motion is generated as explained in section 3.3. The algorithm presented here is a simple and direct tracking control. A more advanced dynamic control for realizing human-like movements is presented in the next section as an extension of this tracking control system. This tracking control also can be used alone, if a user requires only continuous motion and lower computational cost. This tracking control scheme does not require dynamics computations or muscle strength model, making it easily to implement, with low computational cost.

4.1. Angular Acceleration of Each Joint

The angular acceleration for each joint is computed based on the figure's current state (joint angle and angular velocity), and the angular trajectory of the desired motion. As reviewed in section 2.2, PD control servos are widely used for this purpose in existing dynamic simulation systems [6 3 18]. Using a PD controller, the output angular acceleration is computed using the following equations;

$$\ddot{\theta} = k_p(\theta_d - \theta) + k_v(\dot{\theta}_d - \dot{\theta}) \qquad (2)$$

where $(\theta, \dot{\theta})$ is the current joint angle and angular acceleration, $(\theta_d, \dot{\theta}_d)$ is the desired state obtained from a desired joint angular trajectory after Δt, and k_p and k_v are the gain parameters. The parameters need to be tuned for each joint and each motion, making it difficult to contruct a general controller by this approach. In addition, to realize a stable control, a controller should take into account not only one state in the desired angular trajectory after Δt, but also the entire trajectory.

Therefore, we have decided to use a Ferguson curve to compute output angular acceleration. A Ferguson curve is a kind of interpolation curve, such as a B-Spline or Bezier curve. However, while other spline curves are defined by a set of values $(time, value)$ at knot points, a Ferguson curve is defined by $(time, value, derivative)$ at knot points. This feature makes a Ferguson curve suitable for use in our method because the current and desired state of a joint is defined by the joint angle and angular velocity. To compute the output angular acceleration, this method first determines the target point for which the motion is calculated to approach, taken as the closest extremity point to the desired angular trajectory. The projected trajectory, from the current state to the state approaching the desired position of the target point, is approximated by a Ferguson curve, as follows;

$$\begin{aligned} q(s) \;=\; & (2s^3 - 3s^2 + 1)\theta + (-2s^3 + 3s^2)\theta_t + \\ & (s^3 - 2s^2 + s)\dot{\theta} + (s^3 - 2s^2 + s)\dot{\theta}_t \end{aligned} \qquad (3)$$

$$T = target\_time - current\_time, s = \frac{t - current\_time}{T} \qquad (4)$$

where $(\theta_t, \dot{\theta}_t)$ is the desired state of the target point in the desired angular trajectory. By taking the second derivative of the trajectory and letting $s = 0$, the output angular acceleration can be determined, written as

$$\ddot{\theta} = \left\{6\theta_t - 6\theta - 2\dot{\theta}_t - 4\dot{\theta}\right\} / T^2 \qquad (5)$$

While a target point is fixed, the output angular acceleration

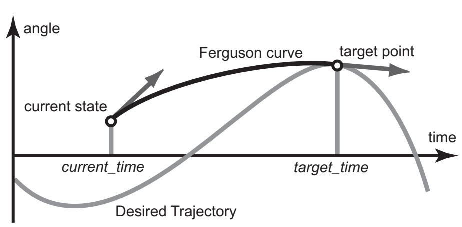

Figure 4: *Tracking control using Ferguson curve.*

is continuous. At the target time, the output angular acceleration becomes discontinuous, as the target point becomes the next extremity point in the desired trajectory. However, the effect on the output angular acceleration is minimal as long as the current state is close to the desired trajectory. In addition, an instant of discontinuity has little influence on the motion trajectory in angular space. Therefore, realistic continuous motion is always realized.

4.2. Angular Acceleration of Limb Joints

When several limbs (arms and legs) of the figure are constrained, all joints in the limbs should be controlled cooperatively. For example, during a double support phase, the joints in both legs should be controlled such that neither foot leaves the ground, and if a figure is holding a ladder with the right arm and right leg, all joints in both limbs should be controlled cooperatively.

We use a human body model in which each limb has 7 DOFs (Figure 2). The angles of the 7 joints are determined from the position and orientation of the pelvis (p, o) (6 DOFs) and the swivel angle of the knee around the vector from the hip joint to the ankle joint s (1 DOF), by analytical inverse kinematics [17 10]. The tracking algorithm determines the spatial and rotational accelerations of the root segment $(\ddot{p}, \ddot{o})$ and the swivel angular acceleration $\ddot{s}$ for constrained limbs using the tracking algorithm presented in section 4.1. The angular acceleration of the joints of the constrained limbs are computed using an analytical inverse kinematics method in the same way as inverse kinematics were used for joint angles. The inverse kinematics algorithm for angular accelerations is easily derived from the inverse kinematics method for angles [17 10].

5. Dynamic Motion Control

This section introduces a dynamic control method to compute an output angular acceleration in response to physical input from the environment. The angular acceleration computed by the tracking control algorithm in the previous section is used as the initial angular acceleration. The output angular acceleration is defined as the sum of the initial acceleration $\ddot{\theta}_{initial}$ and the difference of the angular acceleration $\Delta\ddot{\theta}$ in dynamic motion control as follows;

$$\ddot{\theta}_{output} = \ddot{\theta}_{initial} + \Delta\ddot{\theta}. \qquad (6)$$

Here, $\ddot{\theta}_{output}$, $\ddot{\theta}_{initial}$ and $\Delta\ddot{\theta}$ are n-dimensional vectors, where n is the total number of joints. Each row of the vectors corresponds to a single joint. Comfort and balance control are used in our method to relize dynamic motion control. Under comfort control, when a torque exerted on a joint exceeds the available muscle strength of the joint, joint angular accelerations are controlled so as to reduce the joint stress based on the moment of inertia. Under balance control, when a character is likely to lose balance, angular joint acceleration is controlled so as to maintain balance. When the joint torque is within the available muscle range for all joints and the body balance is maintained on $\ddot{\theta}_{initial}$, no dynamic control is performed and the controller outputs the initial acceleration $\ddot{\theta}_{initial}$ as the output acceleration. In this way, motion close to the desired motion trajectory is realized.

5.1. Control Foundation

The criteria for comfort and balance control are introduced here in terms of the dynamics of articulated figures. The criteria and the use of comfort and balance control is not novel work. The novel part of our work is the dynamic control algorithm that controls angular joint acceleration. Here, the relationship between the criteria and the angular acceleration of a joint is derived for the dynamic control altorithm sedcribed in section 5.3.

5.1.1. Comfort Control

Joint torque that exceeds available muscle strength is considered as the criterion for comfort control. The joint torque τ required to produce the joint acceleration $\ddot{\theta}$ is computed by an inverse dynamics method, expressed as

$$\tau = H(\theta)\ddot{\theta} + C(\theta, \ddot{\theta}) + G(\theta) + F(\theta) \qquad (7)$$

where $H(\theta)$ is the moment of inertia, and $C(\theta, \ddot{\theta})$, $G(\theta)$ and $F(\theta)$ are the influences on torque due to coriolis and centrifugal forces, gravity, and external force, respectively. The dimension of all vectors is n, where n is the number of joints of the figure. For the inverse dynamics, we use the Newton-Eular method [4]. During a double support phase, an approximation [7] [9] is used to determine the forces applied from the upper body to each leg. The required torque of all joints is computed in $O(n)$. The available torque of the ith joint is depend on $(\theta_i, \ddot{\theta}_i)$ as explained for the muscle strength model in section 3.1 and given by

$$\tau_{max,i} = f_{max,i}(\theta_i, \dot{\theta}_i), \;\; \tau_{min,i} = f_{min,i}(\theta_i, \dot{\theta}_i). \qquad (8)$$

The requiered change in joint torque to satisfy the muscle strength constraint is computed for each joint by the following equation;

$$\tau_{stress,i} = \begin{cases} \tau_i - \tau_{max,i}, & \text{if } \tau_i > \tau_{max,i} \\ \tau_i - \tau_{min,i}, & \text{if } \tau_i < \tau_{min,i} \\ 0, & \text{if } \tau_{min,i} < \tau_i < \tau_{max,i} \end{cases} \qquad (9)$$

Comfort control is performed so as to minimize $\tau_{stress,i}$ for all joints.

The relationship between the required change in joint torque $\Delta\tau$ and the corresponding change in angular joint acceleration can be derived from equation (7). The relationship is dependent on the moment of inertia, as follows;

$$\Delta\tau = H(\theta)\Delta\ddot{\theta}. \qquad (10)$$

Each column of the matrix $H(\theta)$ is computed solely from the current angles in $O(n)$ [19]. Comfort control is performed based on the derivation of the joint torque in equation (10).

5.1.2. Balance Control

The zero moment point (ZMP) and minimum moment point (MMP), explained in section 3.3, are used as the criterion for balance control. The position of the ZMP is computed from the spatial accelerations of all segments on the assumption that the ground is defined as $ZMP_y = 0$ according to the following equations [16];

$$ZMP_x = \frac{\sum m_i x_i(\ddot{y}_i - g) - \sum m_i y_i \ddot{x}_i}{\sum m_i(\ddot{y}_i - g)}, \qquad (11)$$

$$ZMP_z = \frac{\sum m_i z_i(\ddot{y}_i - g) - \sum m_i y_i \ddot{z}_i}{\sum m_i(\ddot{y}_i - g)} \qquad (12)$$

where m_i is the mass of the ith segment, (x_i, y_i, z_i) is the position of the ith segment, and $(\ddot{x}_i, \ddot{y}_i, \ddot{z}_i)$ is the spatial acceleration of the ith segment. As the spatial accelerations of segments are computed from the angular acceleration of joints, the position of the ZMP is represented as a function of angular joint acceleration. When ZMP is outside the support area, the MMP becomes the closest point from the ZMP within the support area. Balance control is performed to move the ZMP to the MMP (Figure 3(b)).

The relationship between the position of the ZMP and the angular acceleration of a joint can be derived from equations (11) and (12) by considering the movements of an augmented body [2], defined as the imaginary rigid body supported by a single joint, consisting of all segments from the joint to the end-effecters. The relationship is shown in Figure 5, where M is the mass of the augmented body, l is the vector from the joint to the center of mass of the augmented body, r is the rotational axis of the joint, and a is the spatial acceleration of the augmented body. Using these variables, the spatial derivation of the ZMP can be computed by

$$\frac{\delta ZMP_x}{\delta \ddot{\theta}_i} = \frac{M}{\sum m_i(\ddot{y}_i - g)} \{(p_x - ZMP_x)a_y - p_y a_x\}, \qquad (13)$$

$$\frac{\delta ZMP_x}{\delta \ddot{\theta}_i} = \frac{M}{\sum m_i(\ddot{y}_i - g)} \{(p_z - ZMP_z)a_y - p_y a_z\}. \qquad (14)$$

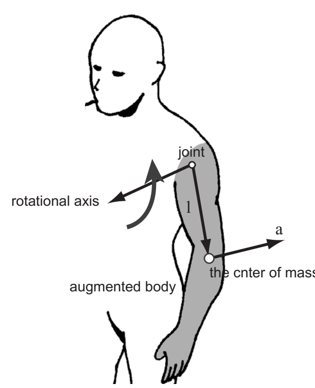

Figure 5: *Velocity of ZMP from rotation of augmented body.*

Balance control is performed based on the spatial derivation of the ZMP in equations (13) and (14).

5.2. Control Strategy

A simple approach for computing $\Delta\ddot{\theta}$ is to solve an optimization problem so as to minimize an objective function such as

$$f(\Delta\ddot{\theta}) = |\tau_{stress}| + |\text{ZMP - MMP}| + |\Delta\ddot{\theta}|. \quad (15)$$

However, solving the optimization problem requires significant computational time because this equation controls a large number of DOFs. Although the objective function is a good strategy for generating robust human motion [9 15 16], it does not reflect motion control based on human experience.

We have developed a control method based on the observation of human movement. The method has been developed dpecifically for human-like figures in a standing double-support phase. First, we categorize dynamic motion control into two types; active and passive control. Under active control, a small number of primary joints are controlled so as to reduce the stress on all joints and to maintain body balance. Under passive control, joints under high stress are controlled so as to reduce their own stress. For example, if a figure has a heavy load in the right hand, active control moves other parts to assist the motion of the right arm by reducing the stress on the right arm, while passive control moves the joints in the right arm based on joint stress.

To perform active control, the human figure is controlled through three parts; the arms, back and legs (Figure 6(a)). We choose primary DOFs for the each part in order to control them efficiently. The arms are controlled through the two angular joint accelerations for each shoulder joint $\Delta\ddot{\theta}_{arms}$ (4 DOFs) (Figure 6(b)). The rotational acceleration of each shoulder around the x-axis and z-axis are controlled; the rotational acceleration around the y-axis is not used because the influence of the motion component on other joints is smaller than that of the other axes in terms of dynamics. If the stress around the y-axis exerted on a joint, the stress is reduced by swinging both shoulder around the x-axis in opposite direction. This means that both shoulders should be controlled corperatively. The back is controlled through the three angular accelerations of the back joint $\Delta\ddot{\theta}_{back}$ (3 DOFs) (Figure 6(c)). The legs are controlled through the spatial acceleration of the pelvis segment $\Delta\ddot{p}_{legs}$ (3 DOFs) (Figure 6(d)), because in this case the legs should be controlled cooperatively so as to satisfy the constraints of both feet, as explained in section 4.2. For the lower body (legs), active and passive control is computed at the same time through $\Delta\ddot{p}_{legs}$. For the upper body (arms, back), the angular acceleration $\ddot{\theta}_{stress}$ (k DOFs) is computed for the passive control of k joints with stress exceeding the available muscle strength. The number k depends on the initial torque due to the initial acceleration $\ddot{\theta}_{initial}$. The difference of the angular acceleration $\Delta\ddot{\theta}$ (n DOFs) in equation (6) is computed by

$$\Delta\ddot{\theta} = S_a\ddot{\theta}_{arms} + S_b\ddot{\theta}_{back} + S_s\ddot{\theta}_{stress} + J_p\ddot{p}_{legs} \quad (16)$$

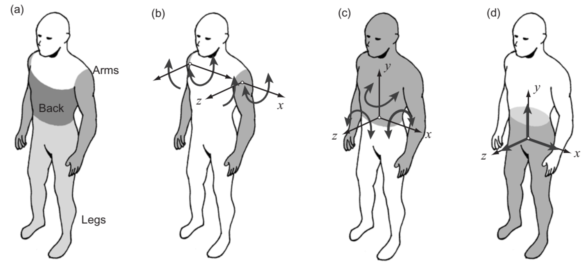

Figure 6: *Control of body parts; (a) all parts, (b) arms, (c) back, and (d) legs*

where S_a, S_b and S_s are the selection matrices that map each controlled joint to the corresponding joint for the body $\Delta\ddot{\theta}$. J_p is the Jacobian matrix ($n \times 3$) that maps the controlled special acceleration to the displacement of all joints in the lower body, computed by inverse kinematics.

5.3. Control Algorithm

In the control algorithm, $\Delta\ddot{\theta}_{arms}$ (4 DOFs), $\Delta\ddot{\theta}_{back}$ (3 DOFs), $\Delta\ddot{p}_{legs}$ (3 DOFs) and $\Delta\ddot{\theta}_{stress}$ (k DOFs) are controlled, each having an effect all the others. This interaction makes it difficult to control all these targets at the same time. Therefore, the algorithm computes each term in order, based on the order of importance.

Active control is applied to the arms, back and legs, in that order. The control of the upper body is more applicable than the control of the lower body in human motion control. The control of the lower body has a significant influence on body balance and the stability of motion, and hence the significant change of the motion of the lower body causes unstable results. Therefore, comfort and balance control, using $\Delta\ddot{\theta}_{arms}$ (4 DOFs) and $\Delta\ddot{\theta}_{back}$ (3 DOFs) are performed first. If the joint stress cannot be reduced or balance cannot be maintained, the lower body is then controlled using $\Delta\ddot{p}_{pelvis}$ (4 DOFs).

In control upper body, passive control is applied before active control. When environmental input is large and the current state is significantly different from the desired motion, the initial acceleration necessarily becomes large. As a result, because the joint stress becomes large and the figure is likely to lose balance, control based on the conditions causes unstable motion. To avoid this, the initial acceleration is first reduced through passive control. Active control is then applied based on the reduced acceleration in order to realize output acceleration close to the initial acceleration. Based on these strategies, the algorithm for dynamic motion control is described as follows;

1. The initial acceleration is computed. $\ddot{\theta}_{initial}$ is computed using the tracking algorithm in section 4.
2. Passive control for upper body. $\Delta\ddot{\theta}_{stress}$ is computed for all stressed joints.

3. Active control for the upper body. $\Delta\ddot{\theta}_{arm}$ then $\Delta\ddot{\theta}_{back}$ are controlled so as to reduce joint stress, and $\Delta\ddot{\theta}_{stress}$ is recomputed as the result of comfort control.
4. Passive and active control for the lower body. $\Delta\ddot{p}_{ppelvis}$ is controlled so as to reduce the stress on joints in the lower body and maintain body balance.
5. Output acceleration $\ddot{\theta}_{output}$ is computed from $\ddot{\theta}_{arms}$, $\ddot{\theta}_{back}$, $\ddot{\theta}_{stress}$, and $\ddot{p}_{legs}$.

5.3.1. Passive Control for Upper Body

Passive control of the upper body involves controlling the change of the angular acceleration of k joints. If the initial angular acceleration of one joint of the k joints is small, then the influence of that joint on other joints is also small. This algorithm controls the angular acceleration of each joint of the k joints separately considering only the moment of inertia H_{ii} affected by the angular acceleration and torque of the individual joints. However, when the current state of a joint differs significantly from the desired motion, the initial angular acceleration of the joint is large and control becomes unstable. Therefore, we compute the change of the angular acceleration of each joint $\Delta\ddot{\theta}_{stress,i}$ in two phases. First, $\Delta\ddot{\theta}'_{stress,i}$ is computed such that $\ddot{\theta}_{initial,i} + \Delta\ddot{\theta}'_{stress,i}$ is realizable within the available torque range of the ith joint when the moment of inertia from other joints is ignored, given by

$$\tau_{min,i} < H_{ii} \cdot (\ddot{\theta}_{initial,i} + \Delta\theta'_{stress,i}) + C_i + G_i + F_i > \tau_{max,i} \tag{17}$$

Second, $\Delta\ddot{\theta}_{stress,i}$ is computed such that $\ddot{\theta}_{initial,i} + \Delta\ddot{\theta}_{stress,i}$ is realizable when the moment of inertia from the angular acceleration of other joints $\ddot{\theta}_{initial} + \Delta\ddot{\theta}'_{stress}$ is considered, given by

$$\begin{aligned}\tau_{min,i} &< H_i \cdot (\ddot{\theta}_{initial} + \Delta\ddot{\theta}'_{stress}) + \\ &H_{ii} \cdot (\Delta\theta_{stress,i} - \Delta\ddot{\theta}'_{stress,i}) + C_i \ + G_i + F_i > \tau_{max,i}\end{aligned} \tag{18}$$

5.3.2. Active Control for Upper Body

Active control of the upper body involves calculating $\Delta\ddot{\theta}_{arm}$ and $\Delta\ddot{\theta}_{back}$, in that order. The rotational acceleration of each part is computed for the comfort control of the jth stressed joint ($\Delta\ddot{\theta}_{arm,cj}$ or $\Delta\ddot{\theta}_{back,cj}$) and for balance control ($\Delta\ddot{\theta}_{arm,b}$ or $\Delta\ddot{\theta}_{back,b}$). The largest acceleration is then used to control the part. When an environment input is applied to figure, the stress of joints and the positional error of the ZMP often occur in the same direction. In that case, the rotational acceleration for rducing the largest stress or for maintaining balance can be expected to help the other stresses and imbalance. If unresolved stresses and imbalance remain, the next part is controlled so as to solve them.

The rotational accelerations $\Delta\ddot{\theta}_{arm,cj}$ and $\Delta\ddot{\theta}_{back,cj}$ for reducing joint stress are computed for the jth composite joint consisting of rotational joints. For example, if the wrist consists of three rotational joints, as in our model, the rotational acceleration required to reduce the stress of the three joints in the wrist $\Delta\ddot{\theta}_{arms,wrist}$ is computed for each joint simultaneously. As mentioned in section 5.1.1, the relationship between the displacements of the ith composite joint and the rotational acceleration of the arms or back is expressed using a submatrix of the moment of inertia matrix $H(\theta)$, given by

$$\Delta\tau_j = H' \Delta\ddot{\theta}_{arms,cj}. \tag{19}$$

The required change of torque $\Delta\tau_j$ is computed from τ_{stress}. The dimension of $\Delta\tau_j$ is always equal to or less than $\Delta\ddot{\theta}_{arms,cj}$. Thus, $\Delta\ddot{\theta}_{arms,cj}$ is redundant. The solution so as to minimize $\left|\Delta\ddot{\theta}_{arms,cj}\right|$ can be computed using the pseudo inverse matrix H'^{+} of H', given by

$$\Delta\ddot{\theta}_{arms,cj} = H'^{+}\Delta\tau_j, \ \ H'^{+} = H'(H'^{t}H')^{-}. \tag{20}$$

The rotational acceleration of the arms for balancing $\Delta\ddot{\theta}_{arms,b}$ is computed in the same way such that the ZMP is moved to the MMP. As described in section 5.1.2, $\delta ZMP / \delta\ddot{\theta}_{arms}$ is computed using equations (13) and (14).

Within the rotational accelerations, $\Delta\ddot{\theta}_{arms,cj}$ is computed for all stressed composite joints and $\Delta\ddot{\theta}_{arms,b}$ is computed for the position of the ZMP, the largest of which is used to control the arms or back. When $\Delta\ddot{\theta}_{arms}$ or $\Delta\ddot{\theta}_{back}$ is too large, the stress on joints in the shoulders or back exceed the available torque. In this case, the rotational acceleration is reduced by passive control using equations (17) and (18).

5.3.3. Active and Passive Control for Lower Body

Control of the lower body is achieved by controlling the change of the spatial acceleration of the pelvis in the same way as active control is applied for the upper body. The change of angular acceleration for all joints in the lower body is controlled indirectly through control of the spatial acceleration of the pelvis. For comfort control, the spatial acceleration of the pelvis is computed for all composite joints in the lower body. The relationship between the joint torques in a composite joint and the spatial acceleration of the pelvis can be derived from equations (16) and (10), written as

$$\Delta\tau_c = H' J_p \ddot{p}_{legs}. \tag{21}$$

For balance control, the relationship between ΔZMP and $\Delta\ddot{p}_{legs}$ is derived from equations (13) and (14) using the weight and center of mass of the upper body. Passive control for the lower body is included in this control algorithm. The pelvis is controlled in the same way to active control for the upper body. The spatial acceleration of the pelvis is computed for the stressed composite joints of the legs and the ZMP. Within the spatial accelerations, the largest acceleration is used to control the lower body. As a result, the joint torque for the output acceleration may exceed the available torque range in this algorithm.

On human movements, when some large stresses work on joints in the lower body or it likely to lose balance, the foot leaves from the ground. During a single phase, by swinging the moving leg or moving the foot to a stable position, more efficient and flexible control is achievable. However, to realize this kind of control, the motion needs to be controlled not only in angular acceleration space but also in angular space. This is beyon the scope of this paper. Therefore, the current algorithm is unable to control a figure successfully, when excessive forces or impulses are applied to the figure and the leg must be moved for stabilization. In such case, joint stresses on some joints are ignored, or the figure falls down by losing balance. Combining our method and motion planning in angular space is one of the most important directions of future work.

6. Results

In this section, we present an experimental result. We created animations based on a keyframe motion sequence and environmental physical input. We used a squatting motion as input. The trajectories of the input motion were represented by a B-Spline. The interval between each frame of dynamic simulation was 1/30 second. Figure 7 shows the images of the generated animation. In Figure 8*, both the input and generated motion are renderd as stick figure, and the control informatin is visualized using arrows. When no physical input is applied (a), the input motion was almost directly tracked. With an 8 kg weight (b), the arms are controlled for balance and to reduce the stress on the back. With a 15 kg weight (c), because active control of the arms could not sufficiently reduce the stress on the back, the back was forced to bend. Subsequently, the figure recovered to the input motion by swinging the arms. In the last animation (d), an impulse is applied to the figure from the front at the first frame. After the impact, the figure attempted to track the input motion while maintaing balance. These animations show that our method produces dynamically changing motion based on input motion and environmental physical input.

The computational time for dynamic motion control on the generated animation is shown in Table 1. The computational time become large when the joint torque of many joints in the initial angular acceleration exceeded the available range because comfort control is computed for each stressed joint. The computational time required for one step of dynamic motion control was 3 milliseconds in the worst case (c). This system generated the animations in real-time.

interaction	total	average	max
(a) no interaction	70.6	0.78	1.0
(b) with 8kg weight	78.8	0.89	2.4
(c) with 15kg weight	102.9	1.43	3.1
(d) impulse applied	72.9	0.81	2.5

Table 1: *Computational time (milliseconds) for dynamic motion control on PC (Pentium III, 800 MHz), the total time for 90 frames, average frame time, and maximum frame time.*

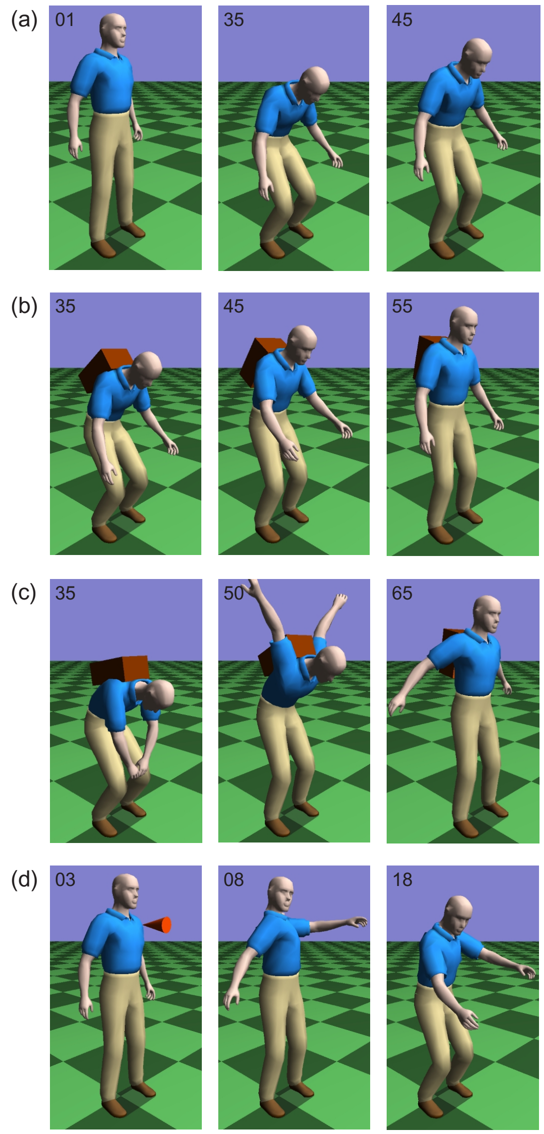

Figure 7: *Squatting animation; (a) no environmental physical input, (b) with 8 kg weight, (c) with 15 kg weight, and (d) impulse applied at the first frame. The numbers on the corner of images shows frame number.*

7. Conclusion

In this paper, we presented a dynamic motion control technique for human-like articulated figures. The key idea of the mothod is to control the joints of a figure in angular acceleration space. This approach ensures continuous and realistically changing motion. The algorithm controls each part of the figure through a minimum number of DOFs, and computes the output angular acceleration in carefully designed steps. This approach has made it possible to generate dynamically changing motion in real-time. In experiments, our system successfully generated chainging motions in response to the weight of a load and an external impulse.

As discussed in section 5.3.4, an important future research goal is to combine the current tracking control technique in angular acceleration space with a dynamic motion planning technique in angular space. In addition, we are going to introduce a more realistic muscle strength model [9]. Using the strength model, animation accounting for the muscle strength of the character will be generated.

Physically based approaches are yet to be widely adopted in computer games. However, such applications require dynamically and realistically changing motion, otherwise are limited to replaying motion sequences created in advance. We believe that the proposed technique will break the limitations of physically based approaches.

Acknowledgements

The work was partially supported by the Grant-in-Aid for Scientific Research (10308012) from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

1. David Baraff, and Andrew Witkin. Physically Based Modeling: Principles and Practice. *SIGGRAPH '97 Course Notes*, 1997.
2. Ronan Boulic, Ramon Mas-Sanso, and Daniel Thalmann. Complex Character Positioning Based on a Compatible Flow Model of Multiple Supports. *IEEE Transactions on Visualization and Computer Graphics*, **3**(3):245–261, July-September 1997.
3. Armin Bruderlin, and Thomas W. Calvert. Goal-Directed, Dynamic Animation of Human Walking. *Computer Graphics (SIGGRAPH '89 Proceedings)*, **23**(3):233–242, 1989.
4. Roy Featherstone. *Robot Dynamics Algorithms*. Kluwer, 1987.
5. Jessica K. Hodgins, and Nancy S. Pollard. Adapting Simulated Behaviors For New Characters. *SIGGRAPH '97 Proceedings*, pages 153–162, 1997.
6. Jessica K. Hodgins, Wayne L. Wooten. David. C. Brogan, and James F. O'Brien. Animating Human Athletes. *SIGGRAPH '95 Proceedings*, pages 71–78, 1995.
7. Hyeongseok Ko, and Norman I. Badler. Animating Human Locomotion with Inverse Dynamics. *IEEE Computer Graphics and Applications*, **16**(2):50–59, 1996.
8. Evangelos Kokkevis, Dimitris Metaxas, and Norman I. Badler. User-Controlled Physics-Based Animation for Articulated Figures. *Proceedings of Computer Animation '96*, 1996.
9. Taku Komura, Yoshihisa Shinagawa, and Tosiyasu L. Kunii. Creating and retargeting motion by the musculoskeletal human body model. *The Visual Computer*, **16**:254–270, 2000.
10. Jehee Lee, and Sung Youg Shin. A Hierarchical Approach to Interactive Motion Editing for Human-like Figures. *SIGGRAPH '99 Proceedings*, pages 39–48, 1999.
11. Philip Lee, Susanna Wei, Jianmin Zhao, and Norman I. Badler. Strength Guided Motion. *Computer Graphics (SIGGRAPH '90 Proceedings)*, **24**(3):253–262, 1990.
12. Matthew Moore, and James Wilhelms. Collision Detection and Response for Computer Animation. *Computer Graphics (SIGGRAPH '88 Proceedings)*, **22**(3):289–298, 1988.
13. Abhilash K. Pandya, James C. Maida, Ann M. Aldridge, Scott M. Hasson, and Barbara J. Woodford. The Validation of a Human Force Model To Predict Dynamic Forces Resulting From Multi-Joint Motions. *Technical Report 3206*, NASA, Houston, Texas, 1992.
14. Zoran Popović, and Andrew Witkin. Physically Based Motion Transformation. *SIGGRAPH '99 Proceedings*, pages 11-20, 1999.
15. Charles Rose, Brian Guenter, Bobby Bodenheimer, and Michael F. Cohen. Efficient Generation of Motion Transitions using Spacetime Constraints. *SIGGRAPH '95 Proceedings*, pages 147-154, 1995.
16. Seyoon Tak, Oh-young Song, and Hyeong-Seok Ko. Motion Balance Filtering. *Computer Graphics Forum*, **19**(3):435–446, 2000.
17. Deepak Tolani, Ambarish Goswami, and Norman I. Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical Models*, **62**:353-388, 2000.
18. Michiel van de Pann. Parameterized Gait Synthesis. *Computer Graphics and Applications*, **16**(2):40–49, March 1996.
19. M. W. Walker, and D. E. Orin. Efficient Dynamic Computer Simulation of Robotic Mechanisms. *Journal of Dynamic Systems, Measurement, and Control*, **104**:205–211, September 1982.
20. Andrew Witkin, and Michael Kass. Spacetime Constraints. *Computer Graphics (SIGGRAPH '88 Proceedings)*, **22**(4):159-168, 1988.
21. Victor B. Zordan, and Jessica K. Hodgins. Tracking and Modifying Upper-body Human Motion Data with Dynamic Simulation. *Computer Animation and Simulation '99 (Proceedings of Eurographics Workshop on Animation and Simulation '99)*, 1999.

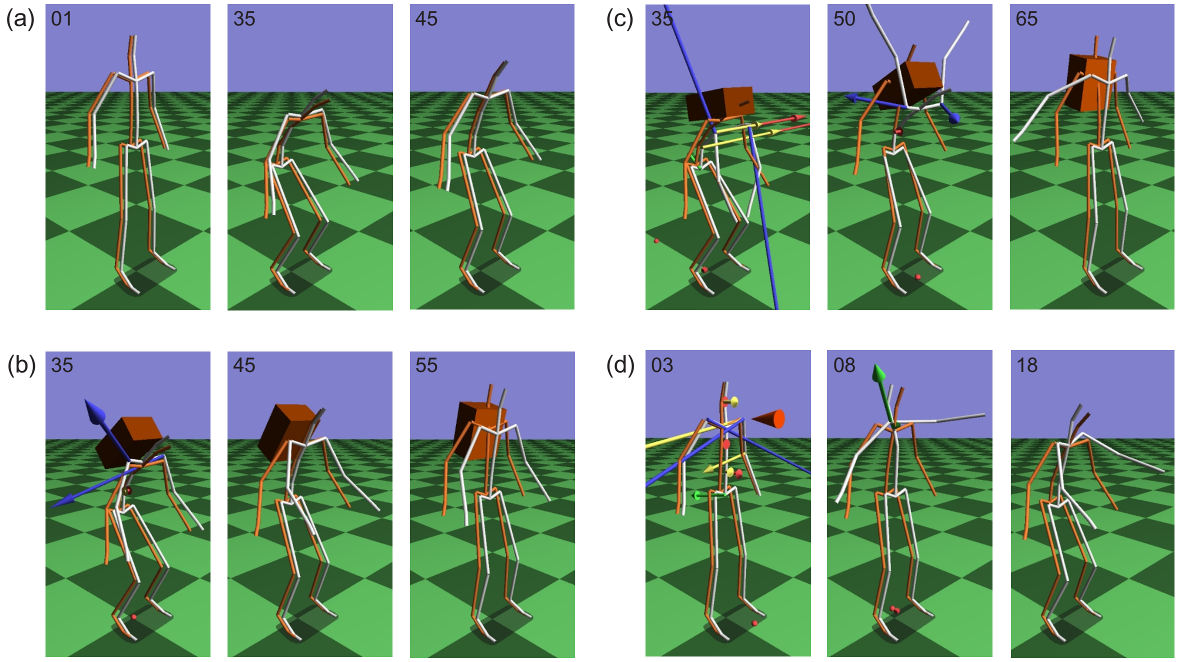

Figure 8: *Squatting animation; (a) no environmental physical interaction, (b) with 8 kg weight, (c) with 15 kg weight, and (d) impulse applied at the first frame. In the images, the figure in orange is the input motion, and the figure in white is the generated motion. A red arrow at joints indicates the stress on the joint. Blue, green and yellow arrows at the joints indicate comfort, balance and passive control, respectively.*