Interactive Motion Generation of Humanoid Robots via Dynamics Filter

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 describes a new method for generating motions of humanoid robots interactively using "dynamics filter," a concept of motion generator for human figures that converts a physically inconsistent motion into consistent one. Interactivity is a key issue of many applications of humanoid robots working in changing environment with human. Since our implementation of dynamics filter only uses time-local information, we can change the reference motion by kinematically combining several motions in response to the interactions with human and environment. We also provide the outline of the dynamics computation methods on which the dynamics filter is based. The methods are also useful for real-time dynamics simulation of human figures including structure-varying open and closed kinematic chains and contacts.

1 Introduction

Interactivity is a key issue in many applications of humanoid robots working in changing and/or unknown environment with human. A robot may need to avoid an obstacle suddenly appearing in front of it, resist against external forces applied to its body, or abruptly change the walking direction as ordered by its master. Most of the proposed motion generation methods assume off-line computation, or accept only small disturbance from the environment, providing poor interactivity in the generated motion. On the other hand, the interaction between a robot and human is usually discussed on the behavior level and physical feasibility of the motion is seldom considered.

The goal of this research is to bridge the gap between these two approaches: develop a motion generator that generates physically consistent motions when any desired behaviors are given through interactions between a robot and the environment or human. The generator has to be on-line, computationally efficient, and applicable to wide range of behaviors.

Motion generation considering dynamics of human figures has been discussed in both humanoid robots and computer animation fields. One of the major approaches is to describe the motion by a few parameters and optimize them using zero moment point (ZMP) [1–4], inverted pendulum mode[5], learning[6], and so on. In any cases, the optimization is done off-line due to the heavy computational load, allowing robots to make only predefined motions. Moreover, parameterization of a motion tends to yield an artificial, unfriendly motion. Some on-line controllers are also proposed[7–9], but they only allow small differences in the environment.

Another approach is to use intelligent control schemes to create various behaviors[10–13]. A problem of this approach is that we need to prepare different controller for each behavior, which leads to difficulty in generating intermediate behaviors. Some other methods make use of motion capture data to generate human-like motions[14, 15], where human models are simplified to cope with the complicated human motion. Therefore, physical consistency of the generated motion in the full human model is not guaranteed.

In this paper, we propose the concept of a *dynamics filter*, a real-time and interactive motion generator for human figures. The basic function of the dynamics filter is to convert a physically inconsistent motion into a consistent one. Any motion data may be input to the filter as a reference motion — even captured from human, created by a professional animator, and so on. Dynamics filter will provide large flexibility in applying the existing motion data to different model or environment. Of course some motion generation methods can be called a dynamics filter[3, 14, 15], but our approach is to develop an on-line dynamics filter which does not require the whole sequence of the reference motion beforehand so that we can change it interactively.

This paper is organized as follows. In section 2 we introduce the basic concept of dynamics filter. Section 3 provides the equation of motion of constrained human figures on which our implementation of dynamics filter is based, followed by the details on the implementation in section 4. We also show some examples of applying the dynamics filter to various situations. In the last section, we introduce our ongoing projects along with the conclusions of this paper.

2 The Concept of Dynamics Filter

Most of the existing methods for generating motions of human figures have problems from the standpoint of interactivity due to their poor flexibility, off-line computation scheme, long computation time, and so on. A dynamics filter is expected to provide a solution for these problems. The procedure for generating motion by dynamics filter is illustrated in Figure 1. 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. We can adopt many techniques for kinematic synthesis of motions [16–19] at this stage. 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.

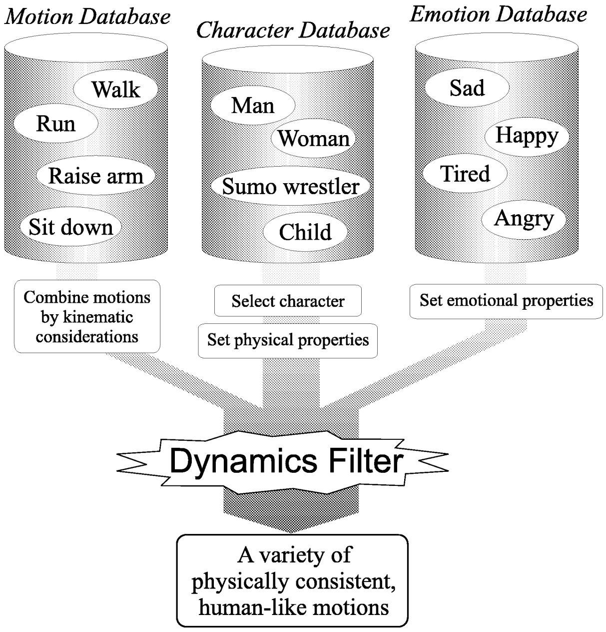

Fig. 1. Motion generation via dynamics filter

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 [3, 14, 15]. An on-line version of a dynamics filter is obviously more difficult because only limited informations are provided to the filter, but essential for interactive motion generation.

It is also worth pointing out that the dynamics filter approach is reasonable from the viewpoint of learning process of human in the following sense: 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. Applying off-line dynamics filter may correspond to practicing a difficult task many times. Simple tasks, on the other hand, may be achieved by a single trial, which in turn may be achieved by an on-line dynamics filter.

For interactive motion generation, we develop an on-line implementation of a dynamics filter. 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 [20].

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 [21–24] to human figures [25–27]. 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

The following subsections describe our methods for computing the dynamics of phenomena observed in the motions of human figures [20, 28].

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 [21, 22]. 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 [29, 30], which is a mostly used to control robots with closed kinematic chains such as a planar five-bar linkage or parallel mechanisms [31]. 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 [28] 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 2 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.

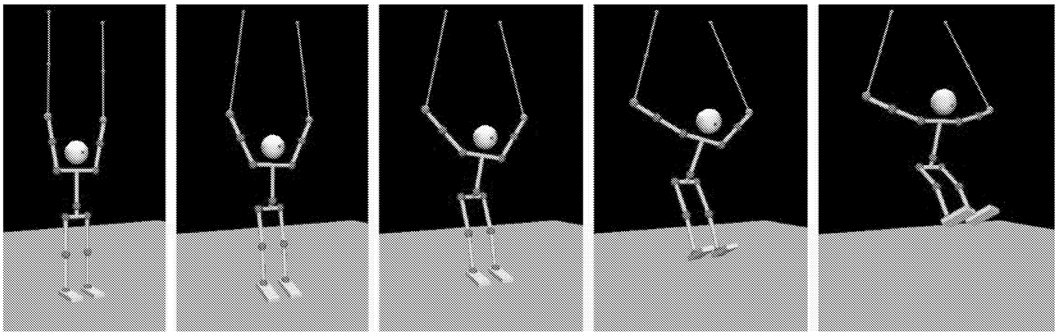

Fig. 2. Simulation of a closed kinematic chain

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 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 [20]. 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 3. 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 4.

The second technique is the maintenance of pointers, that is, how to update the connectivity data in response to structural changes. Thanks to 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 as illustrated in Figure 5.

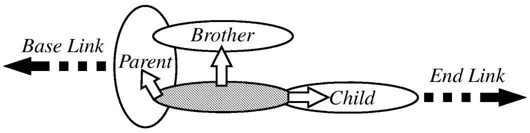

Fig. 3. Three pointers to describe link connectivity

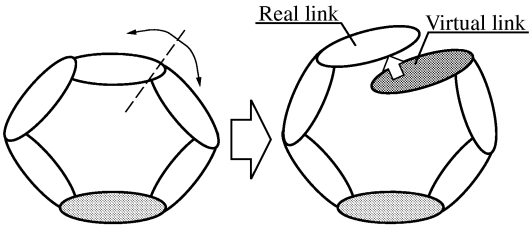

Fig. 4. Virtual link to describe closed kinematic chains

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.

Each human figure is described as an independent open chain at the initial state. If a link in the figure is connected to another one, a virtual link is generated at the connection point following the procedure described above. Therefore, we do not need to reconstruct or re-order the link connectivity of human figures when link connections or joint cuts occur.

The dynamics computation part is programmed to generate the equation of motion automatically based on the link connectivity data. Thus, arbitrary structural changes are allowed and simulated without any manual tasks by the user. This description also enables a dynamics 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[32].

Collision and contact models are categorized into two types. The visco-elastic body model, or penalty model, places a virtual spring and damper between the links in contact, which exert the contact force depending on the virtual deformation of links [33, 34]. The rigid body model, on the other hand, finds the contact force that satisfies the unilateral conditions through some optimization processes [32]. These methods focus on a precise analysis of collisions and contacts for applications where the effects of these phenomena are critical to the performance of the task such as manipulation by fingers. In human figures,

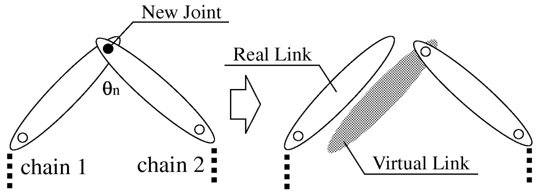

Fig. 5. Creating a virtual link on link connection

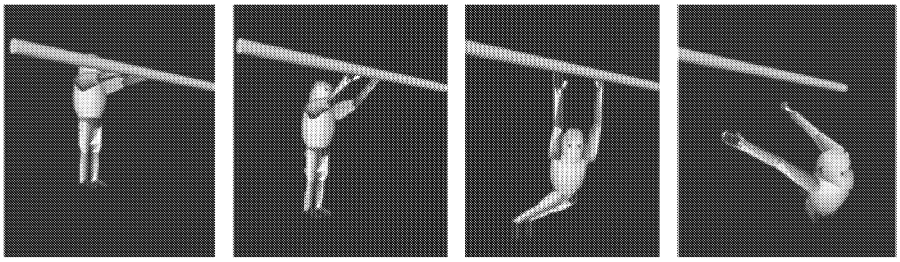

Fig. 6. Dynamics simulation of structure-varying kinematic chain

however, the precision of the contact model is not very important for the motion, because most contacts can be considered as stick contacts and bouncing is rarely observed.

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[32]. This approach has the advantage of numerical stability because large impact forces are omitted.
2. Set full constraint at each contact pair.
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 Outline of the Dynamics Filter

The developed filter consists of two parts — feedback control and optimization based on equation (1). The role of the control part is to compute the desired (but not always feasible) joint accelerations considering the reference motion, current state, and stability. This part itself consists of two feedback sections: local and global. The local feedback section simply computes the initial desired accelerations by local feedback at each joint, which are modified by the global feedback section to improve the stability of the whole body. Given the desired joint accelerations, the optimization part then computes the optimal solution of equation (1) to generate the joint accelerations close to those computed in the control part.

Both parts are designed to require only the information of current state and reference motion. Thanks to this feature, we can modify the reference motion interactively in the middle of filtering.

4.2 Details

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

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

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:

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

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 :

$$\ddot{\boldsymbol{\theta}}_G^{d} = \ddot{\boldsymbol{\theta}}_G^{d0} + \Delta\ddot{\boldsymbol{\theta}}_G^{d} \tag{5}$$

$$\Delta\ddot{\boldsymbol{\theta}}_G^{d} = \boldsymbol{J}_P^{\sharp}(\ddot{\boldsymbol{r}}_P^{d} - \ddot{\boldsymbol{r}}_P^{d0}) \tag{6}$$

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

Optimization 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:

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

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:

$$\ddot{\boldsymbol{\theta}}_G = \ddot{\boldsymbol{\theta}}_G^{0} + \boldsymbol{V}_G\boldsymbol{y} \tag{8}$$

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

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

$$\boldsymbol{y} = \boldsymbol{V}_G^{*}(\ddot{\boldsymbol{\theta}}_G^{d} - \ddot{\boldsymbol{\theta}}_G^{0}) \tag{9}$$

where $\boldsymbol{V}_G^{*}$ is the singularity-robust inverse[35] 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.

4.3 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.

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.

Filtering into a Different Environment We also tried to apply the same walking motion as in Figure 7 to a different environment, in this case a slope, and see what happens. The result is shown in Figure 9. No modification on captured data was made except for the position of the body, which was modified to maintain the same height from the ground.

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 10 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.

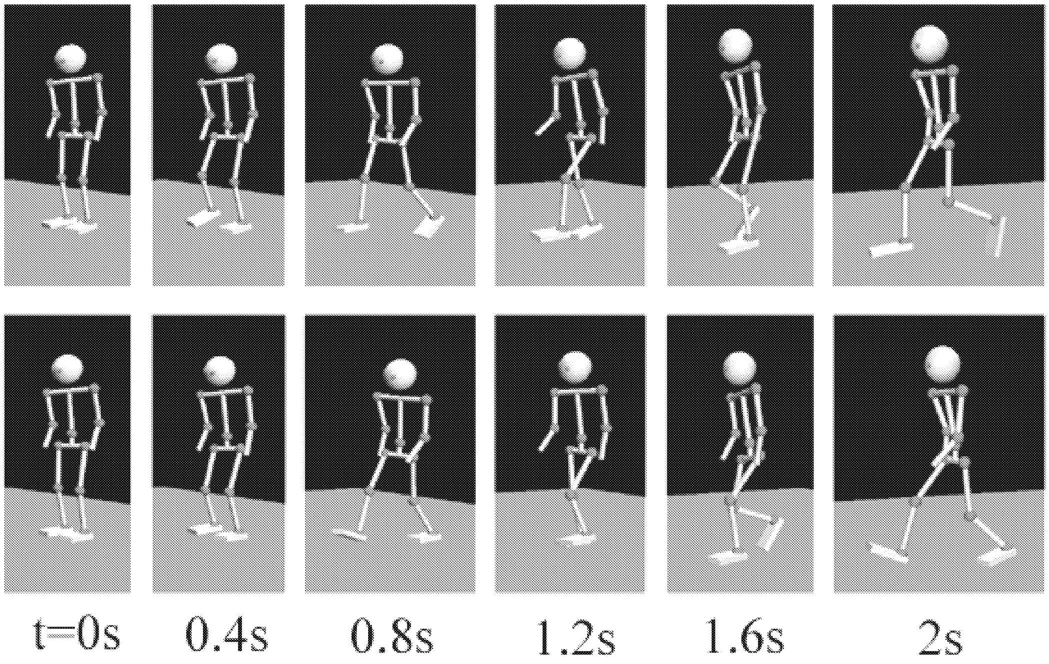

Fig. 7. Captured (above) and filtered (below) walking motions

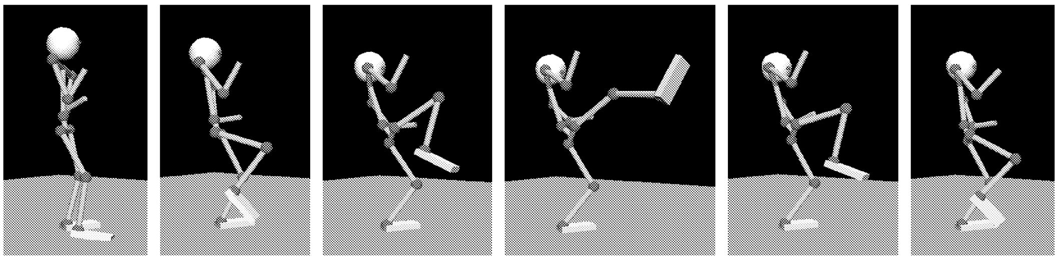

Fig. 8. Karate kick generated by the dynamics filter

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 11, where the figure is controlled to keep standing by the dynamics filter, and pushed in various directions by the user.

4.4 Discussion

The advantages of the dynamics filter are summarized as follows:

- the reference motion may change during filtering; therefore contains high interactivity
- applicable to any motions
- relatively short computation time (although far from real time)

On the other hand, we encountered several difficulties in using the dynamics filter. The main problem is that it is very difficult to tune the parameters (feedback gains, weights for weighted pseudo inverses) and we need to find a set of parameters for each behavior, yeilding the same problem as in behavior-specified controllers. The encouraging fact is, however, it is rather easier to interpolate the parameters than to interpolate the controllers to get intermediate behaviors.

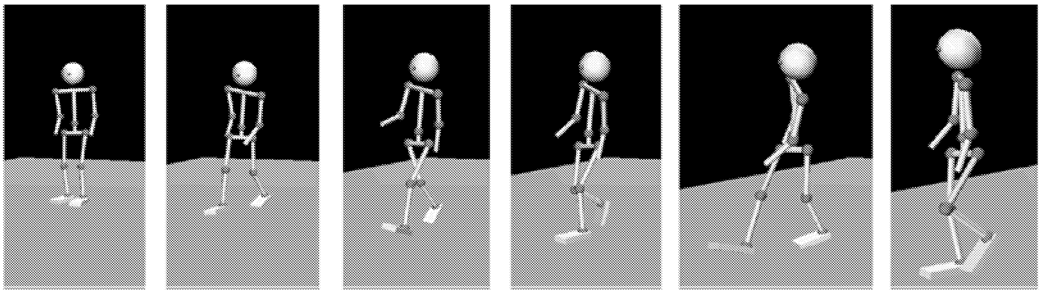

Fig. 9. Walk on a slope

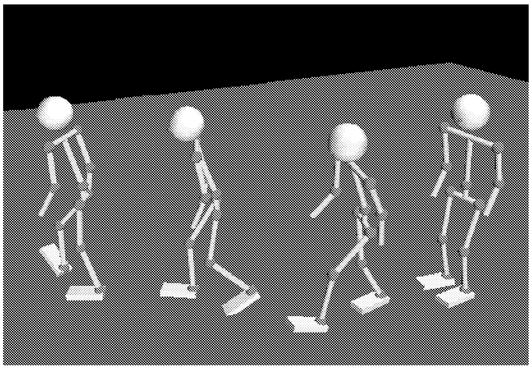

Fig. 10. Motion generated from kinematically combined motion

5 Conclusion and Future Works

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 in interactive motion generation by examples of motions generated from motion capture data.

Currently we have several projects related to this issue listed below:

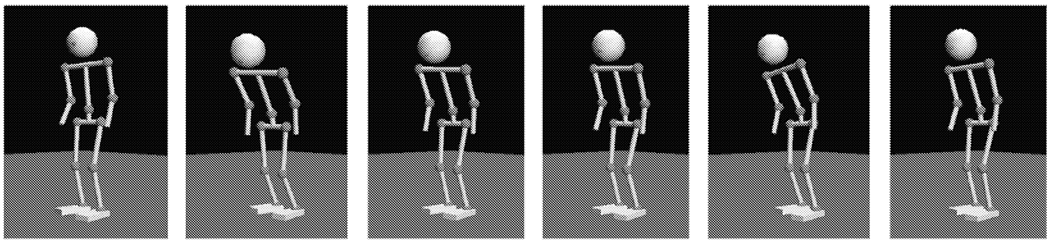

Fig. 11. Example of interactive motion generation: push a standing figure

1. Development and implementation of a parallel dynamics computation algorithm to realize real-time dynamics simulation and motion generation — Recently efficient parallel computation algorithms for the dynamics of general kinematic chains have been proposed [36, 37] and may be a solution for real-time computation.
2. Improvement of dynamics filter in its ability and computation time toward a real-time interactive system — A solution may be to implement several low-level heuristic schemes for stabilization. Also, in order to control a robot using the dynamics filter, we need to consider some more properties of the robot such as limits in joint angles, joint torques, and so on.
3. 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 — The system is expected to provide a model for human brain with perception as input and motion as output. Another interesting application would be a master-slave system for controlling humanoid robots with almost full compatibility between the master and slave parts.

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. J. Yamaguchi, A. Takanishi, and I. Kato. Development of a biped walking robot compensating for three-axis moment by trunk motion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*, pages 561–566, 1993.
2. J.H. Park and Y.K. Rhee. ZMP Trajectory Generation for Reduced Trunk Motions of Biped Robots. In *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 90–95, 1998.
3. A. DasGupta and Y. Nakamura. Making Feasible Walking Motion of Humanoid Robots from Human Motion Captured Data. In *Proceedings of International Conference on Robotics and Automation*, pages 1044–1049, 1999.
4. Q. Huang and S. et al. Kajita. A High Stability, Smooth Walking Pattern for a Biped Robot. In *Proceedings of International Conference on Robotics and Automation*, pages 65–71, 1999.
5. J.H. Park and K.D. Kim. Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3528–3533, Leuven, Belgium, May 1998.
6. C.M. Chew and G.A. Pratt. A general control architecture for dynamic bipedal walking. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3990–3996, San Francisco, CA, April 2000.
7. J. Hu, J. Pratt, and G. Pratt. Adaptive Dynamic Control of a Biped Walking Robot with Radial Basis Function Neural Networks. In *Proceedings of International Conference on Robotics and Automation*, pages 400–405, 1998.
8. Y. Fujimoto, S. Obata, and A. Kawamura. Robust Biped Walking with Active Interaction Control between Foot and Ground. In *Proceedings of International Conference on Robotics and Automation*, pages 2030–2035, 1998.
9. Q. Huang, K. Kaneko, K. Yokoi, S. Kajita, and T. Kotoku. Balance Control of a Biped Robot Combining Off-line Pattern with Real-time Modification. In *Proceedings of International Conference on Robotics and Automation*, pages 3346–3352, 2000.
10. M. van de Panne. Towards Agile Animated Characters. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 682–687, San Francisco, CA, April 2000.
11. W.L. Wooten and J.K. Hodgins. Animation of Human Diving. *Computer Graphics Forum*, 15(1):3–13, 1996.
12. D.C. Brogan, R.A. Metoyer, and J.K. Hodgins. Dynamically Simulated Characters in Virtual Environments. *IEEE Computer Graphics and Applications*, 18(5):58–69, 1998.
13. W.L. Wooten and J.K. Hodgins. Simulating Leaping, Tumbling, Landing, and Balancing Humans. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 656–662, San Francisco, CA, April 2000.
14. Z. Popovic. Editing Dynamic Properties of Captured Human Motion. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 670–675, San Francisco, CA, April 2000.
15. N.S. Pollard and F. Behmaram-Mosavat. Force-Based Motion Editing for Locomotion Tasks. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 663–669, San Francisco, CA, April 2000.
16. Michael Gleicher. Retargetting Motion to New Characters. In *Proceedings of ACM SIGGRAPH '98*, 1998.

17. J. Lee and S.Y. Shin. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of SIGGRAPH'99*, pages 39–48, 1999.
18. C. Rose, M.F. Cohen, and B. Bodenheimer. Verbs and Adverbs: Multidimentional Motion Interpolation. *IEEE Computer Graphics and Applications*, 18(5):32–40, 1998.
19. C. Rose, B. Guenter, B. Bodenheimer, and M.F. Cohen. Efficient Generation of Motion Transitions using Spacetime Constraints. In *Computer Graphics Proceedings, Annual Conference Series*, pages 147–154, 1996.
20. K. Yamane and Y. Nakamura. Dynamics Computation of Structure-Varying Kinematic Chains for Motion Synthesis of Humanoid. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 714–721, 1999.
21. E.J. Haug. *Computer Aided Kinematics and Dynamics of Mechanical Systems.* Allyn and Bacon Series in Engineering, 1989.
22. J. G. de Jalon and E. Bayo. *Kinematic and Dynamic Simulation of Multibody Systems – the Real-Time Challenge.* Springer – Verlag, 1993.
23. R. Featherstone. *Robot Dynamics Algorithm.* Kluwer Academic Publishers, Boston, MA, 1987.
24. K.S. Chang and O. Khatib. Efficient Algorithm for Extended Operational Space Inertia Matrix. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 350–355, 1999.
25. S. McMillan and D.E. Orin. Forward Dynamics of Multilegged Vehicles Using the Composite Rigid Body Method. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 464–470, 1998.
26. O. Bruneau and F.B. Ouezdou. Dynamic Walk Simulation of Various Bipeds via Ankle Trajectory. In *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 58–63, 1998.
27. B. Perrin, C. Chevallereau, and C. Verdier. Calculation of the Direct Dynamics Model of Walking Robots: Comparison Between Two Methods. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 1088–1093, 1997.
28. K. Yamane and Y. Nakamura. Dynamics Computation of Closed Kinematic Chains for Motion Synthesis of Human Figures. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1108–1114, 1999.
29. Y. Nakamura and M. Ghodoussi. Dynamics Computation of Closed-Link Robot Mechanisms with Nonredundant and Redundant Actuators. *IEEE Transactions on Robotics and Automation*, 5(3):294–302, 1989.
30. A. Codourey and E. Burdet. A Body-Oriented Method for Finding a Linear Form of the Dynamic Equation of Fully Parallel Robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1612–1618, 1997.
31. K. Yamane, M. Okada, N. Komine, and Y. Nakamura. Parallel Dynamics Computation and H_∞ Acceleration Control of Parallel Manipulators for Acceleration Display. In *Proceedings of 1998 IEEE International Conference on Robotics and Automation*, pages 2301–2308, 1998.
32. F. Pfeiffer and C. Glocker. *Multibody Dynamics with Unilateral Contacts.* Wiley Series in Nonlinear Science, 1996.
33. D.W. Marhefka and D.E. Orin. Simulation of Contact Using a Nonlinear Damping Model. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 1662–1668, 1996.
34. P.R. Kraus, V. Kunmar, and P. Dupont. Analysis of Frictional Contact Models for Dynamic Simulation. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 976–981, 1998.
35. Y. Nakamura and H. Hanafusa. Inverse Kinematics Solutions with Singularity Robustness for Robot Manipulator Control. *Journal of Dynamic Systems, Measurement, and Control*, 108:163–171, 1986.
36. R. Featherstone. A Divide-and-Conquer Articulated-Body Algorithm for Parallel $O(\log(n))$ Calculation of Rigid-Body Dynamics. Part1: Basic Algorithm. *International Journal of Robotics Research*, 18(9):867–875, September 1999.
37. R. Featherstone. A Divide-and-Conquer Articulated-Body Algorithm for Parallel $O(\log(n))$ Calculation of Rigid-Body Dynamics. Part2: Trees, Loops, and Accuracy. *International Journal of Robotics Research*, 18(9):876–892, September 1999.