Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, CA • April 2000

Dynamics Filter — Concept and Implementation of On-Line Motion Generator for Human Figures

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

Dept. of Mechano-Informatics, Univ. of Tokyo
7-3-1 Hongo Bunkyo-ku Tokyo, 113-8656 JAPAN

Abstract

Humanoid robots are required to make a variety of dynamic and even expressive motions in changing environments. However, the conventional methods for generating humanoid motions fail to achieve this requirement since they can only generate quite artificial and predefined motions through rather complicated optimization processes. In this paper, we propose the concept of "dynamics filer" which transforms a physically inconsistent motion into a consistent one, and provide an example of its implementation using feedback control and local optimization. The optimization is based on the equation of motion of constrained kinematic chains, which is derived from our previously proposed method for computing the dynamics of structure-varying kinematic chains. The proposed method can be applied to on-line motion generator of humanoid robots.

Key Words: Motion Generation, Human Figures, Physical Consistency, Dynamics, Motion Synthesis.

1 Introduction

Generating motions of human figures, including humanoid robots and human characters in computer animations, is of great interest in both robotics and computer graphics (CG) fields. However, since human figures have completely different structures from conventional robotic mechanisms, it is difficult to generate satisfying motions.

Considering the future applications of humanoid robots, their motion generators are desired to satisfy the following requirements:

1. Feasibility — Above all, a generated motion should be physically feasible for the robot, otherwise it would be extremely difficult to achieve stable motion. We call this condition "physical consistency," the minimum requirement for motions of human figures.

2. Flexibility — Humanoid robot, expected to work in complex and dynamically changing environment, are required to modify their motions according to the environment, task, and so on, in real time. Therefore motion generators should have the ability to create wide variety of motions with high computational efficiency.

3. Friendliness — May be an optional condition, but still important. If humanoid robots are used in public, their motions are desired to be natural and smooth from the point of view of human friendliness, contrary to the well-known phrase, "motion like a robot." Sometimes they may be even required to imitate human motion.

Although many motion generators has been proposed for humanoid and biped robots, very few of them satisfy these conditions. Most generators focus only on physical consistency and tries to reduce the number of optimization parameters by applying simple polynomial or other functions[1, 2, 3, 4]. Moreover, in spite of their efforts, they usually take long time to generate a single sequence of motion, and also limits their application to walking, which reduce the flexibility in changing motions. DasGupta et al.[5] proposed a method of generating feasible motion from human motion captured data, but their method is only applicable to walking, or at least cyclic motions. It seems that generating human-like motion is more extensively studied in computer animation fields[6, 7, 8, 9].

In this paper, we propose the concept of "dynamics filter", that modifies a physically inconsistent motion into a consistent one. Section 2 describes the details of dynamic filter. We also provide a method of realizing dynamics filter based on the equation of motion of

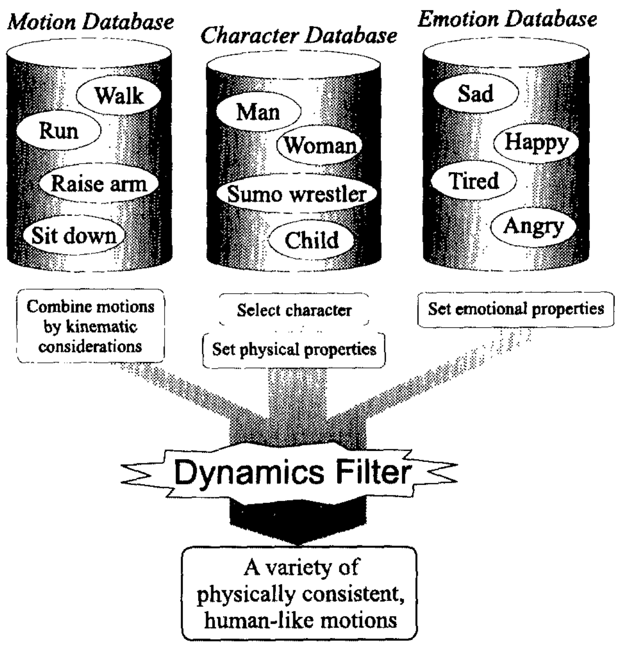


Fig.1: Motion generation using dynamics filter

constrained system described in section 3. Solutions of the equation are optimized to realize a motion as close as possible to the reference through feedback controller and local optimization process presented in section 4. In section 5, several examples of motions generated by putting motion captured data are shown, followed by the conclusions.

2 Concept of Dynamics Filter

"Dynamics filter" is a computational engine that takes a sequence of motion as input, and outputs another sequence which not only preserves the characteristics of the original but also satisfies physical consistency. The input motion may be motion captured, hand-drawn, numerically generated pattern, or result of some kinematic processes of them, which may be physically inconsistent at this stage. Dynamics filter, considering dynamics and other features such as joint limits, environment, collision, and so on, converts the input motion into a physically feasible one which may be realized by the robot with quite simple controller. Motion generation process using dynamics filter would be illustrated as shown in Fig.1. The kinematic synthesizer generates the reference motion by combining a couple of motions from motion database, by which we can reuse any motion data in our hand to generate a new motion without worrying about feasibility. We may apply the methods for kinematic synthesis of motions developed in CG[10, 11]. Physical inconsistency

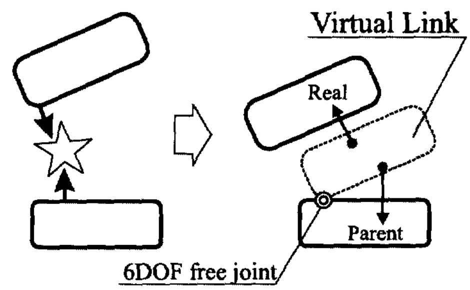


Fig.2: Virtual link generated upon connection

in the output motion of kinematic synthesizer will be corrected by the dynamics filter.

There may be a number of methods for realizing the function of dynamics filter – we present one of them in this paper. The method proposed here is based on the equation of motion of human figures with constraints between the environment, which is derived from our previous research on the dynamics computation of structure-varying kinematic chains[12]. We first calculate the desired joint accelerations by a feedback controller to make the motion close to the reference. Then, among the solutions of the equation, we find the optimal set of acceleration, contact force and joint torque. Since the optimization process is strictly local to each frame, the filter does not need to know the whole reference motion in advance, which means that we can change the reference interactively during the motion.

3 Equation of Motion of Constrained Human Figures

3.1 Description of Constraints via Virtual Links

When a new kinematic constraint appears between a link in a human figure and another link either in itself, another figure, or environment, it is described by a *virtual link* with one of the connected links being its real link and the other the parent[12]. This description allows us to handle any link connections and joint cuts seamlessly using the minimum number of generalized coordinates to achieve computational efficiency.

The virtual link is connected to its parent through a 6 degrees of freedom (DOF) joint as illustrated in Fig.2. Compared to the method in [12], we need to specify the constrained axes, which was implied by the selection of joint type of the virtual link in the previous method.

3.2 Dynamics and Kinematics Equations

Let N_{all} be the number of all joints, and $\boldsymbol{\theta}_{all} \in \boldsymbol{R}^{N_{all}}$ their joint values, including the 6 DOF of base link and virtual links. Since a human figure with kinematic constraints forms a closed kinematic chain, the DOF of its motion N_G is less than N_{all}. N_G is computed by subtracting the number of holonomic constraints from N_{all}. The generalized coordinates of the system $\boldsymbol{\theta}_G \in \boldsymbol{R}^{N_G}$ is formed by selecting appropriate N_G elements from $\boldsymbol{\theta}_{all}$. Typically $\boldsymbol{\theta}_G$ includes position and orientation of the base link and all joint angles in the body. The equation of motion of the human figure is described as:

$$\boldsymbol{\tau}_G = \boldsymbol{A}\ddot{\boldsymbol{\theta}}_G + \boldsymbol{b} \tag{1}$$

where $\boldsymbol{\tau}_G \in \boldsymbol{R}^{N_G}$ is the generalized force acting on the chain, $\boldsymbol{A} \in \boldsymbol{R}^{N_G \times N_G}$ and $\boldsymbol{b} \in \boldsymbol{R}^{N_G}$ are the mass matrix and nonlinear terms, respectively. Velocities and accelerations of $\boldsymbol{\theta}_{all}$ and $\boldsymbol{\theta}_G$ are related by

$$\begin{aligned} \dot{\boldsymbol{\theta}}_{all} &= \boldsymbol{H}\dot{\boldsymbol{\theta}}_G & (2) \\ \ddot{\boldsymbol{\theta}}_{all} &= \boldsymbol{H}\ddot{\boldsymbol{\theta}}_G + \dot{\boldsymbol{H}}\dot{\boldsymbol{\theta}}_G & (3) \end{aligned}$$

where $\boldsymbol{H} \in \boldsymbol{R}^{N_{all} \times N_G}$ is the Jacobian matrix of $\boldsymbol{\theta}_{all}$ with respect to $\boldsymbol{\theta}_G$:

$$\boldsymbol{H} \triangleq \frac{\partial \boldsymbol{\theta}_{all}}{\partial \boldsymbol{\theta}_G}. \tag{4}$$

Using $\boldsymbol{H}$ the relationship of forces $\boldsymbol{\tau}_G$ and $\boldsymbol{\tau}_{all}$ corresponding to $\boldsymbol{\theta}_G$ and $\boldsymbol{\theta}_{all}$, respectively, is described as:

$$\boldsymbol{\tau}_G = \boldsymbol{H}^T \boldsymbol{\tau}_{all}. \tag{5}$$

Refer to [12] for the methods for computing $\boldsymbol{H}$ and $\dot{\boldsymbol{H}}\dot{\boldsymbol{\theta}}_G$.

3.3 Dynamics Equation of Constrained Motion

Suppose N_C axes of virtual links are constrained in total, and let $\boldsymbol{\theta}_C \in \boldsymbol{R}^{N_C}$ be their joint values. The relationship between $\boldsymbol{\theta}_C$ and the generalized coordinates $\boldsymbol{\theta}_G$ is expressed as:

$$\begin{aligned} \dot{\boldsymbol{\theta}}_C &= \boldsymbol{H}_C\dot{\boldsymbol{\theta}}_G & (6) \\ \ddot{\boldsymbol{\theta}}_C &= \boldsymbol{H}_C\ddot{\boldsymbol{\theta}}_G + \dot{\boldsymbol{H}}_C\dot{\boldsymbol{\theta}}_G & (7) \end{aligned}$$

where $\boldsymbol{H}_C \in \boldsymbol{R}^{N_C \times N_G}$ is the Jacobian matrix defined by

$$\boldsymbol{H}_C \triangleq \frac{\partial \boldsymbol{\theta}_C}{\partial \boldsymbol{\theta}_G}. \tag{8}$$

Since $\boldsymbol{\theta}_C$ is a part of $\boldsymbol{\theta}_{all}$, $\boldsymbol{H}_C$ is formed by selecting appropriate rows from $\boldsymbol{H}$.

The constraints are usually described as:

$$\ddot{\boldsymbol{\theta}}_C = \boldsymbol{O}. \tag{9}$$

Suppose constraint forces $\boldsymbol{\tau}_C \in \boldsymbol{R}^{N_C}$ are required to maintain the constriants and the N_J actuated joints are generating joint torques $\boldsymbol{\tau}_J \in \boldsymbol{R}^{N_J}$. From Eq.(5), the generalized force $\boldsymbol{\tau}_G$ is computed by

$$\boldsymbol{\tau}_G = \boldsymbol{H}_C^T \boldsymbol{\tau}_C + \boldsymbol{H}_J^T \boldsymbol{\tau}_J \tag{10}$$

where $\boldsymbol{H}_J \in \boldsymbol{R}^{N_J \times N_G}$ is the Jacobian matrix of the actuated joints $\boldsymbol{\theta}_J \in \boldsymbol{R}^{N_J}$ with respect to the generalized coordinates $\boldsymbol{\theta}_G$, defined by:

$$\boldsymbol{H}_J \triangleq \frac{\partial \boldsymbol{\theta}_J}{\partial \boldsymbol{\theta}_G}. \tag{11}$$

Since $\boldsymbol{\theta}_J$ is also a part of $\boldsymbol{\theta}_{all}$, $\boldsymbol{H}_J$ is formed by selecting appropriate rows from $\boldsymbol{H}$.

Solving Eqs.(1)(7)(9)(10) in terms of $\ddot{\boldsymbol{\theta}}_G, \boldsymbol{\tau}_C$ and $\boldsymbol{\tau}_J$, we get the following linear equation:

$$\begin{pmatrix} \boldsymbol{A} & -\boldsymbol{H}_C^T & -\boldsymbol{H}_J^T \\ \boldsymbol{H}_C & \boldsymbol{O} & \boldsymbol{O} \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{\theta}}_G \\ \boldsymbol{\tau}_C \\ \boldsymbol{\tau}_J \end{pmatrix} = \begin{pmatrix} -\boldsymbol{b} \\ -\dot{\boldsymbol{H}}_C\dot{\boldsymbol{\theta}}_G \end{pmatrix} \tag{12}$$

which can be simplified as:

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{u}. \tag{13}$$

This is the general equation of motion for kinematic chains with additional constraints between the environment.

Note that, in Eq.(12), if the matrix in left hand side has full row rank, the solution forms N_J-dimensional space, which implies that giving appropriate N_J elements of the vector $(\ddot{\boldsymbol{\theta}}_G^T \boldsymbol{\tau}_C^T \boldsymbol{\tau}_J^T)^T$ determines the motion uniquely. The application of the equation may differ depending on which N_J elements to give, as discussed in the next subsection.

3.4 Applications of Equation (12)

Dynamics Simulation If the N_J joint torques $\boldsymbol{\tau}_J$ were given, Eq.(12) can be written as:

$$\begin{pmatrix} \boldsymbol{A} & -\boldsymbol{H}_C^T \\ \boldsymbol{H}_C & \boldsymbol{O} \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{\theta}}_G \\ \boldsymbol{\tau}_C \end{pmatrix} = \begin{pmatrix} \boldsymbol{H}_J^T \boldsymbol{\tau}_J - \boldsymbol{b} \\ -\dot{\boldsymbol{H}}_C\dot{\boldsymbol{\theta}}_C \end{pmatrix} \tag{14}$$

by which the generalized accelerations $\ddot{\boldsymbol{\theta}}_G$ and contact forces $\boldsymbol{\tau}_C$ are computed. This is nothing but dynamics simulation.

We have already proposed a method for the dynamics simulation of structure-varying kinematic chains[12], in which new constraints are imposed by reducing the DOF of new joints, instead of setting explicit constraints in 6 DOF joints as described in this paper. The advantages of the method proposed here are:

- Constraint forces τ_C are directly computed through the simulation. This is convenient for some situations such as catching high bar by hands, where too large constraint force may cause cutting connections.
- Since virtual links introduce no kinematic constraints at all, the DOF of the whole system N_G does not change throughout the simulation. Therefore we do not need to recompute the DOF or re-select the generalized coordinates.
- We can set constraints to arbitrary axes of the 6 DOF's of virtual links, which allows us to set complicated constraints.

Disadvantage is, on the other hand, that we have to compute the inverse matrix of $H_C A^{-1} H_C^T$ in addition to the inverse of A, while solving Eq.(14).

Motion Control or Motion Generation If we can give all elements of $\ddot{\theta}_G$ and determine τ_C and τ_J to realize the generalized acceleration, Eq.(12) would serve as a computed torque controller. However, this is not the case in most human figures.

The reason is that human figures are under actuated due to the free 6 DOF joint of the base. The number of the generalized coordinates N_G is greater than the number of actuated joints N_J by 6. Note that even if we have more actuators than N_G, we cannot solve Eq.(12) for $\ddot{\theta}_G$ which does not satisfy the kinematic constraints of Eqs.(7)(9).

The alternative way would be to give only a part of $\ddot{\theta}_G$ and optimize the rest of them, or assume modifications in accelerations, as we will do in the dynamics filter. This usage of the function is more than a controller; we can call it a motion generator.

Simple extension of Eq.(12) gives a more practical application. Suppose we have desired trajectories $r_p \in R^{N_P}$ of some links, where N_P is the total number of elements to which desired trajectories are given. Acceleration of r_P and the generalized coordinates θ_G are related by:

$$\ddot{r}_P = J_P \ddot{\theta}_G + \dot{J}_P \dot{\theta}_G \qquad (15)$$

where $J_P \in R_G^{N_P \times N}$ is the Jacobian matrix of r_P with respect to the generalized coordinates. Eqs.(12)(15) yields:

$$\begin{pmatrix} A & -H_C^T & -H_J^T \\ H_C & O & O \\ J_P & O & O \end{pmatrix} \begin{pmatrix} \ddot{\theta}_G \\ \tau_C \\ \tau_J \end{pmatrix} = \begin{pmatrix} -b \\ -\dot{H}_C \dot{\theta}_G \\ \ddot{r}_P^d - \dot{J}_P \dot{\theta}_G \end{pmatrix} \qquad (16)$$

where $\ddot{r}_P^d$ is the desired acceleration derived from the trajectory. We can give additional $(N_J - N_P)$ elements of $\ddot{\theta}_G$ to determine the motion of some joints explicitly.

Note that we can get not only the joint torques and contact forces to realize the motion but also the modified generalized acceleration, which can be integrated to check the result, at the same time. Therefore Eqs.(12)(16) can be said to have three functions: (1)generation of physically consistent motion, (2)computation of joint torques to realize the motion, and (3)simulation of the result.

4 Feedback and Optimization

The next problems in using either Eq.(12) or Eq.(16) are:

1. How to ensure that the result motion becomes close to the original, after the modification in acceleration
2. How to select optimal solution of the equation

both of which are to be solved in considerably short time. This section describes our solution for the two problems.

4.1 Feedback Controller

Since the joint accelerations are modified, the result of their integration may be completely different from the reference motion. To ensure that the result becomes close to the reference, we include feedback terms in the desired acceleration $\ddot{\theta}_G^{d0}$ as:

$$\ddot{\theta}_G^{d0} = \ddot{\theta}_G^{ref} + K_D(\dot{\theta}_G^{ref} - \dot{\theta}_G) + K_P(\theta_G^{ref} - \theta_G) \qquad (17)$$

where θ_G^{ref} is the generalized coordinates of reference motion, K_D and K_P are constant gain matrices.

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

$$\ddot{r}_P^d = \ddot{r}_P^{ref} + K_{DP}(\dot{r}_P^{ref} - \dot{r}_P) + K_{PP}(r_P^{ref} - r_P) \qquad (18)$$

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

$$\begin{aligned} \ddot{\theta}_G^d &= \ddot{\theta}_G^{d0} + \Delta\ddot{\theta}_G^d & (19) \\ \Delta\ddot{\theta}_G^d &= J_P^{\sharp}(\ddot{r}_P^d - \ddot{r}_P^{d0}) & (20) \end{aligned}$$

where $\ddot{r}_P^{d0} \triangleq J_P\ddot{\theta}_G^{d0} + \dot{J}_P\dot{\theta}_G$, J_P is the Jacobian matrix of P with respect to the generalized coordinates, and $J_P^{\sharp}$ is the weighted pseudo-inverse of J_P.

4.2 Optimization

The problem now is how to select the best generalized acceleration $\ddot{\theta}_G$, joint torque τ_J, and contact force τ_C from the infinite number of solutions of Eq.(12), from the point of view of realizing a motion as close as possible to the reference. Time-consuming optimization search process may be applied here, however, we apply quite simple local optimization in terms of the acceleration error $|\ddot{\theta}_G^d - \ddot{\theta}_G|$ using pseudo and singularity robust (SR) inverses[13].

First, in preparation for the optimization, we derive the weighted least-square solution of Eq.(13) and the null space of W regardless of the desired acceleration:

$$x = W^{\sharp}u + (E - W^{\sharp}W)y \qquad (21)$$

where $W^{\sharp}$ is the pseudo inverse of W, y an arbitrary vector, and E the identity matrix of the appropriate size. Picking up the upper N_G rows of Eq.(21) corresponding to the generalized accelerations, we get:

$$\ddot{\theta}_G = \ddot{\theta}_G^0 + V_G y \qquad (22)$$

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

Next, we determine the arbitrary vector y to minimize the acceleration error by

$$y = V_G^*(\ddot{\theta}_G^d - \ddot{\theta}_G) \qquad (23)$$

where V_G^* is the SR inverse of V_G.

Finally, substituting the calculated y into Eq.(21), we get the optimized solution of x. Since 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 Discussions

A major character of the proposed method is the high interactivity. Since local optimization is applied, the generator does not need the whole motion data in advance. Therefore we can change the reference motion arbitrarily during the generation. This is an important feature when a robot is used in the real world with changing environment. In addition, this method can be applied to any motion.

The problem of this method is, on the other hand, that local optimization does not guarantee global stability. The result may fall into instability with wrong parameters or highly inconsistent motions. In fact, choosing appropriate paramters requires trial-and-error procedures, which would be solved in the next verision of dynamics filter.

5 Results

The proposed method was implemented on a PC with dual Alpha 21264 500MHz Processors and WindowsNT operating system, using Microsoft Visual C++. Constraints are set based on the result of collision check program using RAPID[14]. The total computation time including feedback, optimization, collision check, and 3D rendering via OpenGL, took approximately 50msec for each frame. In comparison, dynamics simulation using Eq.(14) takes approximately 30msec including collision check and rendering.

Walk Motion captured walking was input to the filter as a reference. The captured and output motions are shown in Fig.3. The kinematics properties of the real human and the human figure are set almost the same, while the dynamics properties are quite different. It may be observed that the vibration of the upper body in the result is larger than in the original, possibly to compensate for the heavy links of the human figure. Slight kinematic differences in the shapes of feet are also corrected by the filter.

Walk on slope The same captured motion was applied to a different environment with the floor being down slope of 5 degrees. The result is shown in Fig.4. It proved that the filter can generate completely new motion from one motion, of course with some limits. On an up slope, for example, we need to modify the captured data more carefully because the toe will hit the floor earlier than in the original motion, thus making the figure fall down.

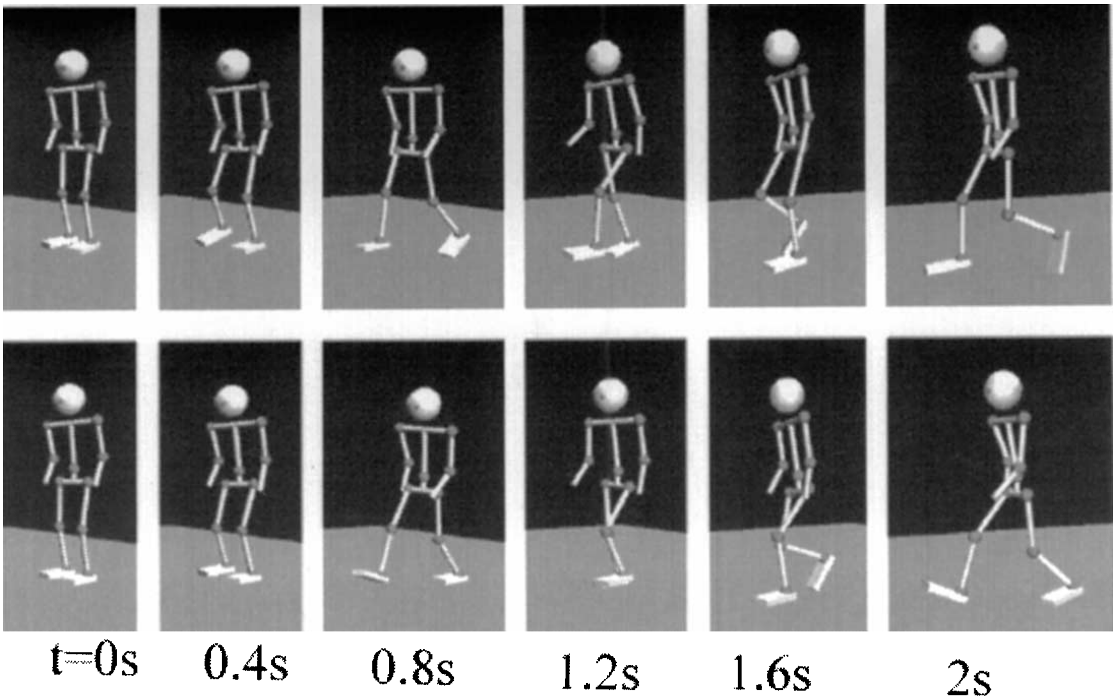


Fig.3: Captured(above) and modified(below) walking motions

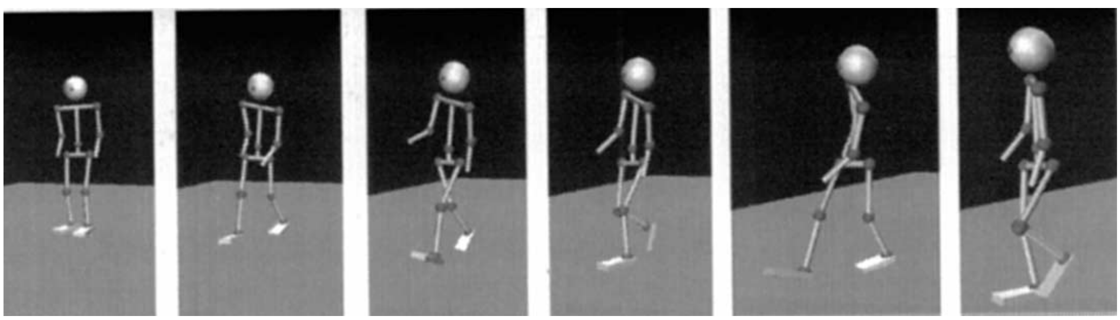


Fig.4: Walking on a slope

Jump and Kick Since our method imposes no assumption on the input motion, it is also applicable to other motions such as jump and kick as shown in Figs.5 and 6.

6 Conclusion

The contributions of this paper are summarized as follows:

1. The concept of "dynamics filter", a tool that converts a physically inconsistent motion into a consistent one, was proposed.
2. Dynamics simulation algorithm for structure-varying kinematic chains, which has several advantages against our previous method[12], was proposed.
3. An implementation of dynamics filter was presented, which has the following advantages compared to previous motion generation methods:
   - Computational efficiency — it takes only 50msec per frame for the whole computation.
   - Interactivity — we are allowed to change the reference motion during the generation.
   - Generality — the method is applicable to any motion; not restricted to specific one.
4. Some examples of generated motions were shown.

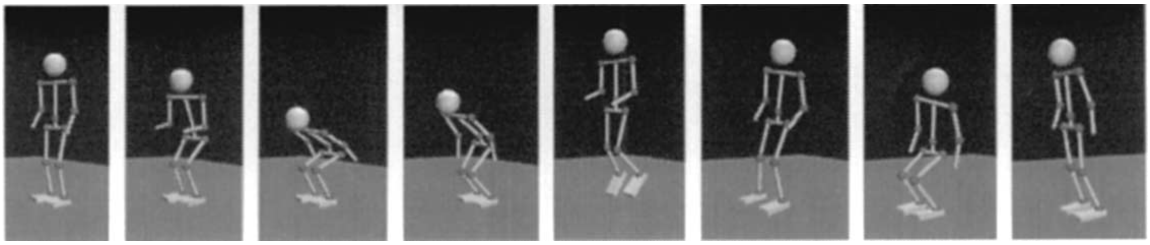


Fig.5: Jump

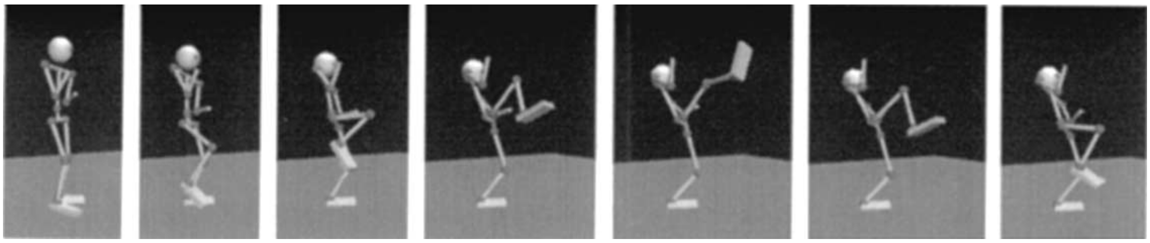


Fig.6: Kick

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

References

[1] S. Kajita and K. Tani: "Experimental Study of Biped Dynamic Walking in the Linear Inverted Pendulum Mode," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp.2885-2891, 1995.

[2] 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.

[3] 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.

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

[5] 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.

[6] Z. Popovic and A. Witkin: "Physically based motion transformation," *Proceedings of SIGGRAPH'99*, pp.11-20, 1999.

[7] 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.

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

[9] J.K. Hodgins, W.L. Wooten, D.C. Brogan, and J.F. O'Brien: "Animating Human Athletics," *Proceedings of ACM SIGGRAPH '95*, pp.71-78, 1995.

[10] C. Rose, M.F. Cohen, and B. Bodenheimer: "Verbs and Adverbs: Multidimentional Motion Interpolation," *IEEE Computer Graphics and Applications*, vol.18, no.5, pp.32-40, 1998.

[11] Michael Gleicher: "Retargetting Motion to New Characters," *Proceedings of ACM SIGGRAPH '98*, 1998.

[12] 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.

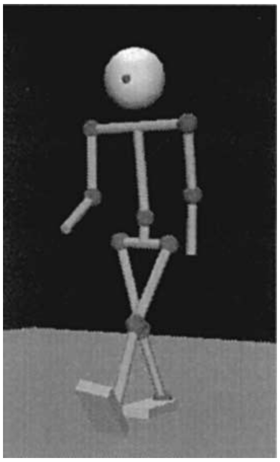
[13] Y. Nakamura and H. Hanafusa: "Inverse Kinematics Solutions with Singularity Robustness for Robot Manipulator Control," *Journal of Dynamic Systems, Measurement, and Control*, vol.108, pp.163-171, 1986.

[14] S. Gottschalk, M.C. Lin, and D. Manocha: "OBB-Tree: A Hierarchical Structure for Rapid Interface Detection," *Proceedings of SIGGRAPH '96*, 1996.

Dynamics Filter --- Concept and Implementation of On-Line Motion Generator for Human Figure

Katsu YAMANE and Yoshihiko NAKAMURA

Dept. of Mechano-Informatics, Univ. of Tokyo



- Dynamics filter: a motion generator that converts a physically inconsistent motion into a consistent one to minimize the size of database required to generate human-like motions for human figures.
- Apply stabilizing feedback control and local optimization based on the equation of motion
- Various motions including those in different environment were created from motion capture data
- Dynamics filter proved to be effective in generating a variety of motions from a small set of motions.