THE DYNAMIC BALANCE CONTROL SYSTEM FOR HUMAN BODY MODEL WITH THE QUADRATIC PROGRAMMING METHOD
二次計画法を用いた人体モデルの動的バランス制御

by

KUDOH Shunsuke
工藤 俊亮

A Master Thesis
修士論文

Submitted to
the Graduate School of
the University of Tokyo
on January 29, 2002
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Information Science

Thesis Supervisor: IKEUCHI Katsushi　池内克史
Professor of Information Science

ABSTRACT

The postural balance system is one of the most fundamental function for human voluntary motion. This system has been analyzed and modeled by many researchers in the past. There are two ways of controling balance: feed-forward and feedback. Many of the feed-forward systems are offline, which take unstable human motion as the input and convert it to the stable one. Real-time feedback systems can manipulate only weak perturbation. In order to generate the human-like motions of recovering the balance such as swinging its arms, it is nessesary to input some motion data that the body has to follow to the feed-forward system. In this thesis, we propose a new feedback balance control system for the human body that can find the optimal motion of keeping its balance against perturbation without giving any feed-forward input beforehand. This system adopts two different strategies: the dynamic balance control with the quadratic programming method and the posture control with PD control. When large perturbation is applied to the model, the motion very similar to the real human such as swinging its arms or bending its waist is generated as the optimal control.

論文要旨

人間の動作にとって，バランスの制御はもっとも基本的な働きのひとつである。それゆえ人間のバランス制御については，従来から様々な解析やモデル化がなされてきている。バランス制御には，フィードフォワードによるものとフィードバックによるものがある。前者によるものの多くはオフラインシステムであり，入力された不安定な動作を安定なものに変換するというものである。一方，後者によるリアルタイム制御では，弱い外乱に対してしか有効に制御を行なうことができない。また，人間にみられる腕を振り回したりしてバランスをとるといった動作は，これまでフィードフォワードであらかじめ指令を与えておかなければならなかった。本論文では，あらかじめ動作の指示を与えなくても外乱に対する最適なバランス制御動作を生成する，フィードバックによるバランス制御の手法を提案する。本システムは，二次計画法による動的バランス制御と PD 制御による姿勢の制御という 2 つの制御を組み合わせて用いる。このモデルに大きな外乱を与えた場合，腕を振り回したり腰を折り曲げたりという人間の動作によく似た動作が最適な制御として求められる。

Acknowledgements

I would like to express my sincere gratitude to my current theses advisor, Prof. Ikeuchi Katsushi, for his kindly support, and my previous theses advisor, Prof. Shinagawa Yoshihisa, for his valuable suggestions and comments on this thesis.

I am very grateful to Dr. Kohmura Taku, CTO of Gsport Inc., for giving me many of useful advice and encouragements, and a new notebook machine. I am also very grateful to all the members of the Shinagawa Laboratory, which suddenly vanished, for helping me much in various ways.

Contents

List of Figures

List of Tables

1 Introduction

The postural balance control system is one of the most fundamental systems that enable human voluntary motion. Every person unconsciously keeps his/her balance during his/her motion. For example, even during a simple motion such as just raising the hand, the trunk of the body is simultaneously controlled in order to counteract force and moment caused by raising the arm.

Researchers in computer graphics have been working on feed-forward human balance systems in order to create realistic human motion animation. These models often take some human gait motion as an input, evaluate the position of the center of mass projected to the ground, or the zero moment point through the motion, and finally fix it if some invalid postures are found.

Ko et al [4] converted unstable motion to a stable one by decreasing the moment applied to the body at the zero moment point to zero by translating and rotating the pelvis and torso every frame. Komura et al [5] calculated the balance motion by minimizing the integral of the auxiliary moment that must be applied to the feet if the zero moment point is out of the support surface. Tak et al [10] used the optimization method similar to the retargeting algorithm proposed by Gleicher [2] for this kind of conversion.

However, these methods are off-line algorithms which cannot be used for online simulations because the whole trajectory of the motion must be known in advance to adjust the motion.

Many researchers have investigated the characteristics of human feedback postural control by analyzing the responses of the human body to various external perturbations [9, 8, 3].

Feedback controllers have also been proposed by researchers in computer graphics who used forward dynamics to simulate human motion. Laszlo et al [7] stabilized the gait motion of a human body model using the limit cycle control, while using either the "up vector" or "swing-center-of-mass" as the regulation variable.

Wooten [11] used PD control as a postural control system, using the position of the center of gravity as a variable. However, such methods assume rather weak external perturbation. and slightly strong force can definitely cause the body to fall down.

Oshita [12] used the optimization calculation for generating the motion. This algorithm finds a physically consistent motion under some constraints when a target motion is given. The torque acting on joints and the position of ZMP are optimized in this optimization.

In the field of biomechanics, many researchers work on the human balance control. Some of them investigated features of balance control of the real human [13, 14]. They actually applied perturbations to the real human, and measured the force, the velocity, or several physical parameters. Others investigated the motion of balance recovery by stepping [15, 16]. In these researches, the motion of the real human is analyzed. On the other hand, there are some researches of simulating human balance control in biomechanics [17, 18, 19]. However, these researches use the simple inverse-pendulum model, which is not a multi-body human model.

In the field of robotics, the dynamic balance control for human models is investigated by many researchers. Yamane et al [20] proposed a method to convert a physically inconsistent motion to a physically consistent one. This was not implemented on real humanoid robots. Kagami et al [21] proposed another method for generating dynamically-stable motion for humanod robots, and implemented it on humanoid robots. However, these are the method to convert a original motion to a stabler one, then it cannot deal with unexpected large perturbations.

For biped balance, Horak et al [1] found that human use two strategies for keeping the balance, the ankle strategy and hip strategy. The ankle strategy is a strategy to use mainly the ankle joint to restore the position of the center of gravity back to the equilibrium state. This strategy is chosen when the foot surface is long enough relative to the foot length, so that the subject can fully use his/her toes to push back the body. When the foot surface is short relative to the foot length, the subject uses the hip and trunk joint to keep the balance, which is called the hip strategy. Kuo et al [6] theoretically analyzed such strategies using the musculoskeletal model.

However, we know that people use one more strategy to keep their balance: the arm strategy. When a person is about to lose his/her balance, and is under a condition that he/she cannot step out one leg, the arms are rotated recursively to work as the final servo to move the center of mass back over the feet. This strategy is effective in returning the center of mass over the feet, because the angular momentum of the trunk of the body is canceled out by the angular momentum generated by the rotation of the arms. This kind of motion can be observed when a person is wearing a snow board and is about lose the balance.

In this thesis, we propose a new dynamic balance control system for a human body model which is composed of five body segments, the foot, shank, thigh, trunk and arm. This is designed as the feedback system. Using our system, the human-like body motion is obtained as the optimal control for recovering its balance, and the arm strategy appears without any prior feed-forward input when large perturbation force is applied to the body. The motion of recovery closely resembles those by real human. Our system can counteract very large external perturbation force, and therefore it can be utilized for the feedback balance system of humanoid robots.

2 Algorithm

2.1 Overview

In this algorithm, the human body is represented as a multi-body model, which consists of fifteen linked rigid bodys, and has thirty-four degrees of freedom (DOF).

The overview of this algorithm is shown in Figure 2.1. In the first stage, the state of the human body model, such as the angles, the angular velocity, and the angular acceleration of the joints, is obtained.

In the second stage, the coefficients for obtaining the external force, the moment around the center of mass, and ZMP are calculated. They depend on the state obtained in the previous stage, and they are used for the optimization in the next stage.

In the third stage, the optimal torque for keeping the balance of the model is calculated. There are two strategies for the calculation; one is the PD control and the other is the optimization by the quadratic programming method. Superior one is chosen in each turn.

In the fourth stage, the optimal torque is applied to the model, and in the last stage, the state is integrated by the time step. When a turn is finished, the first stage of the next turn starts about a new state.

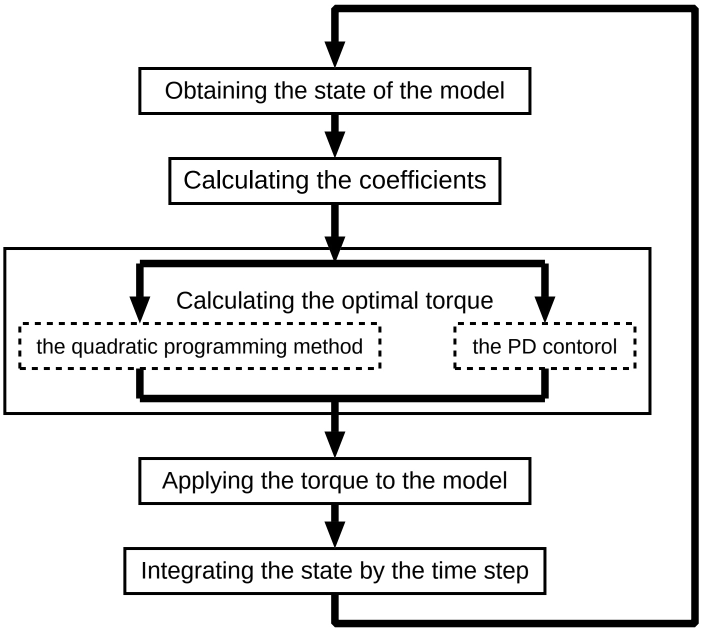


Figure 2.1: The overview of the algorithm

As shown above, the optimization is closed in each turn, then the balance control can be done locally. It is one of important features of this algorithm, because it makes the real-time control possible.

2.2 Body Model

The human body model used in this algorithm is as Figure 2.2. It consists of fifteen linked rigid bodys, and has fourteen joints and thirty-four DOF.

There are two types of joins; one is a pin joint and the other is a ball joint. The former has one DOF, and the latter has three DOF as shown in Figure 2.3.

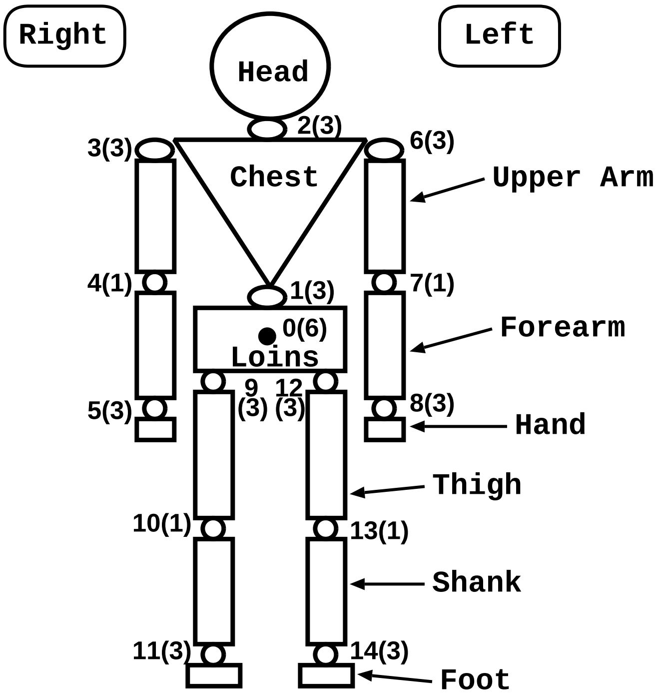


Figure 2.2: The human body model

In this thesis, there is a constraint that both feet always touch the ground and that the model cannot make a step. Because of this, the closed-loop topology occurs at the legs, and then DOF in legs are reduced from fourteen DOF to eight DOF. The optimization must be done about this reduced DOF, not about DOF of the original joints. Figure 2.4 shows DOF considering the closed-loop constraint.

Now, two variables of $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ are defined. The former is a vector whose elements represent the angles of the joints, and the latter is a vector whose elements represent the values of DOF considering the closed-loop constraint. The relational equation about these two variables is discussed later.

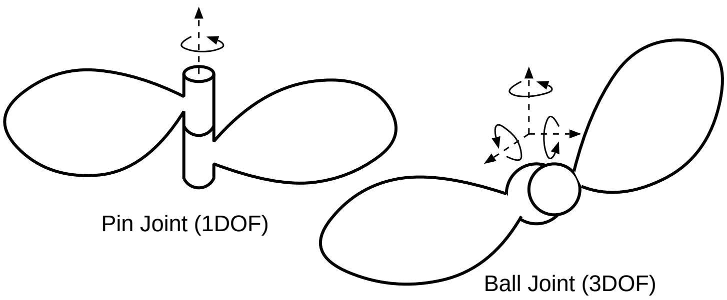


Figure 2.3: Two types of joints (pin and ball)

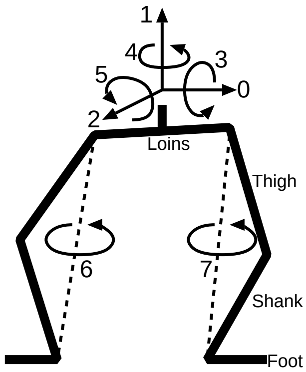


Figure 2.4: The closed-loop structure at the legs

2.3 Zero Moment Point (ZMP)

The zero moment point (ZMP) is the key concept for the dynamic balance control. Considering the static balance control, it is important that the projection of the center of mass is in the foot supporting area. ZMP is a similar concept to the projection of the center of mass in the static balance control. ZMP is the point at which the moment acts from the ground is zero. Thus, all the ground reaction force can be replaced with the equivalent force acting on ZMP (Figure 2.5). In the dynamic balance control, it is important that ZMP is in the foot support area.

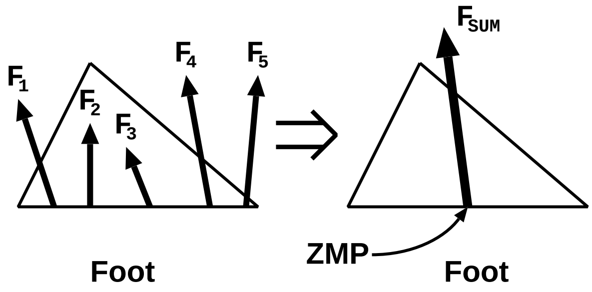


Figure 2.5: The concept of ZMP

Let $\boldsymbol{p} = {}^t(x_{\mathrm{ZMP}},\ y_{\mathrm{ZMP}},\ z_{\mathrm{ZMP}})$ be the position of ZMP, and the following equation is obtained:

$$\boldsymbol{n} + (\boldsymbol{s} - \boldsymbol{p}) \times m(\ddot{\boldsymbol{s}} - \boldsymbol{g}) = \boldsymbol{0}, \tag{2.1}$$

where m is the mass of the body, $\boldsymbol{s}$ is the position of the center of mass, $\boldsymbol{n}$ is the moment around the center of mass, and $\boldsymbol{g} = {}^t(0, g_y, 0)$ is the vector of the gravity acceleration. This equation means that the moment acting around the center of mass is equivalent to the moment generated by the force acting on ZMP. ZMP is calculated by solving this equation. Practically, $\boldsymbol{p}$ is written as ${}^t(x_{\mathrm{ZMP}},\ 0,\ z_{\mathrm{ZMP}})$, because the y-element of the foot's position is 0. Solving the above equation about the x- and z-element of ZMP, the following formula is given:

$$x_{\mathrm{ZMP}} = \frac{n_z + s_x m(\ddot{s}_y - g_y) - s_y m \ddot{s}_x}{m(\ddot{s}_y - g_y)} \tag{2.2}$$

$$z_{\mathrm{ZMP}} = \frac{-n_x - s_y m \ddot{s}_z + s_z m(\ddot{s}_y - g_y)}{m(\ddot{s}_y - g_y)}. \tag{2.3}$$

2.4 Coefficients

2.4.1 What are the coefficients

In each frame, the angle and the angular velocity of the joints are fixed, and only the angular acceleration of the joints can be changed by the system. Thus, the postural adjustment must be performed by controlling the angular acceleration. In fact, $\ddot{\boldsymbol{\varphi}}$, elements of which represent the acceleration of respective DOF, is optimized by the system. For the purpose of this optimization, it is necessary to find the functions from $\ddot{\boldsymbol{\varphi}}$ to the angular acceleration of the joints ($\ddot{\boldsymbol{\theta}}$), the external force ($\boldsymbol{f}$), the moment by $\boldsymbol{f}$ around the center of mass ($\boldsymbol{n}$), and ZMP. These functions are written as the linear equations (ZMP is written as fractions of linear equations):

$$\ddot{\boldsymbol{\theta}} = J\ddot{\boldsymbol{\varphi}} + \boldsymbol{k} \tag{2.4}$$

$$\boldsymbol{f} = C_f\ddot{\boldsymbol{\varphi}} + \boldsymbol{d}_f \tag{2.5}$$

$$\boldsymbol{n} = C_n\ddot{\boldsymbol{\varphi}} + \boldsymbol{d}_n \tag{2.6}$$

$$x_{\mathrm{ZMP}} = \frac{{}^t\boldsymbol{\alpha}_x\ddot{\boldsymbol{\varphi}} + \beta_x}{{}^t\boldsymbol{\alpha}_c\ddot{\boldsymbol{\varphi}} + \beta_c} \tag{2.7}$$

$$z_{\mathrm{ZMP}} = \frac{{}^t\boldsymbol{\alpha}_z\ddot{\boldsymbol{\varphi}} + \beta_z}{{}^t\boldsymbol{\alpha}_c\ddot{\boldsymbol{\varphi}} + \beta_c}. \tag{2.8}$$

The proof that the above functions can be written as the linear equations is given below.

- $\ddot{\boldsymbol{\theta}} = J\ddot{\boldsymbol{\varphi}} + \boldsymbol{k}$

When $\boldsymbol{\varphi}$ is given, $\boldsymbol{\theta}$ is determined uniquely. Therefor, the elements of $\boldsymbol{\theta}$ are a function of $\boldsymbol{\varphi}$:

$$\theta_i = \theta_i(\boldsymbol{\varphi}). \tag{2.9}$$

The derivatives of this are

$$\dot{\theta}_i = \sum_j \dot{\varphi}_j \frac{\partial \theta_i}{\partial \varphi_j} \tag{2.10}$$

$$\begin{aligned} \ddot{\theta}_i &= \sum_j \left\{ \ddot{\varphi} \frac{\partial \theta_i}{\partial \varphi_j} + \sum_k \dot{\varphi}_j \dot{\varphi}_k \frac{\partial^2 \theta_i}{\partial \varphi_j \partial \varphi_k} \right\} \\ &= \sum_j \ddot{\varphi}_j \frac{\partial \theta_i}{\partial \varphi_j} + \sum_j \sum_k \dot{\varphi}_j \dot{\varphi}_k \frac{\partial^2 \theta_i}{\partial \varphi_j \partial \varphi_k}. \end{aligned} \tag{2.11}$$

The second term of (2.11) is a constant because $\boldsymbol{\varphi}$ and $\dot{\boldsymbol{\varphi}}$ are fixed in each frame. Thus, the right side of (2.11) can be written as a linear expression of $\ddot{\boldsymbol{\varphi}}$.

- $\boldsymbol{f} = C_f \ddot{\boldsymbol{\varphi}} + \boldsymbol{d}_f$

  The position of the center of mass is also a function of $\boldsymbol{\varphi}$:

$$\boldsymbol{s} = \boldsymbol{s}(\boldsymbol{\varphi}), \tag{2.12}$$

then, in the same way as the above, it can be said that the second derivative of $\boldsymbol{s}$ is written as a linear function of $\boldsymbol{\varphi}$. Because the external force is expressed as

$$\boldsymbol{f} = m\ddot{\boldsymbol{s}} - m\boldsymbol{g}, \tag{2.13}$$

when m is the mass of the body and $\boldsymbol{g}$ is the gravity acceleration, the external force $\boldsymbol{f}$ can be written as a linear expression of $\ddot{\boldsymbol{\varphi}}$.

- $\boldsymbol{n} = C_n \ddot{\boldsymbol{\varphi}} + \boldsymbol{d}_n$

  $\boldsymbol{n}$, the moment around the center of mass, is written as

$$\boldsymbol{n} = \sum_i \left\{ (\boldsymbol{s}_i - \boldsymbol{s}) \times m \ddot{\boldsymbol{s}}_i + I_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (I_i \boldsymbol{\omega}_i) \right\}, \tag{2.14}$$

while $\boldsymbol{s}_i$ is the center of mass of each rigid body, I_i is the moment of inertia of it, and $\boldsymbol{\omega}_i$ is the angular velocity of it around $\boldsymbol{s}_i$. The first term of the right side of (2.14) can be written as a linear expression of $\ddot{\boldsymbol{\varphi}}$, because $\ddot{\boldsymbol{s}}$ is a linear function of $\ddot{\boldsymbol{\varphi}}$. The second term can also be written as a linear expression of $\ddot{\boldsymbol{\varphi}}$, because $\boldsymbol{\omega}_i = \boldsymbol{\omega}_i(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}})$, then $\dot{\boldsymbol{\omega}}_i$ is a linear function of $\ddot{\boldsymbol{\varphi}}$:

$$\dot{\omega}_i = \sum_j \left(\dot{\varphi}_j \frac{\partial \omega_i}{\partial \varphi_j} + \ddot{\varphi}_j \frac{\partial \omega_i}{\partial \dot{\varphi}_j} \right). \tag{2.15}$$

The third term is a constant. Therefor $\boldsymbol{n}$ can be written as a linear expression of $\ddot{\boldsymbol{\varphi}}$.

- $x_{\mathrm{ZMP}} = \frac{{}^t\boldsymbol{\alpha}_x \ddot{\boldsymbol{\varphi}} + \beta_x}{{}^t\boldsymbol{\alpha}_c \ddot{\boldsymbol{\varphi}} + \beta_c}, \quad z_{\mathrm{ZMP}} = \frac{{}^t\boldsymbol{\alpha}_z \ddot{\boldsymbol{\varphi}} + \beta_z}{{}^t\boldsymbol{\alpha}_c \ddot{\boldsymbol{\varphi}} + \beta_c}$

  Rewriting (2.2) and (2.3) with $\boldsymbol{f}$ and $\boldsymbol{n}$:

$$x_{\mathrm{ZMP}} = \frac{n_z + s_x f_y - s_y f_x}{f_y} \tag{2.16}$$

$$z_{\mathrm{ZMP}} = \frac{-n_x - s_y f_z + s_z f_y}{f_y}, \tag{2.17}$$

then these can be written as the fractions of linear expressions of $\ddot{\boldsymbol{\varphi}}$, because $\boldsymbol{f}$ and $\boldsymbol{n}$ are linear functions of $\ddot{\boldsymbol{\varphi}}$,

2.4.2 Obtaining Coefficients

As shown above, $\ddot{\boldsymbol{\theta}}$, $\boldsymbol{f}$, and $\boldsymbol{n}$ are expressed as the linear functions of $\ddot{\boldsymbol{\varphi}}$:

$$\begin{aligned} \ddot{\boldsymbol{\theta}} &= J\ddot{\boldsymbol{\varphi}} + \boldsymbol{k} \qquad (2.18) \\ \boldsymbol{f} &= C_f\ddot{\boldsymbol{\varphi}} + \boldsymbol{d}_f \qquad (2.19) \\ \boldsymbol{n} &= C_n\ddot{\boldsymbol{\varphi}} + \boldsymbol{d}_n. \qquad (2.20) \end{aligned}$$

On the other hand, these values are calculated as the functions of $\boldsymbol{\varphi}$, $\dot{\boldsymbol{\varphi}}$, and $\ddot{\boldsymbol{\varphi}}$, using the inverse dynamics about the human model:

$$\begin{aligned} \ddot{\boldsymbol{\theta}} &= \ddot{\boldsymbol{\theta}}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}, \ddot{\boldsymbol{\varphi}}) \qquad (2.21) \\ \boldsymbol{f} &= \boldsymbol{f}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}, \ddot{\boldsymbol{\varphi}}) \qquad (2.22) \\ \boldsymbol{n} &= \boldsymbol{n}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}, \ddot{\boldsymbol{\varphi}}). \qquad (2.23) \end{aligned}$$

Therefore, the coefficients can be determined in the following way.

Now, we take the case of $\ddot{\boldsymbol{\theta}}$ for a example. Let $\mathbf{0}$ be a null vector and $\mathbf{1}_i$ be a vector whose elements are zero except for the i-th element:

$$\mathbf{1}_i = {}^t(0, \ldots, 0, \overset{i\text{-th}}{\overset{\vee}{1}}, 0, \ldots, 0). \qquad (2.24)$$

First, the constant $\boldsymbol{k}$ is determined by $\ddot{\boldsymbol{\theta}}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}, \mathbf{0})$, because (2.18) shows

$$\ddot{\boldsymbol{\theta}}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}, \mathbf{0}) = \boldsymbol{k}. \qquad (2.25)$$

Next, $\boldsymbol{j}_1$, the first column of J, is determined by $\ddot{\boldsymbol{\theta}}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}, \mathbf{1}_1) - \boldsymbol{k}$, because (2.18) shows

$$\ddot{\boldsymbol{\theta}}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}, \mathbf{1}_1) = J\mathbf{1}_1 + \boldsymbol{k} = \boldsymbol{j}_1 + \boldsymbol{k}. \qquad (2.26)$$

$$
\begin{aligned}
&\boldsymbol{k} = \ddot{\boldsymbol{\theta}}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}, \mathbf{0}) \\
&\boldsymbol{d}_f = \boldsymbol{f}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}, \mathbf{0}) \\
&\boldsymbol{d}_n = \boldsymbol{n}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}, \mathbf{0}) \\
&\texttt{for}\ (i = 1;\ i \leq N_{DOF};\ \texttt{++}i)\ \{ \\
&\qquad \boldsymbol{j}_i = \ddot{\boldsymbol{\theta}}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}, \mathbf{1}_i) - \boldsymbol{k} \\
&\qquad \boldsymbol{c}_{f\,i} = \boldsymbol{f}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}, \mathbf{1}_i) - \boldsymbol{d}_f \\
&\qquad \boldsymbol{c}_{ni} = \boldsymbol{n}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}, \mathbf{1}_i) - \boldsymbol{d}_n \\
&\}
\end{aligned}
$$

Figure 2.6: The way to determine the coefficients

In the same way, $\boldsymbol{j}_i$ is determined by $\ddot{\boldsymbol{\theta}}(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}, \mathbf{1}_i) - \boldsymbol{k}$.

In the cases of $\boldsymbol{f}$ and $\boldsymbol{n}$, the coefficients are determined in this way. The pseudo-program to determine these coefficients is shown in Figure 2.6. When J, C_f, C_n, $\boldsymbol{k}$, $\boldsymbol{d}_f$, and $\boldsymbol{d}_n$ are determined, the coefficients about ZMP, $\boldsymbol{\alpha}_x$, $\boldsymbol{\alpha}_z$, $\boldsymbol{\alpha}_c$, β_x, β_z, and β_c, are calculated by (2.16) and (2.17).

2.5 Optimization

There are two strategies of optimization in this algorithm, using the quadratic programming method and using the PD control. In this section, first, the descriptions of each strategy are given, and then the way to choose the strategy is explained.

2.5.1 Quadratic Programming Method

The quadratic programming strategy is used when the perturbations are large, and the aim is to return the center of mass to the initial position. For attaining this goal, the motion is generated under the constraint that the accelelation of the center of mass has a proper value. In this strategy, it is not important that the posture of the model becomes similar to the initial posture, and hence an optimal motion is generated with no relation to the initial posture.

The quadratic programming method is a variation of the mathematical programming method. It is a method to solve the quadratic programming problem, which consists of a objective function and constraints, and finds the value of variables minimizing the objective function. The objective function is given as a quadratic function, and the constraints are given as the linear equations or linear inequations. In this algorithm, the variable is $\ddot{\boldsymbol{\varphi}}$, then the objective function and the constraints must be written as functions of $\ddot{\boldsymbol{\varphi}}$.

First, we define the objective function. In this algorithm, the parameters to be minimized are the square sum of the angular acceleration of joints and the opposit of the y-element of the acceleration of the center of mass. The latter is required to continue standing upright. The model will sit down without it, because the posture of sitting down is stabler than that of standing upright. The former is for generating the low-cost motion. The reason why the angular acceleration is chosen instead of the torque, is that the torque acting at leg joints is not determined uniquely because of the closed-loop problem, while the acceleration is determined uniquely.

The square sum of the angular acceleration of joints is written as

$$ {}^t\ddot{\boldsymbol{\theta}} C_\theta \ddot{\boldsymbol{\theta}}, \tag{2.27} $$

while C_θ is a weight matrix, which is a diagonal matrix. As the objective function must be a quadratic function of $\ddot{\boldsymbol{\varphi}}$, the above formula is rewritten as

$$ \begin{aligned} &{}^t\left(J\ddot{\boldsymbol{\varphi}} + \boldsymbol{k}\right) C_\theta \left(J\ddot{\boldsymbol{\varphi}} + \boldsymbol{k}\right) \\ &\quad = {}^t\ddot{\boldsymbol{\varphi}} \left({}^tJC_\theta J\right) \ddot{\boldsymbol{\varphi}} + \left(2\,{}^t\boldsymbol{k} C_\theta J\right) \ddot{\boldsymbol{\varphi}} + \text{Const.} \end{aligned} \tag{2.28} $$

Constants have no relation to a objective function.

Because the external force is written as

$$ \boldsymbol{f} = C_f \ddot{\boldsymbol{\varphi}} + \boldsymbol{d}_f = m\ddot{\boldsymbol{s}} - m\boldsymbol{g}, \tag{2.29} $$

the y-element of the acceleration of the center of mass is written as

$$ \ddot{s}_y = \frac{{}^t\boldsymbol{c}_{f_y}}{m} \ddot{\boldsymbol{\varphi}} + \left(\frac{d_{f_y}}{m} + g_y\right), \tag{2.30} $$

while $\boldsymbol{c}_{f_y}$ is the second row of C_f and d_{f_y} are the y-elements of $\boldsymbol{d}_f$.

Next, we define the constraints. In this algorithm, constraints are set about the following things:

1. The range of the angles, the angular velocity, and the angular acceleration of joints.

2. The boundary in which ZMP can exist.

3. The acceleration of the center of mass.

4. The symmetry of the closed-loop of legs.

Now, let the range of the angles, the angular velocity, and the range of the angular acceleration be

$$\boldsymbol{\theta}_{\min} \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_{\max}, \tag{2.31}$$

$$\dot{\boldsymbol{\theta}}_{\min} \leq \dot{\boldsymbol{\theta}} \leq \dot{\boldsymbol{\theta}}_{\max}, \tag{2.32}$$

$$\ddot{\boldsymbol{\theta}}_{\min} \leq \ddot{\boldsymbol{\theta}} \leq \ddot{\boldsymbol{\theta}}_{\max}. \tag{2.33}$$

All constraints must be expressed as those about the acceleration, and hence the above ones can be rewritten as:

$$\xi_{\min i} < \ddot{\theta}_i < \xi_{\max i} \quad (1, 2, \dots, N_{\text{joints}}), \tag{2.34}$$

where

$$\xi_{\min i} = \begin{cases} 0 & (\theta_i < \theta_{\min i} \text{ or } \dot{\theta}_i < \dot{\theta}_{\min i}) \\ \ddot{\theta}_{\min i} & (\text{othrewise}) \end{cases} \tag{2.35}$$

$$\xi_{\max i} = \begin{cases} 0 & (\theta_i > \theta_{\max i} \text{ or } \dot{\theta}_i > \dot{\theta}_{\max i}) \\ \ddot{\theta}_{\max i} & (\text{othrewise}) \end{cases} . \tag{2.36}$$

It means that if the angle or the angular velocity of a joint is out of the range, the angular acceleration is generated only in the direction to make the motion slow. These have to be rewritten again to the constraints about $\ddot{\boldsymbol{\varphi}}$ as follows:

$${}^t\boldsymbol{j}_i\ddot{\boldsymbol{\varphi}} \geq \xi_{\min i} - k_i \tag{2.37}$$

$${}^t\boldsymbol{j}_i\ddot{\boldsymbol{\varphi}} \leq \xi_{\min i} - k_i, \tag{2.38}$$

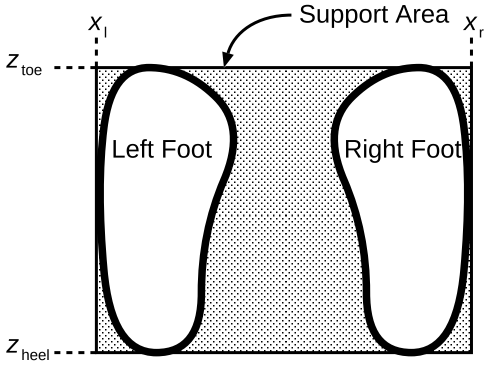


Figure 2.7: The area in which ZMP can exist.

while ${}^t\boldsymbol{j}_i$ is the i-th row of J and k_i is the i-th element of $\boldsymbol{k}$.

The next constraint is about the area in which ZMP can exist. Figure 2.7 shows the area. Because it is considered a rectangle here, the constraint is written as

$$x_r \leq \quad x_{\mathrm{ZMP}} \quad \leq x_l \tag{2.39}$$

$$z_{\mathrm{heel}} \leq \quad z_{\mathrm{ZMP}} \quad \leq z_{\mathrm{toe}}. \tag{2.40}$$

They must be rewritten as the inequations of $\ddot{\boldsymbol{\varphi}}$:

$$\begin{aligned}
{}^t(x_r\boldsymbol{\alpha}_c - \boldsymbol{\alpha}_x)\ddot{\boldsymbol{\varphi}} &\leq \beta_x - x_r\beta_c & (2.41)\\
{}^t(x_l\boldsymbol{\alpha}_c - \boldsymbol{\alpha}_x)\ddot{\boldsymbol{\varphi}} &\geq \beta_x - x_l\beta_c & (2.42)\\
{}^t(z_{\mathrm{heel}}\boldsymbol{\alpha}_c - \boldsymbol{\alpha}_z)\ddot{\boldsymbol{\varphi}} &\leq \beta_z - z_{\mathrm{heel}}\beta_c & (2.43)\\
{}^t(z_{\mathrm{toe}}\boldsymbol{\alpha}_c - \boldsymbol{\alpha}_z)\ddot{\boldsymbol{\varphi}} &\geq \beta_z - z_{\mathrm{toe}}\beta_c. & (2.44)
\end{aligned}$$

The third constraint is about the acceleration of the center of mass. The x-element and the z-element of the acceleration of the center of mass are determined by its position and velocity. In fact, they are determined to be proportional to the opposite of the position and the velocity:

$$\ddot{s}_x \propto -s_x, \quad \ddot{s}_x \propto -\dot{s}_x \tag{2.45}$$

$$\ddot{s}_z \propto -s_z, \quad \ddot{s}_z \propto -\dot{s}_z. \tag{2.46}$$

These can be written as the linear equations of $\ddot{\boldsymbol{\varphi}}$:

$$\ddot{s}_x = f_x(s_x, \dot{s}_x) \tag{2.47}$$

$$\ddot{s}_z = f_z(s_z, \dot{s}_z). \tag{2.48}$$

The last constraint is about the symmetry of the closed-loop topology at the legs. Joins in the legs are not controlled directly, but the parameters of $\ddot{\boldsymbol{\varphi}}$ are controlled, and hence the structure of legs is not considered in optimization. On the other hand, the structure, especially the symmetry, is very important in postural adjustment by real humans; i.e., they move in the way such that the symmetry at legs is not broken as much as possible.

In this algorithm, constraints about symmetry are expressed as constraints that "the angles of two knee joints are to be as same as possible and that of two x-elements of ankle joints are also to be as same as possible". This is written as follows:

$$\ddot{\boldsymbol{\theta}}_{\text{rknee}} \lesseqgtr \ddot{\boldsymbol{\theta}}_{\text{lknee}} \qquad (\boldsymbol{\theta}_{\text{rknee}} \gtreqless \boldsymbol{\theta}_{\text{lknee}}) \tag{2.49}$$

$$\ddot{\boldsymbol{\theta}}_{x\text{rankle}} \lesseqgtr \ddot{\boldsymbol{\theta}}_{x\text{lankle}} \qquad (\boldsymbol{\theta}_{x\text{rankle}} \gtreqless \boldsymbol{\theta}_{x\text{lankle}}), \tag{2.50}$$

where the subscripts "rknee" and "lknee" stand for the right and the left knees, and "$x$rankle" and "$x$lankle" stand for the x-elements of the right and the left ankles. These are rewritten as the constraints about $\ddot{\boldsymbol{\varphi}}$:

$$^{t}(\boldsymbol{j}_{\mathrm{rknee}} - \boldsymbol{j}_{\mathrm{lknee}})\ddot{\boldsymbol{\varphi}} \lesseqgtr k_{\mathrm{lknee}} - k_{\mathrm{rknee}} \qquad (\boldsymbol{\theta}_{\mathrm{rknee}} \gtrless \boldsymbol{\theta}_{\mathrm{lknee}}) \tag{2.51}$$

$$^{t}(\boldsymbol{j}_{x\mathrm{rankle}} - \boldsymbol{j}_{x\mathrm{lankle}})\ddot{\boldsymbol{\varphi}} \lesseqgtr k_{x\mathrm{lankle}} - k_{x\mathrm{rankle}} \qquad (\boldsymbol{\theta}_{x\mathrm{rankle}} \gtrless \boldsymbol{\theta}_{x\mathrm{lankle}}). \tag{2.52}$$

There is no constraint about them in case

$$\boldsymbol{\theta}_{\mathrm{rknee}} = \boldsymbol{\theta}_{\mathrm{lknee}} \quad \text{or} \quad \boldsymbol{\theta}_{x\mathrm{rankle}} = \boldsymbol{\theta}_{x\mathrm{lankle}}. \tag{2.53}$$

In summary, the quadratic programming problem is as follows:

$$\text{minimize}\quad {}^t\ddot{\boldsymbol{\varphi}}\left({}^tJC_\theta J\right)\ddot{\boldsymbol{\varphi}}+\left(2\,{}^t\boldsymbol{k}C_\theta J-{}^t\boldsymbol{c}_{f_y}/m\right)\ddot{\boldsymbol{\varphi}}, \tag{2.54}$$

subject to:

$${}^t\boldsymbol{j}_i\ddot{\boldsymbol{\varphi}} \geq \xi_{\min i}-k_i \qquad (1,2,\ldots,N_{\text{joints}}) \tag{2.55}$$

$${}^t\boldsymbol{j}_i\ddot{\boldsymbol{\varphi}} \leq \xi_{\max i}-k_i \qquad (1,2,\ldots,N_{\text{joints}}) \tag{2.56}$$

$${}^t(x_r\boldsymbol{\alpha}_c-\boldsymbol{\alpha}_x)\ddot{\boldsymbol{\varphi}} \leq \beta_x-x_r\beta_c \tag{2.57}$$

$${}^t(x_l\boldsymbol{\alpha}_c-\boldsymbol{\alpha}_x)\ddot{\boldsymbol{\varphi}} \geq \beta_x-x_l\beta_c \tag{2.58}$$

$${}^t(z_{\text{heel}}\boldsymbol{\alpha}_c-\boldsymbol{\alpha}_z)\ddot{\boldsymbol{\varphi}} \leq \beta_z-z_{\text{heel}}\beta_c \tag{2.59}$$

$${}^t(z_{\text{toe}}\boldsymbol{\alpha}_c-\boldsymbol{\alpha}_z)\ddot{\boldsymbol{\varphi}} \geq \beta_z-z_{\text{toe}}\beta_c \tag{2.60}$$

$$\ddot{s}_x=f_x(s_x,\dot{s}_x) \tag{2.61}$$

$$\ddot{s}_z=f_z(s_z,\dot{s}_z) \tag{2.62}$$

$${}^t(\boldsymbol{j}_{\text{rknee}}-\boldsymbol{j}_{\text{lknee}})\ddot{\boldsymbol{\varphi}} \lesseqgtr k_{\text{lknee}}-k_{\text{rknee}} \quad (\boldsymbol{\theta}_{\text{rknee}} \gtreqless \boldsymbol{\theta}_{\text{lknee}}) \tag{2.63}$$

$${}^t(\boldsymbol{j}_{x\text{rankle}}-\boldsymbol{j}_{x\text{lankle}})\ddot{\boldsymbol{\varphi}} \lesseqgtr k_{x\text{lankle}}-k_{x\text{rankle}} \quad (\boldsymbol{\theta}_{x\text{rankle}} \gtreqless \boldsymbol{\theta}_{x\text{lankle}}). \tag{2.64}$$

2.5.2 PD control

The strategy of PD control is adopted for small perturbations and for the final posture adjustment. The aim of this is to return the posture of the model to the initial one, then the motion to make the posture similar to the initial one is generated.

The PD control is a way to determine the acceleration of joints depending on the difference of the current position and velocity from the target ones:

$$\ddot{\boldsymbol{\theta}} = K_p(\boldsymbol{\theta} - \boldsymbol{\theta}_0) + K_d(\dot{\boldsymbol{\theta}} - \dot{\boldsymbol{\theta}}_0), \tag{2.65}$$

while $\boldsymbol{\theta}_0$ and $\dot{\boldsymbol{\theta}}_0$ are the target angle and angular velocity, and $K_p(< 0)$ and $K_d(< 0)$ are constants. These constants are determined to maximize $|K_p|$ and $|K_d|$ while keeping the ratio of these and making ZMP within the support area.

2.5.3 Choice of the Strategy

In this section, the way to choose the appropriate strategies from the quadratic programming method or the PD control is described. The advantage of the balance control with the quadratic programming method is that it is possible to generate the optimal motion with no relation to the initial posture. However, because of this advantage, this strategy also generate a big motion against a small perturbation, and hence the model cannot keep the posture of standing upright. On the other hand, the PD control can keep the model standing upright, but cannot deal with large perturbations at all.

Considering these features, the choice of the strategies is as follows. When the projection of the center of mass is near the center of the foot support area, the PD control is chosen, and otherwise the quadratic programming method is chosen. In this algorithm, the situation that the projection is near the center of the foot support area is considered that the model is under a stable state. The concept of this choice is shown in

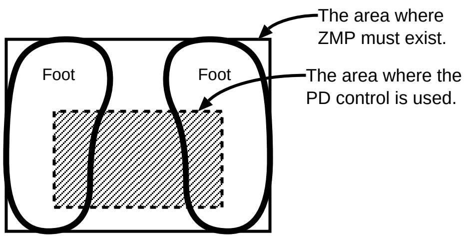


Figure 2.8: The area where the PD control is used, and the area where the quadratic programming method is used.

Figure 2.8.

3 Experiments

3.1 Implementation

The proposed algorithm is implemented on a PC with a Intel Pentium4 Processor (1.7GHz) and Linux 2.2.17, using the C++ language, egcs-2.91.66. The SD/FAST library, a product of Symbolic Dynamics Inc., is used for the dynamic simulation, the Qt library, a product of Trolltech, is used for the GUI toolkit, and Mesa 3.4.2, a OpenGL compatible 3D Graphics library, is used for rendering the results. The analysis takes approximately 1 second per a frame.

3.2 Experiments

The physical parameters of the model are determined based on the author's body. The mass and the length of each rigid body is defined as Table 3.1. The distance between shoulder joints is forty centimeters, and that between hip joints is thirty centimeters. The moment of inertia is calculated by approximation cylinder using these value.

The four patterns of perturbation are added to the model:

- the force of 500N for 0.1 second from the backward direction,

elements	mass (kg)	length (cm)
head	4.90	25
chest	18.06	40
loins	12.04	20
upper arm	2.52	25
forearm	1.54	25
hand	0.49	10
thigh	7.98	40
shank	3.71	40
foot	1.26	25

Table 3.1: The weight and length of rigid bodys

- the force of 300N for 0.1 from the forward direction,
- the force of a sinusoidal wave ($100 \sin(2\pi t)$ [N]) from the forward direction,
- the force of 200N for 0.1 from the right direction.

The analysis is performed every 0.01 second, and all perturbations act on near the center of mass. The balance control is begun after 0.2 second since perturbations occur.

3.3 Results

The results of the experiments are shown in Figure 3.1–3.4. In the first experiment, when the force of 500N acts from the back, the model keeps its balance by swinging its arms and bending down.

The meaning of these is as the following. When the force acts, the angular momentum is generated and makes the model fall down forward. In order to reduce the effect of it, the model must generate the angular momentum of the same direction, and the motion of swinging its arms and bending down is very effective for the purpose.

These motions are often observed as the balance recovery motion by real humans. They, probably empirically, select these motion for keeping their balance.

In the second experiment, when the force of 300N acts form the front, the model also swing its arms to keep its balance. In this case, the direction of the force is opposite from the previous case, and then the rotation of the arms is opposite, too. Because the waist cannot bend backward as much as forward, the arms must be swung more than the previous case.

In the third experiment, the force of a sinusoidal wave always acts on the model. The frequency of it is 1Hz. In this case, the PD control is usually chosen, and sometimes, when the model cannot keep its balance by the PD control, the quadratic programming method is chosen and the arms are quickly moved for adjusting the angular momentum acting on the model.

In the last experiment, the force of 200N acts from the right side direction. The model also keeps its balance by swing its arms in this case. However, when the force acts from the side directions, the model can keep its balance against the smaller force than when it acts from the front or the back. It is probably because there are a fewer DOF in the direction of the sides than in the direction of the front and the back.

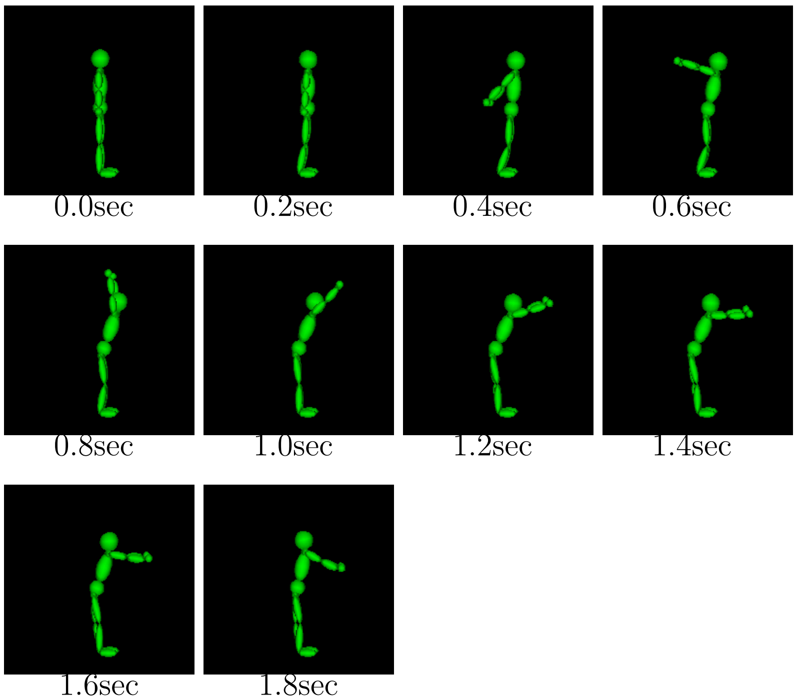


Figure 3.1: The result (1)

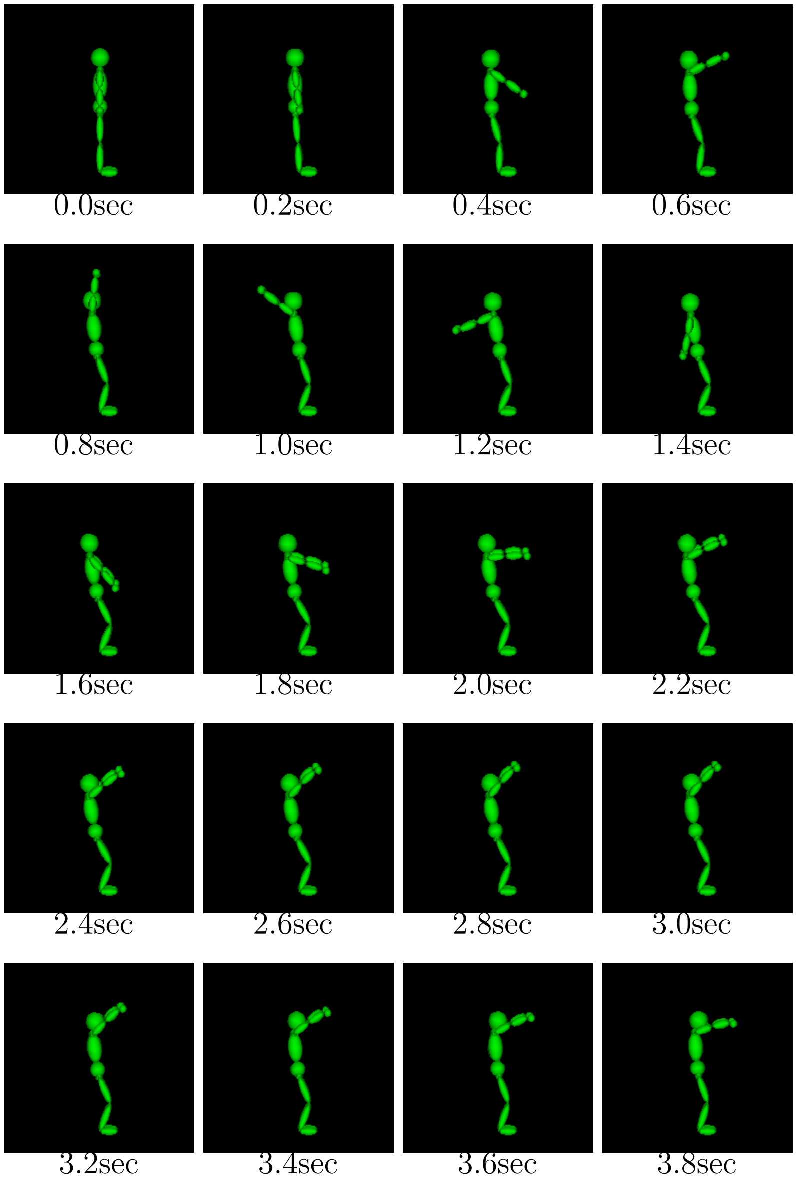


Figure 3.2: The result (2)

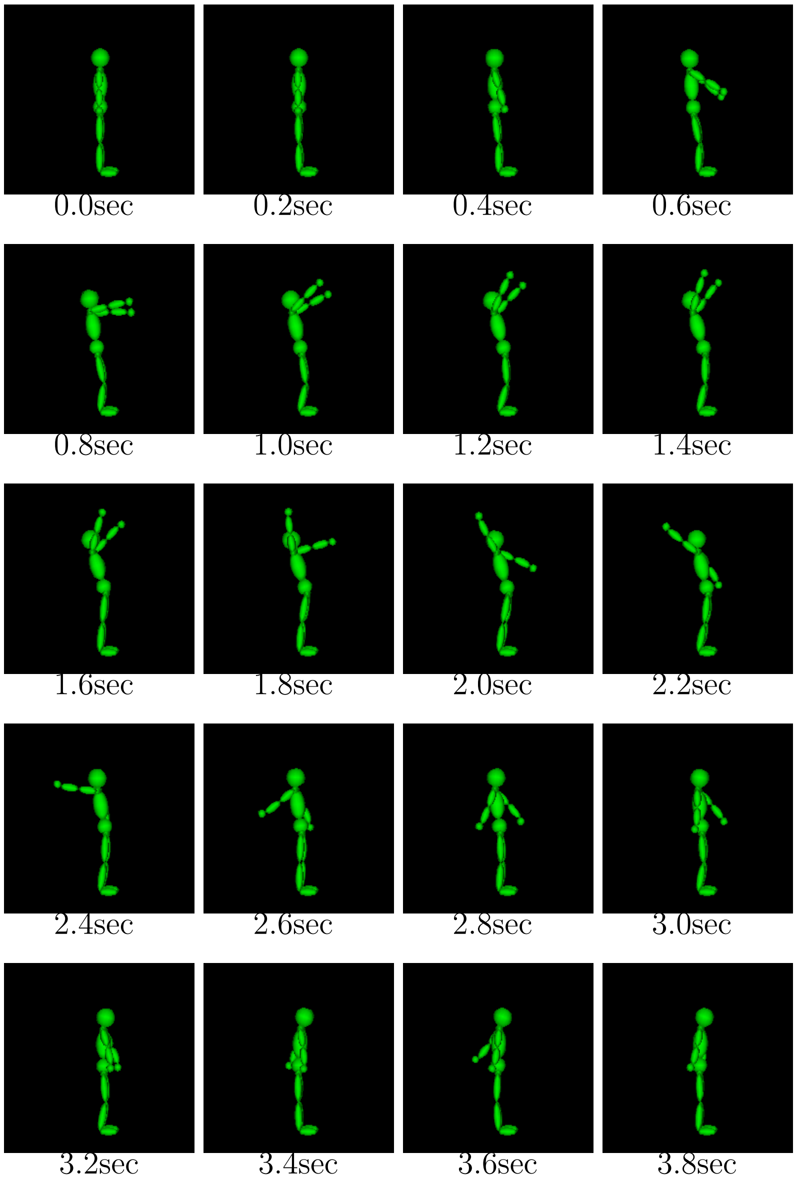
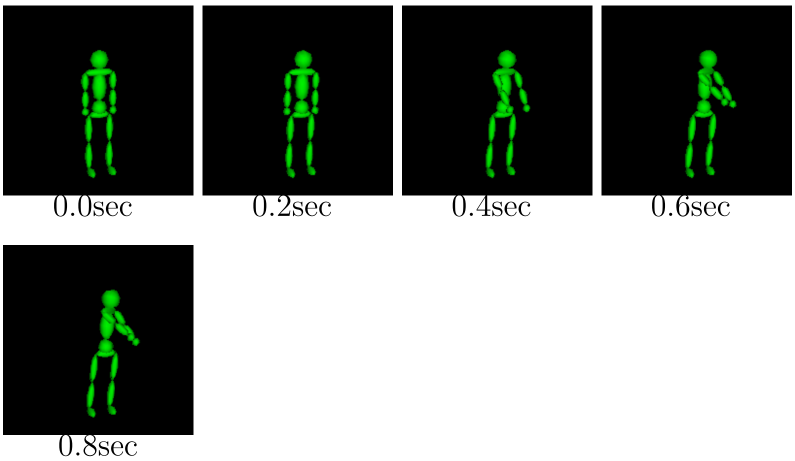


Figure 3.3: The result (3)



0.0sec 0.2sec 0.4sec 0.6sec 0.8sec

Figure 3.4: The result (4)

4 Conclusions

In this thesis, a new algorithm for the postural adjustment of the human body model is proposed and implemented. This method is a feedback system and can deal with large perturbations. It switches two methods to generate motions; one uses the quadratic programming method and the other uses the PD control. The former is for large perturbations and the latter is for small perturbations or final posture adjustment. The choice of them depends on the position of the center of mass.

In the experiments, the motion which is similar to the real human motion appeared, such as swinging the arms and bending down. We empirically know that these motion are effective for keeping our balance, and this fact is experimentally confirmed by the proposed optimization calculation.

5 Future work

In the proposed algorithm, the feet must touch the ground and cannot move. However, this constraint is too strong. When a large perturbation is applied, stepping motion is usually selected for preventing the body from falling down. Therefore, it is desirable that the algorithm allows the stepping motion.

Moreover, although only standing upright is permitted for the initial posture in this algorithm, it is not a reasonable constraints from the viewpoint of the applications of this algorithm. In the future, the algorithm will be able to deal with various situations, such as the perturbation during the gait.

References

[1] Horak F.B. and Nashner L.M. Central programming of postural movements: adaptation to altered support-surface configurations. *Journal of Neurophysiology*, 55:1369–1381, 1986.

[2] Michael Gleicher. Retargetting motion to new characters. *Computer Graphicsi Proceedings, Annual Conference Series*, pages 33–42, 1998.

[3] Akimasa Ishida and Shinji Miyazaki. Maximum likelihood identification of a posture control system. *IEEE Transactions on Biomedical Engineering*, BME-34(1):1–5, 1987.

[4] Hyeongseok Ko and Norman I. Badler. Animating human locomotion with inverse dynamics. *IEEE Computer Graphics and Applications*, March:50–59, 1996.

[5] Taku Komura, Yoshihisa Shinagawa, and Tosiyasu L. Kunii. Creating and retargetting motion by the musculoskeletal human body model. *The Visual Computer*, (5):254–270, 2000.

[6] Arthur D. Kuo and Felix E. Zajac. Human standing posture: Multi-joint movement strategies based on biomechanical constraints. *Progress in Brain Research*, 97, 2000.

[7] Joseph Laszlo, Michiel van de Panne, and Eugene Fiume. Limit cycle control and its application to the animation of balancing and walk-

ing. *Computer Graphics (Proceedings of SIGGRAPH 96)*, 30:155–162, 1996.

[8] Woollacott M.H., von Hosten C, and R ösblad B. Fixed patterns of rapid postural responses among leg muscles during stance. *Experimental Brain Research*, 30:13–24, 1977.

[9] Moore S.P., Rushmer D.S., Windus S.L., and Nashner L.M. Human automatic postural responses: responses to horizontal perturbations of stance in multiple directions. *Experimental Brain Research*, 73:648–658, 1988.

[10] Seyoon Tak, Oh young Song, and Hyeong-Seok Ko. Motion balance filtering. *Computer Graphics Forum*, 19(3), 2000.

[11] Wayne Wooten. Simulations of Leaping, Tumbling, Landing, and Balancing Humans. PhD thesis, Georgia Institue of Technology, 1997.

[12] M. Oshita and A. makinouchi. A Dynamic Motion Control Technique for Human-like Articulated Figures. *Eurotraphics 2001*, vol.20, Number3, 2001

[13] Mian-Ju Gu, Albert B. Schults, Nei T. Shepard and Neil B. Alexander. Postural control in young and elderly adults when stance is perturbed: Dynamics. *Jounal of Biomechanics*, 29(3), 319–329, 1996

[14] S. Rietdyk, A.E. Patla, D.A. Winter, M.G.Ishac, C.E. Little. Balance recoverty from medio-lateral perturbations of the upper body during standing. *Journal of Biomechanics*, 32:1149–1158, 1999

[15] M.C.Do, C.Schneider, R.K.Y.Chong. Factors influencing the quick onset of stepping follwing postural perturbation. *Journal of Biomechanics*, 32:795–802, 1999

[16] Elizabeth T. Hsiao, Stephen N. Robinovitch. Biomechanical influences on balance recovery by stepping. *Journal of Biomechanics*, 32:1099–1106, 1999

[17] Yi-Chung Pai and James Patton. Center of mass velocity-position predictions for balance control. *Journal of Biomechanics*, 30(4), 347–354, 1997

[18] Yi-Chung Pa, Kamran Iqbal. Simulated movement termination for balance recovery: can movement strategies be sought to maintain stability in the persence of slipping or forced sliding? *Journal of Biomechanics*, 32:779–786, 1999

[19] Elisabetta Papa, Aurelio Cappozzo. A telescopic inverted-pendulum model of the musculo-skeletarl system and its use for the analysis of the sit-to-stand motor task. *Jounal of Biomechanics*, 32:1205–1212, 1999

[20] K. Yamane and Y. Nakamura. Dynamics Filter — Concept and Implementation of On-Line Motion Generator for Human Figures. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp.688-695, 2000

[21] S. Kagami, F. Kanehiro, Y. Yamiya, M. Inaba and H. Inoue. Auto Balancer: An Online Dynamic Balance Compensation Scheme for Humanoid Robots. *Proceedings of The Fourth International Workshop on Algorithmic Foundations of Robotics (WAFR 2000)*, 2000