Real-Time Interactive Motion Generator of Human Figures

Yoshihiko NAKAMURA*1*2 and Katsu YAMANE*1
(E-mail: [nakamura, katz]@ynl.t.u-tokyo.ac.jp)

*1 Department of Mechano-Informatics, University of Tokyo
7-3-1 Hongo Bunkyo-ku Tokyo, 113-8656 JAPAN

*2 CREST, Japan Science and Technology Corporation

Abstract

This paper summarizes our research related to motions of human figures. Our final goal is to develop a motion generating system that creates realistic motions of human figures interacting with the user in real time, based on computational methods for the dynamics and kinematics of kinematic chains. Toward this goal, we propose the concept of dynamics filter which generates physically natural motions from any reference motion allowing interactive inputs by the user. Dynamics filter has a potential ability to generate virtually infinite variety of motions from a relatively small motion database or even motions captured from a human in real time. We have also developed a prototype of the real-time dynamics filter, which makes use of our efficient computing methods for the dynamics of human figures.

1. Introduction

Human figures are defined as kinematic models of the human body with links connected by mechanical joints (Figure 1). Motion generation of human figures is of great interest in robotics as well as in computer graphics. Humanoid robots are expected to work in place of human in various fields such as welfare, plant maintenance, entertainment, and so on. In computer graphics, the demand for high quality content with human characters is growing stronger along with the development and spread of multimedia technologies. The common keyword connecting these two issues is the motion of human figures. Motions of humanoid robots are neither simple nor predefined as in conventional industrial robots — we have to control the unstable body of robots to create almost infinite variation of motions. This situation is quite similar to human characters in computer animations, where we have to make various motions according to their role in virtual environments.

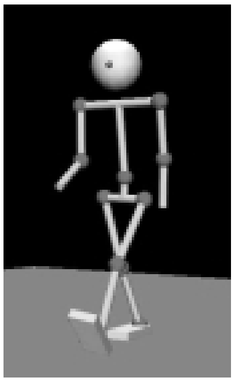

Figure 1: A human figure

To achieve this goal, we propose the concept of a dynamics filter, a real-time and interactive motion generator for human figures. The basic function of the filter is to convert a physically inconsistent motion into a consistent one, to which the following features are added:

1. Take human motions, either captured or drawn, as reference instead of created from scratch, in order to generate expressive human-like motions that convey emotion.

2. Execute the computation in real time in order to allow interactive inputs or adapt to dynamically changing environments

In this paper, we propose the concept of dynamics filter and introduce our first implementation based on our previous research on dynamics of structure-varying kinematic chains [1]. Before describing the implementation of a dynamics filter in detail, we first provide a summary of the methods for computing the dynamics of human figures, followed by the details on our dynamics filter. We also provide some examples of applying the dynamics filter to various motions. In the last section, we introduce our ongoing projects along with the conclusions of this paper.

2. The Concept of Dynamics Filter

Many methods have been proposed for generating motions for humanoid robots [2, 3, 4] and animations [5, 6, 7]. However, most of them have problems from the standpoint of interactivity:

1. Poor flexibility — Each method is only applicable to limited motions, walking in most cases, which means that we need to prepare many motion generators corresponding to different motions.
2. Off-line computation — The whole sequence of motion is required before generating motion. This would be a fatal disadvantage in interactive motion generation because we cannot modify the reference motion once the computation starts.
3. Long computation time — Another problem is that they require much computation to generate a single sequence of motion.

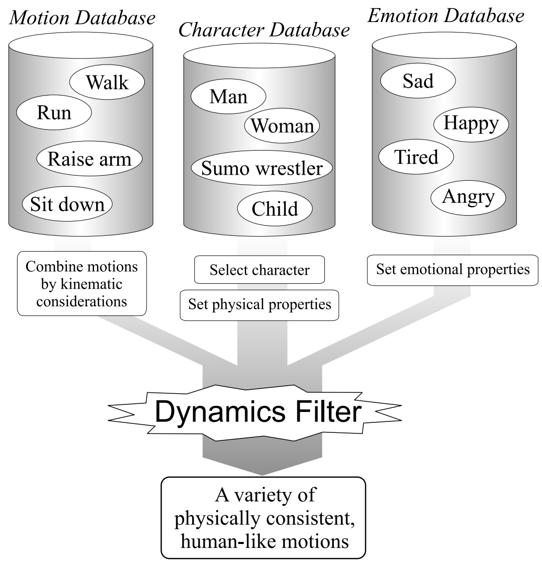

Figure 2: Motion generation via dynamics filter

A dynamics filter is expected to provide a solution for these problems. The procedure for generating motion by dynamics filter is illustrated in Figure 2. First, several properties, such as the motion (walk / run / sit, ...), the model (mass / link length, ...), character (male / female, adult / child, ...) and emotion (happy / angry / sad, ...) are selected and combined kinematically. Next, the combined reference motion is input to the dynamics filter, which outputs a physically consistent motion close to the reference. Users may take some trial-and-error experiments with the dynamics filter to meet their taste. In interactive systems, the reference motion may change during the computation according to user inputs.

This approach is reasonable from the viewpoint of learning process of human. We first imitate just the kinematics of a motion watching the others — then adapt the motion to the dynamics of our own body and the environment.

Implementation of the filter may be off-line or on-line. An off-line filter, making use of the whole sequence of the input motion prior to the filtering, will generate motions of high quality and stability. This type of dynamics filter would be good for creating artistic films in computer graphics, or a motion library for humanoid robots. Previous research has already realized this type of dynamics filter [4, 5, 6, 7].

An on-line version of a dynamics filter is more difficult because only limited informations are provided to the filter, but interesting from the viewpoint of interactive motion generation. This feature is essential for humanoid robots moving in dynamically changing environments and human characters in some applications such as games.

We provide an on-line implementation of a dynamics filter in this paper. This is difficult and constitutes a contribution because others only have built off-line filters. The filter is based on the equation of motion of closed and structure-varying kinematic chains developed for dynamics simulation of human figures [1].

3. Dynamics Computation of Human Figures

3.1. Previous Work and Requirements

Since the human figure can be modeled in terms of kinematic chains, we can apply algorithms developed in multibody dynamics and robotics [8, 9, 10, 11] to human figures [12, 13, 14]. However, human figures have quite different properties compared to conventional robot manipulators from the point of view of dynamics computation as listed below:

1. Many degrees of freedom (DOF) — Human figures usally contain many DOF's, even the simplest model has more than 20 DOF.

2. Complicated closed kinematic chains — Human figures often form complicated closed kinematic chains by holding links in the environment, their own body, or other figures, for which most dynamics algorithms require a large computational load.

3. Structural changes — On catch or release of links by hands, human figures change their link structure dynamically during the motion.

4. Collisions and contacts — A human figure frequently collides with the environments, other human figures, or even itself during motion.

5. Under actuation — Since human figures have no fixed link, they are always under actuated, that is, the DOF of motion is larger than the number of actuators. Therefore, we need to consider the physical consistency of motion in generating motions for human figures.

6. Requirement for interactivity — Interactivity is likely to be the most important feature of future applications of human figures. Human figures move in dynamically changing environment interacting with humans, in contrast to conventional manipulators.

Considering these points, we decided to take a new approach toward the dynamics computation of human figures. Our objectives are:

1. To compute the dynamics of complicated kinematic chains efficiently, in real time

2. To handle open and closed kinematic chains seamlessly to enable on-line computation of structure-varying kinematic chains

3. To compute the dynamics of collisions and contacts efficiently

3.2. Closed Kinematic Chains

Dynamics of closed kinematic chains requires consideration of reaction forces in closed loops. Lagrange multipliers are often applied to compute the reaction force [8, 9]. However, the computational cost of Lagrange multipliers is too large to be applied to real-time or interactive simulation.

An alternative approach is to apply the principle of virtual work [15, 16], which is a mostly used to control robots with closed kinematic chains such as a planar five-bar linkage or parallel mechanisms [17]. This method has the advantage of high computational efficiency which enables real-time control of manipulators. Until recently no general algorithm was known for computing the Jacobian matrix essential for applying the principle of virtual work. We have found a solution for this problem [18] and extended this approach to general closed kinematic chains.

The basic equation of motion of human figures is described as:

$$\begin{pmatrix} \boldsymbol{A} & -\boldsymbol{H}_C^T & -\boldsymbol{H}_J^T \\ \boldsymbol{H}_C & \boldsymbol{O} & \boldsymbol{O} \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{\theta}}_G \\ \boldsymbol{\tau}_C \\ \boldsymbol{\tau}_J \end{pmatrix} = \begin{pmatrix} -\boldsymbol{b} \\ -\dot{\boldsymbol{H}}_C \dot{\boldsymbol{\theta}}_G \end{pmatrix} \tag{1}$$

which can be summarized as:

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{u} \tag{2}$$

where

$\boldsymbol{A}$	:	inertial tensor
$\boldsymbol{b}$	:	centrifugal, Coriollis and gravitational forces
$\boldsymbol{\theta}_G$	:	the generalized coordinates
$\boldsymbol{\tau}_C$	:	constraint forces at connected joints
$\boldsymbol{\tau}_J$	:	joint torques
$\boldsymbol{H}_C$	=	$\partial \boldsymbol{\theta}_C / \partial \boldsymbol{\theta}_G$
$\boldsymbol{H}_J$	=	$\partial \boldsymbol{\theta}_J / \partial \boldsymbol{\theta}_G$
$\boldsymbol{\theta}_J$	:	joint angles

and $\boldsymbol{\theta}_C$ is the variable that represents the constraint condition by $\ddot{\boldsymbol{\theta}}_C = \boldsymbol{O}$. If the joint torques $\boldsymbol{\tau}_J$ are known, we can compute the generalized acceleration $\ddot{\boldsymbol{\theta}}_G$ and constraint forces $\boldsymbol{\tau}_C$ by equation (1).

Figure 3 shows an example of a dynamics simulation of a closed kinematic chain, where a human figure is trying to raise its body with its arms. The human figure has 28 DOF, and each rope consists of spherical and rotational joints. The figure holds the ends of ropes by spherical joints, forming a closed kinematic chain. Forward dynamics computation for this 48 DOF system, including the 6 DOF of the base body, takes approximately 32 ms on DEC Alpha 21264 500 MHz processor.

3.3. Structure-Varying Kinematic Chains

Most conventional methods for dynamics computation of kinematic chains assume that the link connectivity of the system does not change during the simulation. However, this is obviously not the

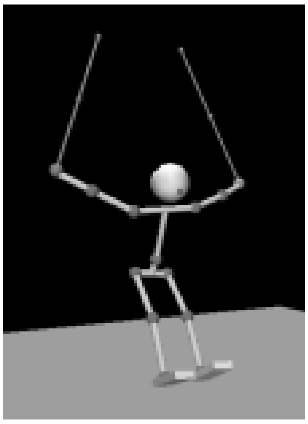

Figure 3: Simulation of a closed kinematic chain

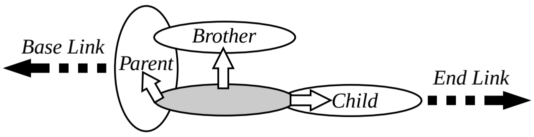

Figure 4: Three pointers to describe link connectivity

case in human motions. In fact, walking, the most common motion of human, can be said to have three different kinematic chains. We call such systems structure-varying kinematic chains and regard this as one of the key issues in the dynamics of human figures. To handle such situations by conventional methods, we would need to prepare all the possible structures in advance and switch among them during the simulation.

To solve this problem, we developed two techniques regarding the description of link connectivity [1]. The first is the method of description itself, where we use three pointers ("parent," "child," and "brother") per link to indicate its neighboring links as illustrated in Figure 4. However, these pointers are not capable of describing closed kinematic chains because the parent-child relationship makes an infinite loop. To describe closed kinematic chains, we virtually cut a joint in a loop and introduce an additional link called virtual link to describe the connection at the virtually cut joint. Another pointer, real, is added to the three pointers, to hold the relationship of real and virtual links. The concept is illustrated in Figure 5.

The second technique is the maintenance of pointers, that is, how to update the connectivity data in response to structural changes. Thanks to

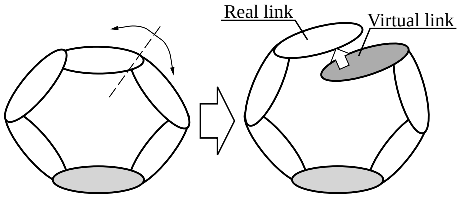

Figure 5: Virtual link to describe closed kinematic chains

the introduction of virtual links, the procedure is quite simple:

1. Link connection — If two links are connected by a new joint, create a new virtual link whose parent is one of the connected links, and set its real pointer to the other.

2. Joint cut — If a joint is cut, delete the virtual link associated with the cut joint.

By these simple procedures, link connectivity data is automatically updated according to the change in the kinematic chain.

The dynamics computation part is programmed to generate the equation of motion automatically based on the link connectivity. Thus, arbitrary structural changes are allowed and simulated without any manual tasks by the user. For the dynamics filter also, this description enables the filter to use the same strategy throughout the motion without concern for the change of structure.

Figure 6 is an example of an interactive application that makes use of these techniques. When the user clicks the mouse button, the monkey releases his hands one by one and finally falls down.

3.4. Collisions and Contacts

Contacts may be viewed as a structural change, because additional constraints appear at the contact point. They differ from joint connection in two points: transition of constraint and unilateral conditions. Therefore, modeling of collisions and contacts is still an open research issue in multibody dynamics area[19].

Collision and contact models are categorized into two types. The visco-elastic body model, or penalty model, places a virtual spring and damper at the point at which contact forces are exerted[20, 21]. The rigid body model, on the other hand, finds the contact force that satisfies

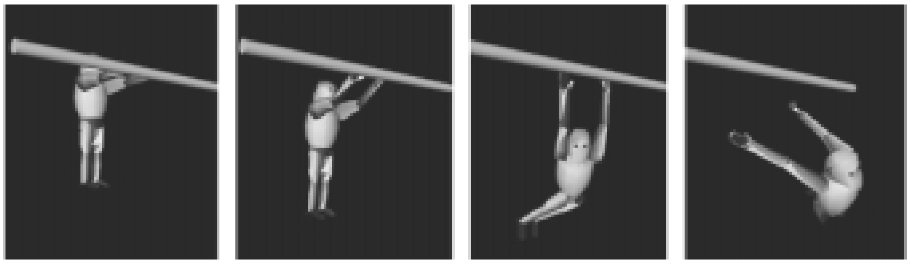

Figure 6: Dynamics simulation of structure-varying kinematic chain

the unilateral conditions through some optimization processes [19].

Our method is based on a rigid body model in the sense that we do not consider the deformation of links. However, to reduce the compuational load, we take the advantage of the fact that the precision of the contact model is not very important in human figures and apply a simplified computation with an iterative procedure:

1. Compute the discontinuous change of joint velocities using conservation of linear and angular momentum and Newton's impact law[19]. This approach has the advantage of numerical stability because large impact forces are not handled.
2. Set full constraint at each contact point.
3. Compute the constraint force to maintain the constraint condition $\ddot{\boldsymbol{\theta}}_C = \boldsymbol{O}$ by equation (1).
4. Check the feasibility of the constraint force.
5. If infeasible constraint forces were found, modify the constraint and return to 3, otherwise accept the constraint.

This method is more efficient than usual rigid body methods for two reasons: (1) it does not exceed four iterations, and (2) the computation for each iteration is very small.

4. Dynamics Filter Implementation

4.1. Basic Equations

The developed filter consists of two parts — feedback control and optimization based on equation (1). The control part computes the desired joint accelerations considering the reference motion, current state, and stability. Then the optimization part computes the optimal solution of equation (1) to generate the joint accelerations close to those computed in the control part.

First, initial desired acceleration of generalized coordinates $\ddot{\boldsymbol{\theta}}_G^{d0}$ is computed by simple joint angle and velocity feedback:

$$\ddot{\boldsymbol{\theta}}_G^{d0} = \ddot{\boldsymbol{\theta}}_G^{ref} + \boldsymbol{K}_D(\dot{\boldsymbol{\theta}}_G^{ref} - \dot{\boldsymbol{\theta}}_G) + \boldsymbol{K}_P(\boldsymbol{\theta}_G^{ref} - \boldsymbol{\theta}_G) \quad (3)$$

where $\boldsymbol{\theta}_G^{ref}$ is the generalized coordinates in reference motion, and $\boldsymbol{K}_D$ and $\boldsymbol{K}_P$ are constant gain matrices.

Then, in order to consider the global stability, the feedback of position and orientation of a specified point $\boldsymbol{P}$ in the upper body are included as follows. The desired acceleration of $\boldsymbol{P}$, $\ddot{\boldsymbol{\theta}}_P^d$, is computed by a similar feedback law as:

$$\ddot{\boldsymbol{r}}_P^d = \ddot{\boldsymbol{r}}_P^{ref} + \boldsymbol{K}_{DP}(\dot{\boldsymbol{r}}_P^{ref} - \dot{\boldsymbol{r}}_P) + \boldsymbol{K}_{PP}(\boldsymbol{r}_P^{ref} - \boldsymbol{r}_P) \quad (4)$$

where $\boldsymbol{r}_P^{ref}$ is the position and orientation of $\boldsymbol{P}$ in the reference motion, which can be obtained by forward kinematics computation, $\boldsymbol{K}_{DP}$ and $\boldsymbol{K}_{PP}$ are constant gain matrices, and $\boldsymbol{r}_P$ is the current position and orientation of $\boldsymbol{P}$. The initial desired acceleration of the generalized coordinates $\ddot{\boldsymbol{\theta}}_G^{d0}$ is modified into $\ddot{\boldsymbol{\theta}}_G^d$, so that the desired acceleration of $\boldsymbol{P}$, $\ddot{\boldsymbol{r}}_P^d$, is realized :

$$\begin{aligned} \ddot{\boldsymbol{\theta}}_G^d &= \ddot{\boldsymbol{\theta}}_G^{d0} + \Delta\ddot{\boldsymbol{\theta}}_G^d && (5) \\ \Delta\ddot{\boldsymbol{\theta}}_G^d &= \boldsymbol{J}_P^\sharp(\ddot{\boldsymbol{r}}_P^d - \ddot{\boldsymbol{r}}_P^{d0}) && (6) \end{aligned}$$

where $\ddot{\boldsymbol{r}}_P^{d0} \stackrel{\triangle}{=} \boldsymbol{J}_P\ddot{\boldsymbol{\theta}}_G^{d0} + \dot{\boldsymbol{J}}_P\dot{\boldsymbol{\theta}}_G$, $\boldsymbol{J}_P \stackrel{\triangle}{=} \partial\boldsymbol{J}_P/\partial\boldsymbol{\theta}_G$, and $\boldsymbol{J}_P^\sharp$ is the weighted pseudo-inverse of $\boldsymbol{J}_P$.

Now we proceed to the optimization part. Solutions of equation (1) represents all the feasible combination of $\ddot{\boldsymbol{\theta}}_G, \boldsymbol{\tau}_C$ and $\boldsymbol{\tau}_J$. The task of the optimization part is to find the optimal solution of equation (1) to realize the desired acceleration. The optimized accelerations are integrated to derive the joint angle data.

First, in preparation for the optimization, we derive the weighted least-square solution of equation (2) and the null space of $\boldsymbol{W}$ regardless of the desired acceleration:

$$\boldsymbol{x} = \boldsymbol{W}^{\sharp}\boldsymbol{u} + (\boldsymbol{E} - \boldsymbol{W}^{\sharp}\boldsymbol{W})\boldsymbol{y} \quad (7)$$

where $\boldsymbol{W}^{\sharp}$ is the pseudo inverse of $\boldsymbol{W}$, $\boldsymbol{y}$ an arbitrary vector, and $\boldsymbol{E}$ the identity matrix of the appropriate size. Picking up the upper rows of equation (7) corresponding to the generalized accelerations, we get:

$$\ddot{\boldsymbol{\theta}}_G = \ddot{\boldsymbol{\theta}}^0_G + \boldsymbol{V}_G\boldsymbol{y} \quad (8)$$

where $\ddot{\boldsymbol{\theta}}^0_G$ is the generalized acceleration in the least-square solution.

Next, we determine the arbitrary vector $\boldsymbol{y}$ to minimize the acceleration error by

$$\boldsymbol{y} = \boldsymbol{V}^*_G(\ddot{\boldsymbol{\theta}}^d_G - \ddot{\boldsymbol{\theta}}_G) \quad (9)$$

where $\boldsymbol{V}^*_G$ is the singularity-robust inverse[22] of $\boldsymbol{V}_G$.

Finally, substituting $\boldsymbol{y}$ into equation (7), we get the optimized solution of $\boldsymbol{x}$. Since $\boldsymbol{x}$ includes the generalized acceleration, joint torques and constraint forces all in one, the optimization part plays three roles at the same time: (1) computation of optimized motion, (2) computation of joint torques to realize the computed acceleration, and (3) dynamics simulation of the result.

4.2. Applications

In the following examples, the additional control point $\boldsymbol{P}$ was taken at the neck and its position and orientation were computed off-line, although it is easy to compute them on-line.

4.2.1. Filtering Raw Motion Capture Data

Figure 7 compares the captured (above) and filtered (below) walking motions. Although small latency is observed in the filtered motion, the result is satisfactory. This method is applicable to any motion as shown in Figure 8, which means that we do not need to prepare different filters for different motions.

4.2.2. Filtering Kinematically Synthesized Motion

The dynamics filter accepts not only raw captured data but also kinematic combinations of captured motions. The human figure in Figure 9 makes a turn of 30 degrees, by simply giving motion obtained by smoothly connecting two walking motions heading different directions as reference, considering no dynamic effects such as centrifugal force. This result shows that we can make a human figure walk along any path by indicating the direction using some input device.

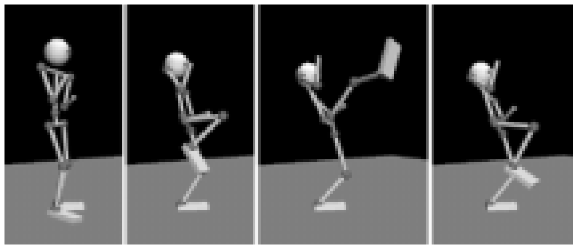

Figure 8: Karate kick generated by the dynamics filter

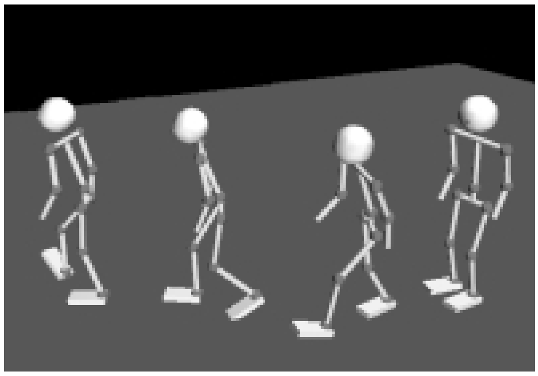

Figure 9: Motion generated from kinematically combined motion

4.2.3. Interactive Motion Generation

Note that the optimization applied here is strictly local to each frame, which means that this filter has the ability to realize a real-time and interactive motion generator. Although we cannot call it a "real-time dynamics filter" because the computation takes 70 to 80 ms per frame with an Alpha 21264 500MHz processor, we tried to interact with the figure filter as in Figure 10, where the figure is controlled to keep standing by the dynamics filter, and pushed in various directions by the user.

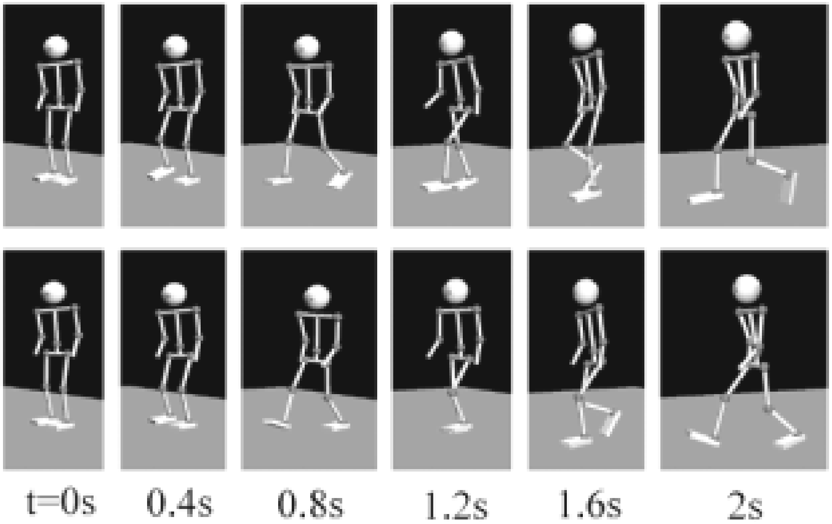

Figure 7: Captured (above) and filtered (below) walking motions

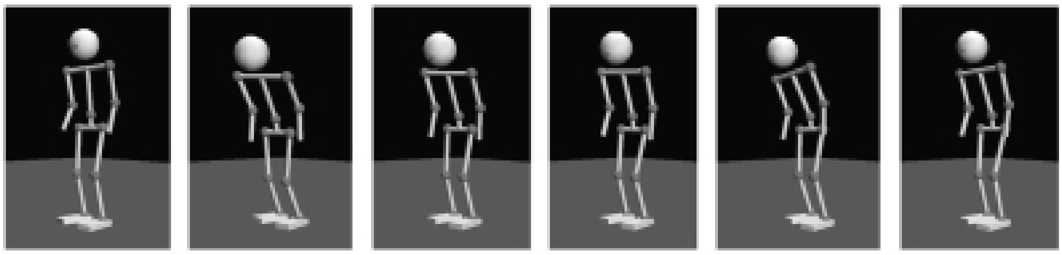

Figure 10: Example of interactive motion generation: push a standing figure

5. Conclusion

This paper presents our research issues toward a motion generator for human figures. The conclusions of this paper are summarized as follows:

1. The concept of a dynamics filter is proposed. A dynamics filter is a motion generator that creates a physically consistent motion from any reference motion, allowing interactive inputs from the user.
2. Basic dynamics computation methods used in the implementation of dynamics filter were presented. The methods can handle various phenomena observed in the motions of human figures, such as:
   (a) general closed kinematic chains;
   (b) structure-varying kinematic chains; and
   (c) collisions and contacts.
3. An implementation of an on-line dynamics filter was also introduced and proved the potential ability of the dynamics filter by examples of motions generated from motion capture data.

Currently we have several projects related to this issue:

1. Improvement of dynamics filter in its ability and computation time toward a real-time interactive system
2. Development of a behavior capture system, which not only captures the motion of humans but also their 3D shape, interaction with the environments, and mental state via various sensors
3. Controlling a real humanoid robot using the behavior capture system as an input device to make the robot imitate the motion of humans

Acknowledgements

This research was supported by the Humanoid Robotics Project, NEDO, Japan, and the CREST Program of the Japan Science and Technology Corporation.

References

[1] K. Yamane and Y. Nakamura: "Dynamics Computation of Structure-Varying Kinematic Chains for Motion Synthesis of Humanoid," Proceedings of IEEE International Conference on Robotics and Automation, pp.714–721, 1999.

[2] J.H. Park and Y.K. Rhee: "ZMP Trajectory Generation for Reduced Trunk Motions of Biped Robots," Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol.1, pp.90–95, 1998.

[3] Q. Huang and S. Kajita et al.: "A High Stability, Smooth Walking Pattern for a Biped Robot," Proceedings of International Conference on Robotics and Automation, pp.65–71, 1999.

[4] A. DasGupta and Y. Nakamura: "Making Feasible Walking Motion of Humanoid Robots from Human Motion Captured Data," Proceedings of International Conference on Robotics and Automation, pp.1044–1049, 1999.

[5] Z. Popovic and A. Witkin: "Physically Based Motion Transformation," Proceedings of SIGGRAPH'99, pp.11–20, 1999.

[6] H. Ko and N.I. Badler: "Animating Human Locomotion with Inverse Dynamics," IEEE Transactions on Computer Graphics, vol.16, no.2, pp.50–59, 1996.

[7] Michiel van de Panne: "From Footprints to Animation," Computer Graphics Forum, vol.16, no.4, pp.211–223, 1997.

[8] E.J. Haug: Computer Aided Kinematics and Dynamics of Mechanical Systems, Allyn and Bacon Series in Engineering, 1989.

[9] J. G. de Jalon and E. Bayo: Kinematic and Dynamic Simulation of Multibody Systems – the Real-Time Challenge, Springer – Verlag, 1993.

[10] R. Featherstone: Robot Dynamics Algorithm, Kluwer, 1987.

[11] K.S. Chang and O. Khatib: "Efficient Algorithm for Extended Operational Space Inertia Matrix," Proceedings of IEEE International Conference on Robotics and Automation, pp.350–355, 1999.

[12] S. McMillan and D.E. Orin: "Forward Dynamics of Multilegged Vehicles Using the Composite Rigid Body Method," Proceedings of IEEE International Conference on Robotics and Automation, pp.464–470, 1998.

[13] O. Bruneau and F.B. Ouezdou: "Dynamic Walk Simulation of Various Bipeds via Ankle Trajectory," Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol.1, pp.58–63, 1998.

[14] B. Perrin, C. Chevallereau, and C. Verdier: "Calculation of the Direct Dynamics Model of Walking Robots: Comparison Between Two Methods," Proceedings of IEEE International Conference on Robotics and Automation, vol.2, pp.1088–1093, 1997.

[15] Y. Nakamura and M. Ghodoussi: "Dynamics Computation of Closed-Link Robot Mechanisms with Nonredundant and Redundant Actuators," IEEE Transactions on Robotics and Automation, vol.5, no.3, pp.294–302, 1989.

[16] A. Codourey and E. Burdet: "A Body-Oriented Method for Finding a Linear Form of the Dynamic Equation of Fully Parallel Robots," Proceedings of IEEE International Conference on Robotics and Automation, pp.1612–1618, 1997.

[17] K. Yamane, M. Okada, N. Komine, and Y. Nakamura: "Parallel Dynamics Computation and H_∞ Acceleration Control of Parallel Manipulators for Acceleration Display," Proceedings of 1998 IEEE International Conference on Robotics and Automation, pp.2301–2308, 1998.

[18] K. Yamane and Y. Nakamura: "Dynamics Computation of Closed Kinematic Chains for Motion Synthesis of Human Figures," Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1108–1114, 1999.

[19] F. Pfeiffer and C. Glocker: Multibody Dynamics with Unilateral Contacts, Wiley Series in Nonlinear Science, 1996.

[20] D.W. Marhefka and D.E. Orin: "Simulation of Contact Using a Nonlinear Damping Model," Proceedings of the 1996 IEEE International Conference on Robotics and Automation, pp.1662–1668, 1996.

[21] P.R. Kraus, V. Kunmar, and P. Dupont: "Analysis of Frictional Contact Models for Dynamic Simulation," Proceedings of the 1998 IEEE International Conference on Robotics and Automation, pp.976–981, 1998.

[22] Y. Nakamura and H. Hanafusa: "Inverse Kinematics Solutions with Singularity Robustness for Robot Manipulator Control," Journal of Dynamic Systems, Measurement, and Control, vol.108, pp.163–171, 1986.