

---

# Balance Control via Adjusting CoM and CoP Reference Trajectories

Student: Chih-Hsiang Yu

Advisor: Dr. Wen-Chieh Lin

Institutes of Multimedia Engineering  
College of Computer Science  
National Chiao Tung University



## Abstract

In this thesis, we present an approach for simulated characters to perform specific motions. Using a momentum-based technique, our controller can maintain a simulated character's balance by controlling the character's center of mass (CoM) and center of pressure (CoP). In order to control the CoM and the CoP, the controller needs to follow reference trajectories of the CoM and the CoP. Based on the ground reaction forces, we propose a novel method to dynamically adjust the reference CoM for generating physically plausible and natural motions. In addition, due to lack of information about the CoP in reference motions, we determine the reference CoP according to the adjusted reference CoM by considering the moment of force acting at the supporting ankle joint. Our multi-objective optimization technique, including momentum and tracking objective terms, solves the optimal joint velocities for the next time step. Given a motion capture or key-frame data, our controller can maintain the full-body balance in presence of disturbances while tracking a specific motion simultaneously. We demonstrate our results that our control scheme is capable of standing and locomotion simulation in varying environments.

---

# Acknowledgments

I would like to thank my advisor, Professor Wen-Chieh Lin for his guidance, inspirations and encouragement. Thanks to my colleagues in our lab for their assistances and discussions. Lastly, I would like to thank my parents and my girlfriend for their love and support.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Related Work</b>	<b>4</b>
<b>3</b>	<b>Approach</b>	<b>8</b>
3.1	System overview . . . . .	8
3.2	Reference trajectory adjustment . . . . .	9
3.3	Balance mechanics based on momentum control . . . . .	14
3.4	Optimization . . . . .	16
3.4.1	Linear momentum objective . . . . .	17
3.4.2	Angular momentum objective . . . . .	18
3.4.3	Tracking objective . . . . .	20
3.4.4	Multi-objective optimization . . . . .	21
<b>4</b>	<b>Experimental Results</b>	<b>23</b>
4.1	Comparison . . . . .	23
4.2	Simulation in different environments . . . . .	25
4.3	Effects with the objective weights and the PD gains . . . . .	33
4.4	Discussion and limitation . . . . .	36
<b>5</b>	<b>Conclusion and Future Work</b>	<b>37</b>
	<b>Bibliography</b>	<b>38</b>

# List of Figures

2.1	Yin et al. use the finite state machine and the PD controller to develop a strategy for control of biped locomotion. (Image from [YLvdP07]) . . . . .	5
2.2	Kudoh et al. control the CoM and the ZMP (which is the same as CoP) to maintain standing balance. (Image from [KKI06]) . . . . .	6
2.3	Macchietto et al. control balance based on momentum to make sure that the simulated character can actively respond to the external disturbances. (Image from [MZS09]) . . . . .	7
3.1	System overview. . . . .	9
3.2	We adjust the reference CoM by considering the current positions of the supporting feet. . . . .	12
3.3	Determine the reference CoP. . . . .	13
3.4	Momentum control mechanics. . . . .	16
3.5	Layout of the optimization. . . . .	17
4.1	The parameters used in ODE simulation. . . . .	23
4.2	Comparing our method to those methods which the projection of the reference CoM is fixed at the center of the supporting region, each figure shows that our resulting motion is closer to the reference motion in frames. . . .	24
4.3	Directly following the trajectory of the reference CoM from motion data would cause the character tipping over. . . . .	25
4.4	The results are obviously different between the methods for locomotion simulation. Since the CoM in motion data is usually in front of the center of the supporting region in order to achieve a running motion, fixing the reference CoM may limit the character nearly in the original place. . . .	25



4.5	A result of maintaining balance on the floating box. The simulated character, whose thorax is pushed by external forces ((a)-(c) and (g)), can correct imbalance. . . . .	27
4.6	A result of maintaining balance on a dynamically shaking plane, which dynamically changes the inclination (between -20 and +20 degrees) during simulation time steps. . . . .	28
4.7	Given the reference frames from MOCAP data (a), the bottom three rows show the results that the simulated character performs repeated squatting motions with/without pushing by external forces in different environments. . . . .	29
4.8	The simulation results of walking on the flat ground ((a) and (c)) and on the floating box while the external forces push into the thorax of the character ((b) and (d)). . . . .	30
4.9	A simulation result of walking on the slope plane (15 degrees). . . . .	31
4.10	The simulation results of running on the flat ground, $\mu = 0.8$ ((a) and (c)) and on the slippery ground with the low-friction, $\mu = 0.08$ ((b) and (d)). . . . .	32
4.11	(a) and (b) show that the character loses the ability for maintaining balance and produces unnatural motions for locomotion without linear momentum term. Without angular momentum term, (c) and (d) show the character losses the ability to react against sudden rotation caused by external forces and the controller is more difficult to maintain the upper body upright. . . . .	34
4.12	Comparing the results between (a) the PD gains ( $k_t = 15$ , $d_t = 2$ ) and (b) the PD gains ( $k_t = 15$ , $d_t = 30$ ) for the desired accelerations of tracking, lower derivative gain (a) is more likely to be influenced by external forces. Our experimental result uses the PD gains ( $k_t = 15$ , $d_t = 8$ ) for the desired accelerations of tracking (c). . . . .	35
4.13	If the simulated character is pushed by forces in the air, it will correct imbalance after its supporting feet contacts with the ground. . . . .	36

# List of Tables

# 1 Introduction

Animating the behavior of simulated character is a challenging problem in computer animation. Unlike passive dynamic phenomena, such as cloth, smoke and fluids which can be simulated based on physics, it is more difficult to simulate a character's motion since it has to fulfill the control goals and handle the unexpected external disturbances in physical simulation environments. Due to lack of motor skills, it is very difficult to generate natural motions. Fortunately, the progress of data-driven techniques benefits style generation of a variety of realistic human motions. In order to taking advantages of the realism of motion data, it needs a robust controller to drive the simulated character to follow the reference motions. Most importantly, the controller should be capable of maintaining balance while encountering unexpected disturbance and producing physically plausible motions.

Most methods for balance control focus on a certain type of motion simulations, such as standing balance and locomotion. In traditional researches of balance control, it is important to control the center of mass (CoM); however, simply controlling the CoM for balance would cause the simulated character to tip over since rotational motion is not considered. By using the center of pressure (CoP) as an indicator for the tilting of the simulated character, researchers have used both the CoM and CoP to maintain balance recently [KKI02, KKI06, MZS09, WZ10]. Note that CoP is always identical to zero moment point (ZMP) in a balanced gait [VB04]. Nevertheless, controlling CoM and CoP needs reference trajectories from motion capture data. Although the reference trajectory of the CoM can be computed through forward kinematics, directly following it may not achieve good balance control because of model and environment differences. On the other hand, there also exist difficulties to obtain the reference trajectory of CoP due to lack of CoP information in motion data. Given reference trajectories of CoM and CoP,

it is difficult to build a physically-based controller to control the CoM and CoP of the simulated character, which one actively respond to the environments different from motion data. In order to achieve standing balance and locomotion simulation, we propose to use a physically-based controller to maintain the simulated character's balance by adjusting its CoM and CoP reference trajectories.

A common method for standing balance is to control the CoM by regulating the projection of a reference CoM at the center of the supporting region [AdSP07, MZS09, JYL09]. However, Pai and Patton [PP97] proposed that the position of the CoM is changed and may locate inside or outside of the supporting region while simulating walking, stepping and standing up motions. For standing balance and locomotion simulation, we adjust the reference trajectory of the CoM by considering the current positions of the supporting feet. In addition, we determine the reference CoP using the analytic relationship between the CoM and CoP described by Winter et al. [WPF<sup>+</sup>96]. In order to control the CoM and CoP, our architecture, using the linear momentum and angular momentum to control the balance of simulated character, is physically based on momentum proposed by Macchietto et al. [MZS09]. According to their approach, the linear momentum and the angular momentum can control the CoM and CoP simultaneously to maintain standing balance, we further adjust the reference CoM and the reference CoP thus the controller has ability to achieve not only standing balance but also locomotion simulation.

This thesis introduces a momentum-based approach to control the balance of a simulated character, which can respond appropriately to unexpected events and simultaneously track the reference motion. In order to adapt to different environments and motions, we dynamically adjust the CoM reference trajectory based on the ground reaction forces at each time step. Using the analytic relationship between the CoM and CoP described in [WPF<sup>+</sup>96], which considers the moment of force acting at the supporting ankle joint, we determine the reference CoP according to the adjusted reference CoM. Our controller uses a optimization technique to govern the CoM and CoP by the objective terms of the linear and angular momentum. Due to the balance control by the linear and angular momentum, the objective term of tracking then follows the trajectory of the reference motion to make sure that the simulated character can achieve the specified motions and has the ability to actively respond to external disturbances. Finally, our multi-objective optimization routine solves the optimal joint velocities for next time step.

Given a motion of standing or locomotion, our approach can produce physically plausible motions. Due to the robust control, our resulting simulation is able to achieve the full-body balance while following the trajectory of motion data in different environments, including dynamic planes and varying friction. In addition, without any precomputation, our simulation can be executed in real-time. The contributions of this thesis are as follows. We propose an approach, which dynamically adjusts the reference CoM trajectory, to produce more realistic balance behaviors in human motions. Furthermore, we also apply the relationship between the CoM and CoP to modify the reference CoP [WPF<sup>+</sup>96]. The proposed CoM and CoP adjustment approach can enhance the robustness of balance strategy based on momentum control.

The rest of the thesis is organized as follows. We first review related work in chapter 2. In chapter 3, we describe our approach in details. In chapter 4, we show our experimental results. Finally, we conclude the thesis in chapter 5.

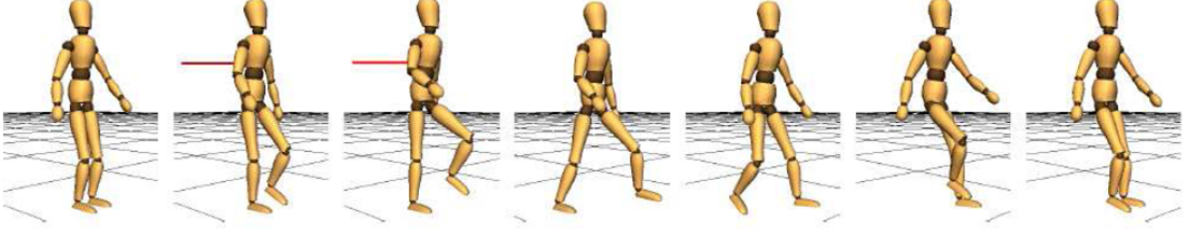


## 2 Related Work

Several approaches have been proposed to maintain the balance of a simulated character by combining data-driven techniques and physically based simulation [AdSP07, dSAP08a, DSAP08b, MZS09, YL10]. The main contribution of these works is that the simulated character has the ability to actively respond to environments that are different from those in MOCAP data. Our control method is similar to [MZS09], which optimizes multiple objectives based on controlling the momentum for standing balance. The main difference is that our approach would adjust the reference trajectories of the CoM and CoP, which affect the linear and angular momentum for balance. Comparing to other data-driven controllers described above, our work requires neither non-linear optimization, such as quadratic programming (QP), nor precomputation techniques, ensuring that our optimization problem can be solved in real-time.

To simulate locomotion, researchers have used the proportional-derivative (PD) controller to track human motion data, to produce the locomotion of humanoid character. Yin et al. [YLvdP07] reduced control design to few kinematics poses with the help of a balance strategy for walking and running (see Figure 2.1). Lee et al. [LKL10] controlled biped locomotion by modulating the reference trajectory. Coros et al. [CBvdP09] designed a balance controller for bipedal system allowing for the simulated character completing the skill and purpose, such as walking to a target location. Some researchers described the approaches to locomotion based on the features of movement [dLMH10] or the styles of reference motions [dSAP08a]. In addition, several researchers simulated the balance recovery responses to trips [SCCH09] and the protective responses to fall [PCC<sup>+</sup>09]. Most of above researchers proposed methods for locomotion using solely the PD controller, which rely heavily on motion data for producing physically plausible motions. In this thesis we propose an approach for locomotion balance based on momentum.

In order to design a balance controller, researchers have found that it is important

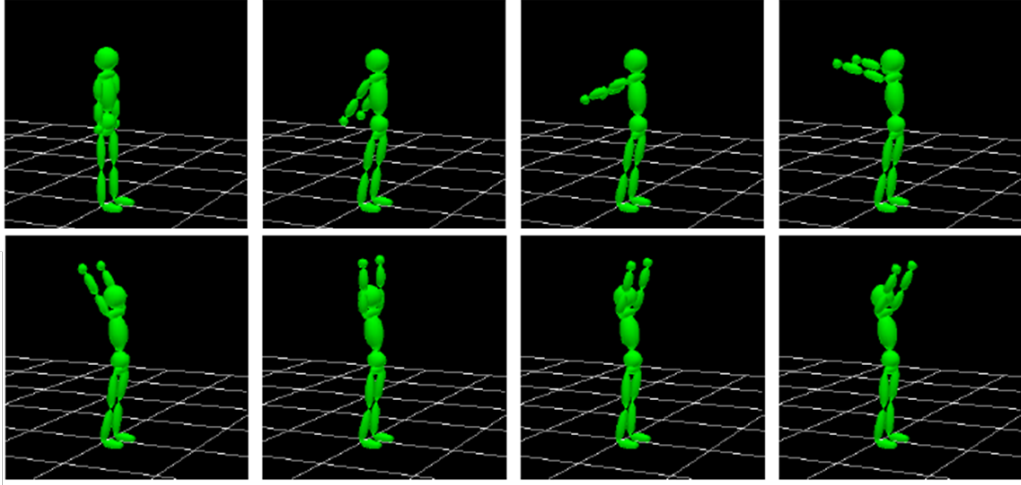


**Figure 2.1:** Yin et al. use the finite state machine and the PD controller to develop a strategy for control of biped locomotion. (Image from [YLvdP07])

to control the location of the CoM [ZH02, AdSP07, JYL09, MZS09, WZ10]. Several researchers take advantages of CoP for robust control [KKI02, KKI06, MZS09, WZ10] (see Figure 2.2) instead of controlling the CoM solely for balance [ZH02, AdSP07]. The approach described by Kudoh et al. [KKI02, KKI06] used the QP technique that is controlled via the CoM and zero moment point (ZMP), which is defined as CoP in this thesis, to maintain standing balance and respond to the external disturbance. There are few differences between our method and theirs. First, they do not control CoP precisely while they allow CoP to move freely within the supporting polygon. Second, there are no techniques to track the reference motion in their QP formulation, so it responds to the external disturbance without tracking the motion capture example. In addition, in contrast to the more complicated and iterative QP formulation that contains many inequality constraints, our momentum based optimization formulation can be executed more efficiently due to the simplified system of linear equation and the fewer constraints.

We can compute the reference trajectory of the CoM from motion data by forward kinematics, however, it is difficult to obtain the reference CoP due to lack of information about that in motion data. Several researchers have demonstrated that CoP is interrelated to CoM. Kudoh and his colleagues proposed that the position of ZMP (in our definition, CoP) is controlled along with the CoM and the linear acceleration of the CoM [KKI02, KKI06]. Winter and his colleagues used the inverted pendulum (IP) model to propose the relationship between these two commonly measured variables (the CoM and the CoP) and the horizontal acceleration of the CoM [WPF<sup>+</sup>96, WPP<sup>+</sup>98]. Their analysis shows that the projected difference between the CoP and CoM is indeed proportional to the horizontal acceleration of the CoM in both the sagittal and frontal planes of the simulated character. According to their analytic relationship, we calculate the reference CoP by considering the current CoM and the desired acceleration of the CoM on both the sagittal and frontal

planes of the simulated character.



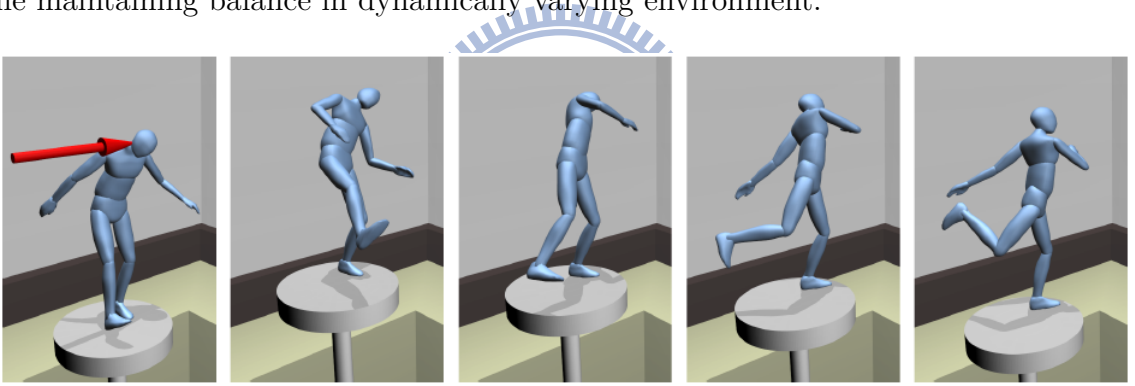
**Figure 2.2:** Kudoh et al. control the CoM and the ZMP (which is the same as CoP) to maintain standing balance. (Image from [KKI06])

In order to control the CoM and CoP of a simulated character, our controller utilizes the physical nature of momentum to follow the reference trajectories of the CoM and CoP. In robotics, many researchers have pointed out the importance of momentum. There are several advantages to maintain balance by momentum control. First, as the optimization expressions are reduced to only global momentum patterns, we can solve a much smaller set of unknowns over a much simpler set of constraints. Therefore, we can solve the optimization problems quickly and easily. Second, the momentum patterns can encode more about the natural motion than just physical correctness. For example, because it is complicate to model the natural motion of passive elements, such as muscle activities, we can encode the torque usage and the energy change in specific momentum patterns. Recent researches in computer animation have focused on simulation based on momentum. Some researchers used momentum as constraints to synthesize new motions by preserving the original dynamic behavior of the input motion [LP02, ALP06]. Macchietto et al. [MZS09] used the physical properties of momentum for balance control (see Figure 2.3). Wu and Zordan [WZ10] further introduced an approach to determine when and where to step based on momentum analysis. Sok et al. [SYLH10] presented a framework to edit the momentum and external forces in motion capture sequence for producing a variety of behaviors. One common theme in all of these robotics papers is that their methods are physically based on the feature of momentum.

A number of methods have begun to investigate using the angular momentum for



balance control because the angular momentum governs the CoP, which influences the rotation of simulated characters. Kajita et al. [KKK<sup>+</sup>03] generated the whole body motion of a humanoid character such that the resulted total linear and angular momentum become the specified values. Goswami and Kallem [GK04] suggested that the rate of change of angular momentum of the entire robot, computed at its center of mass, is such a measure. Popovic et al. [PHH04] presented a control strategy for biologically realistic walking based on the principle of spin angular momentum regulation. These methods have the same viewpoint that they use the angular momentum as a damper to avoid sudden rotation and reduce the external disturbance. Combining linear and angular momentum, which simultaneously control the CoM and CoP for maintaining balance, we further use the robust control scheme to reproduce different motions referring in motion data. In contrast to the previous works that used the momentum control only for balanced standing, we further use a momentum based method to simulate the human locomotion while maintaining balance in dynamically varying environment.



**Figure 2.3:** Macchietto et al. control balance based on momentum to make sure that the simulated character can actively respond to the external disturbances. (Image from [MZS09])

## 3 Approach

We now introduce our balance control approach, which is based on the law of momentum. At each simulation time step, our method adjusts the reference trajectories of the CoM and the CoP in order to make the controller reproduce the character’s motion by referencing the motion capture data (section 3.2). Based on the relationship between the linear and angular momentum, our balance controller uses the adjusted reference CoM and CoP to control a character’s momentum for balance (section 3.3). Finally, the optimal joint velocities, which make sure that the desired momentum changes are reconciled with the objective of tracking the reference motion, are computed (section 3.4).

### 3.1 System overview

Figure 3.1 shows an overview of our approach consisting of the following components:

- *Reference trajectory adjustment.* This component considers a character’s current state, including angular positions  $\theta$  and velocities  $\dot{\theta}$ , specified in the generalized coordinate as well as the trajectory of the character’s CoM computed from motion data, to adjust the reference CoM (denoted as  $c_r$ ) and the reference CoP (denoted as  $p_r$ ). We describe this component in section 3.2.
- *Optimization.* Once the reference CoM and CoP are adjusted, we compute the optimal joint velocities  $\dot{\theta}_*$  that minimize an objective function consisting of a linear momentum term, an angular momentum term and a tracking term. While the tracking term references the joint coordinates from the motion capture data, the linear momentum and angular momentum objectives use the information of the reference CoM and the reference CoP as the main balance strategy simultaneously. We demonstrate this component in section 3.4.

### 3.2 Reference trajectory adjustment

- *Dynamics simulation.* We use the Open Dynamics Engine (ODE) as the simulation environment. Given the optimal joint velocities which are computed from the *Optimization* component, the ODE simulates the character model under the influence of the external perturbations and the ground reaction forces (GRF), denoted  $f$ . And at each time step, we can obtain the current character's state from ODE.

At each time step, the controller (including the *Reference trajectory adjustment* and the *Optimization* components) yields the optimal joint velocities for the next time step based on the current simulating state of the character. The optimization algorithm solves the quadratic objective functions which has few constraints that should be subjected to, so that it solves the problems efficiently. Due to the GRF and the disturbances by external forces detected in *Dynamics simulation* component, the error from the optimal joint velocities are yielded. Since the optimization algorithm has no knowledge about the GRF and the external forces except for the gravity, it corrects the error by the feedback of the *Optimization* component. We demonstrate the balance mechanics in next section.

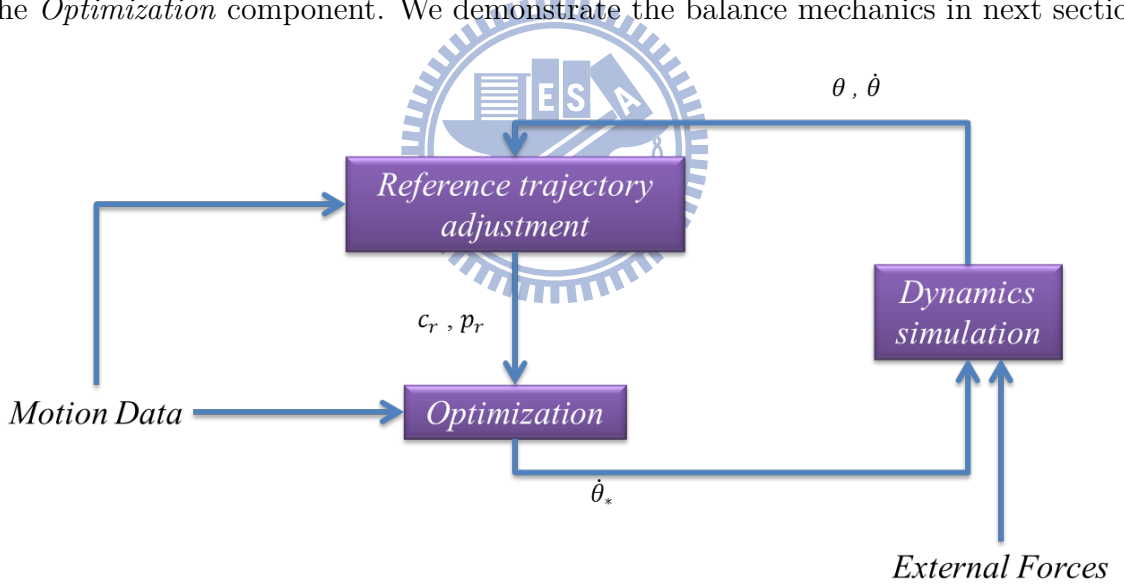


Figure 3.1: System overview.

## 3.2 Reference trajectory adjustment

As the data-driven techniques developed recently, some researchers have proposed methods that can achieve impressive balance control. In fact, most balance controllers track the projection of a reference CoM that is fixed at the center of the supporting area

### 3.2 Reference trajectory adjustment

[AdSP07, MZS09, JYL09]. This results in unnatural motions because a human may not always keep the projection of CoM at the center of supporting area in real cases. We can ameliorate the problem by directly tracking the reference position of the CoM obtained from motion data, expressed as

$$c_r(u) = \tilde{c}(u), \quad (3.1)$$

where  $u$  is the frame number of motion data to be tracked at the current simulation time step,  $\tilde{c}(u)$  is the position of the CoM at frame  $u$  in motion data. We compute the locations of the CoM at each frame by forward kinematics to reproduce the real trajectory of motion data. By following the reference trajectory of the CoM, the resulting motions are closely to the reference motions since the location of reference CoM can move during frames.

However, directly following the trajectory of the reference CoM from motion data usually does not work even if the balance controller is built based on the laws of physics. It is because of the following two reasons. First, there exists model differences between the simulated character and the MOCAP subject. Second, in interactive physical simulation, the real CoM trajectory may deviate from the reference CoM during simulation, especially in locomotion simulation in which the reference CoM is frequently outside of the supporting polygon. In that way, if the simulated character's motion is deviate from the reference trajectory due to the external disturbance or varying environment, the character would tip over.

Unlike standing balance, simulating for locomotion needs more careful control over the CoM since the reference CoM in locomotion simulation is not always within the supporting region. Without moving the reference CoM forward the supporting region, the character would step nearly in the original place; however, a character may risk in falling if the CoM is outside the supporting region. During simulation, the positions of bodies may deviate from the reference motion in presence of disturbances; however, the position of the supporting feet usually does not change while the character attempts to correct for imbalance. Based on the position of the supporting feet, we dynamically adjust the reference CoM at each time step according to the tracking frame. Since the supporting feet with larger ground reaction force is more influential to the reference CoM,

### 3.2 Reference trajectory adjustment

we propose an approach by considering the ground reaction forces to adjust the reference CoM. The position of the reference CoM can be expressed as

$$c_r(u) = w_{lf}(x_{lf} + \tilde{r}_{lf}(u)) + w_{rf}(x_{rf} + \tilde{r}_{rf}(u)), \quad (3.2)$$

where  $x_{lf}$  and  $x_{rf}$  are current positions of the left feet and the right feet (see Figure 3.2), and  $\tilde{r}_{lf}(u)$  and  $\tilde{r}_{rf}(u)$  are the vectors from the left feet and the right feet to the CoM of frame  $u$  in motion data.  $w_{lf}$  and  $w_{rf}$  are supporting weights, which depend on the supporting forces of the left feet and the right feet at each time step, we state the supporting weights as

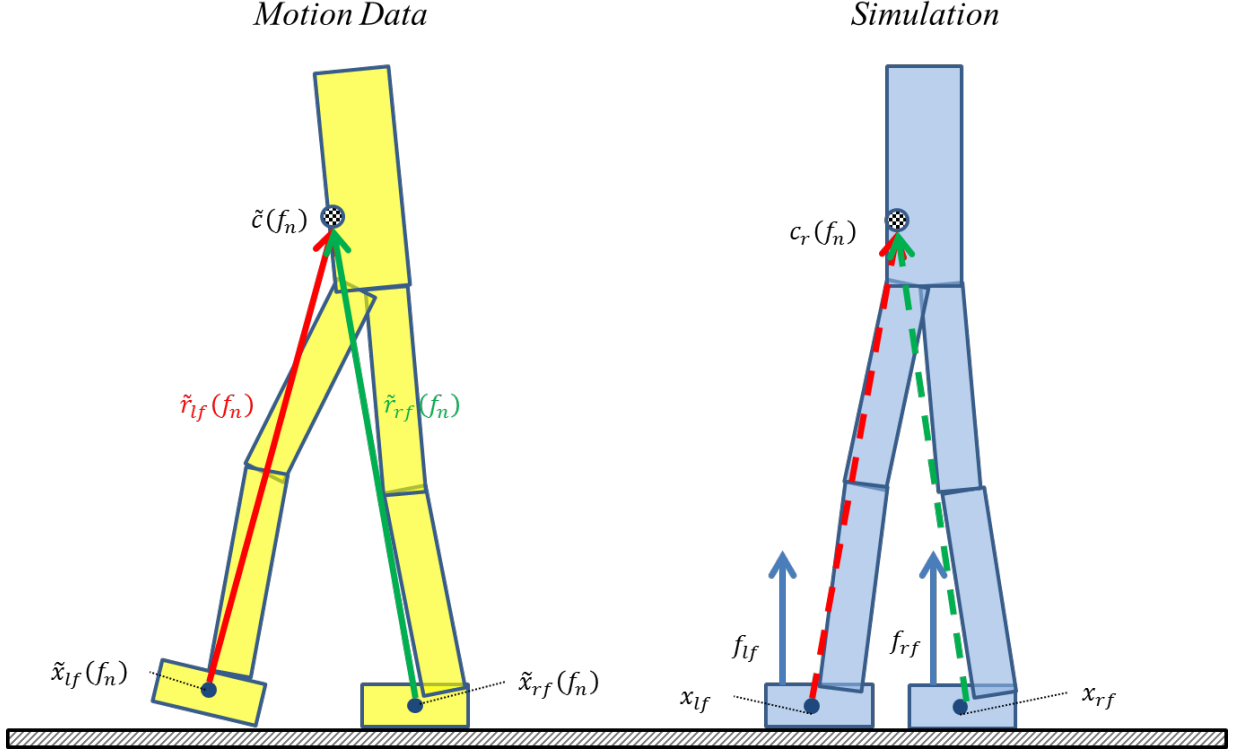
$$w_{lf} = \frac{f_{lf}}{f_{lf} + f_{rf}}, w_{rf} = \frac{f_{rf}}{f_{lf} + f_{rf}}, \quad (3.3)$$

where  $f_{lf}$  and  $f_{rf}$  are supporting forces of the left feet and the right feet in the normal direction of the supporting plane. Based on the ground reaction forces, our proposed approach dynamically revises the reference CoM at each time step by transforming the relative positions, between the supporting feet and the CoM, from motion data into simulations. Although regulating the projection of the reference CoM at the center of the supporting region is a choice for robust balance [MZS09], our method ensures the adjusted reference CoM closely to the CoM in motion data for better tracking within balance.

In contrast to the reference CoM, which can be computed through forward kinematics, it is difficult to obtain the reference CoP due to lack of information about the reference CoP in motion data. Macchietto et al. [MZS09] choose the center of supporting area as the reference CoP to prevent the CoP from moving too fast. They further suggest use a second optimization stage, which computes an ideal location of the reference CoP by ignoring the objective of angular momentum at the first stage. They then replace the reference CoP by the value of the desire CoP computed at the first stage for simulation at the second stage. However, the optimization is computationally expensive so it does not work in real-time. Moreover, they assume that the supporting area is not movable. Their approach would not work on those motions in which a character's supporting area changes rapidly, such as walking and running.

If the CoM is horizontally accelerated backward, then the projected CoP is ahead of the projected CoM and vice versa. We determine the reference CoP according to the

### 3.2 Reference trajectory adjustment



**Figure 3.2:** We adjust the reference CoM by considering the current positions of the supporting feet.

analysis described by Winter et al. [WPF<sup>+</sup>96]. They use the inverted pendulum (IP) model to describe the relationship among the CoM, CoP and the horizontal acceleration of the CoM. Based on an inverted pendulum model in a balance state, since the sum of the clockwise torque and counterclockwise torque acting about the supporting ankle joint is equal to the torque of the supporting ankle joint, they validate the relationship between the CoM and the CoP. They further demonstrate that the projected difference between the CoP and the CoM ( $p - c$ ) is proportional to the horizontal acceleration of the CoM on both the sagittal and frontal planes of the simulated character (see Figure 3.3). The reference CoP on the sagittal plane is expressed as

$$p_r^{sa} = c^{sa} - \left( \frac{I^{sa}}{Wh} \right) \ddot{c}_{des}^{sa}, \quad (3.4)$$

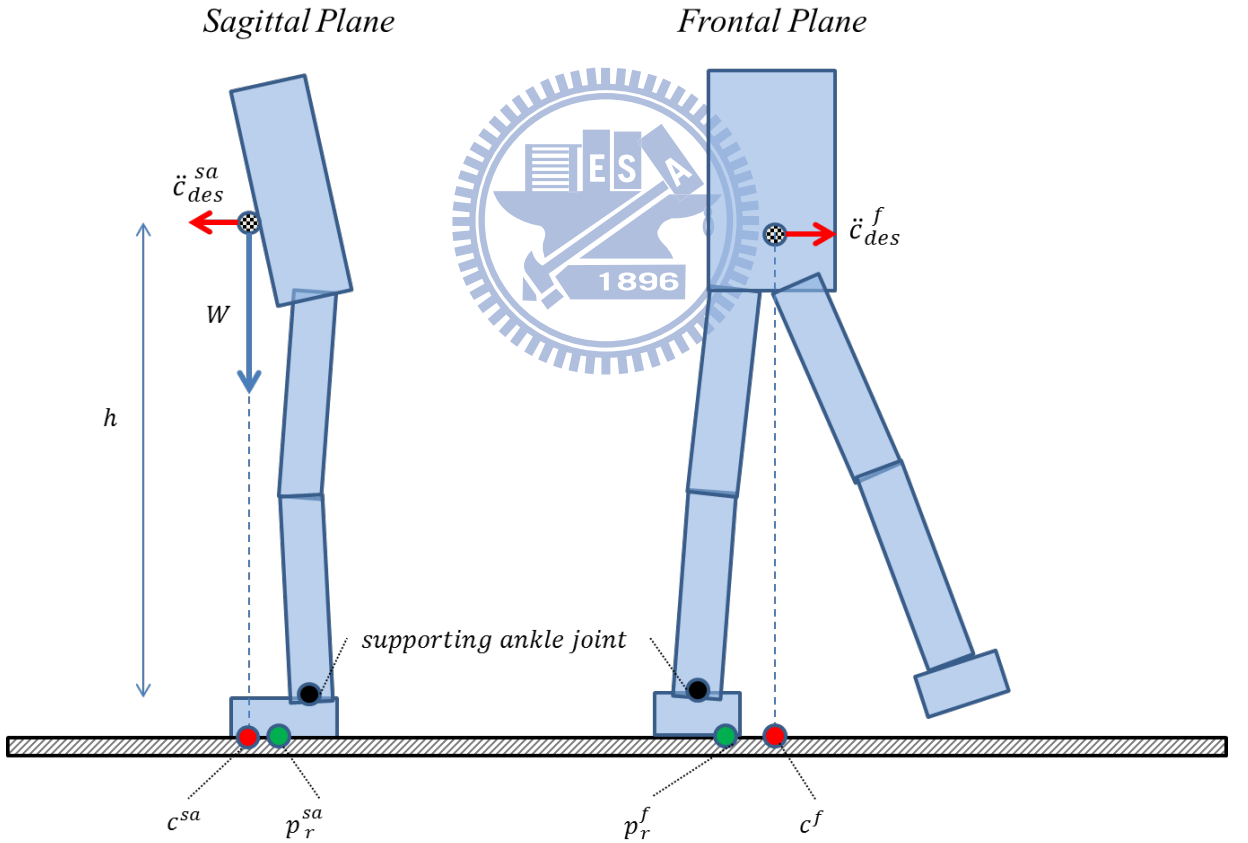
where  $c^{sa}$  is the projected position of the current CoM on the sagittal plane,  $\ddot{c}_{des}^{sa}$  is the desired acceleration of CoM in horizontal direction derived from Equation 3.13 (which is described in subsection 3.4.1) on the sagittal plane.  $I^{sa}$  is the total inertia of all body segments about the supporting ankle joint on the sagittal plane,  $W$  is the weight of the

### 3.2 Reference trajectory adjustment

character (minus the supporting feet). Note that we select the ankle joint with the larger supporting force as the supporting ankle joint. The reference CoP in the frontal plane is expressed as

$$p_r^f = c^f - \left( \frac{I^f}{Wh} \right) \ddot{c}_{des}^f, \quad (3.5)$$

where  $c^f$  is the projected position of the current CoM on the frontal plane,  $\ddot{c}_{des}^f$  is the desired acceleration of CoM in horizontal direction derived from Equation 3.13 (which is described in subsection 3.4.1) on the frontal plane, and  $I^f$  is the total inertia of all body segments about the supporting ankle joint on the frontal plane. In addition, it will cause the rotation of the character if the projection of the reference CoP is outside the supporting region. Therefore, we set a limitation to limit  $p_r$  within the supporting region.



**Figure 3.3:** Determine the reference CoP.

Our method adjusts the reference trajectory of the CoM and determine the location of the reference CoP for tracking the reference trajectory well. In addition, our method improve the control approach proposed by [MZS09] into a robust one that can be widely

### 3.3 Balance mechanics based on momentum control

used for simulation of locomotion. At each time step our controller updates the reference trajectory of the CoM and the CoP to change the reference goals in the objectives of linear and angular momentum, described in subsection 3.4.1 and subsection 3.4.2. We take the advantage of momentum-based balance control such that the simulated character can actively respond to the dynamic environments in real-time. Comparing our work to those regulating the projection of the reference CoM at the center of supporting region, we demonstrate our results can simulate more physically plausible motions than theirs in chapter 4.

## 3.3 Balance mechanics based on momentum control

We employ the technique based on momentum control proposed by Macchietto et al. [MZS09]. According to their approach, the control laws are designed to maintain balance. We can express the linear momentum of each body as

$$L_i = m_i v_i, \quad (3.6)$$

where  $m_i$  is the mass of the  $i$ th body and  $v_i$  is the velocity of the  $i$ th body. By computing the linear momentum of each body and summing them up, we can obtain the character's linear momentum,  $L = \sum_{i=1}^n L_i$ . The angular momentum of each body is expressed in

$$H_i = I_i w_i + r_i \times m_i v_i \quad (3.7)$$

where  $I_i$  is the  $3 \times 3$  inertia matrix of the  $i$ th body,  $w_i$  is the angular velocity of the  $i$ th body, and  $r_i$  is the vector from the  $i$ th body to the CoM. We can compute the character's angular momentum using  $H = \sum_{i=1}^n H_i$ .

The forces and torques applied to each body of the character are equivalent to the changes of the linear and angular momentum. Considering a standing character without external disturbances, the net force of all bodies, which is expressed as the derivative of the linear momentum  $\dot{L}$ , is equivalent to the sum of the ground reaction forces (GRF) and the gravitational force. Similarly, the net torque of all bodies, which is the derivative of angular momentum  $\dot{H}$ , is equivalent to the cross product of  $s = p - c$  and the ground reaction forces  $f$ , where  $p$  is the CoP and  $c$  is the CoM. We summarize the equations of



### 3.3 Balance mechanics based on momentum control

the linear and angular momentum derivatives as

$$\dot{L} = mg + f \quad (3.8)$$

$$\dot{H} = s \times f, \quad (3.9)$$

where  $m$  is the total mass of all body segments, i.e.,  $m = \sum_{i=1}^n m_i$ ,  $g$  is the gravitational acceleration, and  $f$  is the ground reaction forces.

As described above, the character's linear momentum, expressed in

$$L = \sum_{i=1}^n L_i = \sum_{i=1}^n m_i v_i = m\dot{c}, \quad (3.10)$$

where  $\dot{c}$  is the linear velocity of the CoM. If we can control the linear momentum effectively, it is the same as controlling the linear velocity of the CoM. The derivative of the linear momentum is computed by

$$\dot{L} = \sum_{i=1}^n m_i a_i = m\ddot{c}, \quad (3.11)$$

where  $\ddot{c}$  is the linear acceleration of the CoM. By observing Equation 3.8: without any external perturbations other than  $f$ , we obtain the GRF from the derivative of the linear momentum. Equation 3.9 demonstrates how to control the derivative of the angular momentum: with the ground reaction forces from Equation 3.8 and the known information of the current CoM (which is derived from the simulation state), we control the derivative of the angular momentum by specifying the reference CoP (section 3.2). Figure 3.4 shows the process how we control the linear and angular momentum through controlling over the reference CoM and the reference CoP.

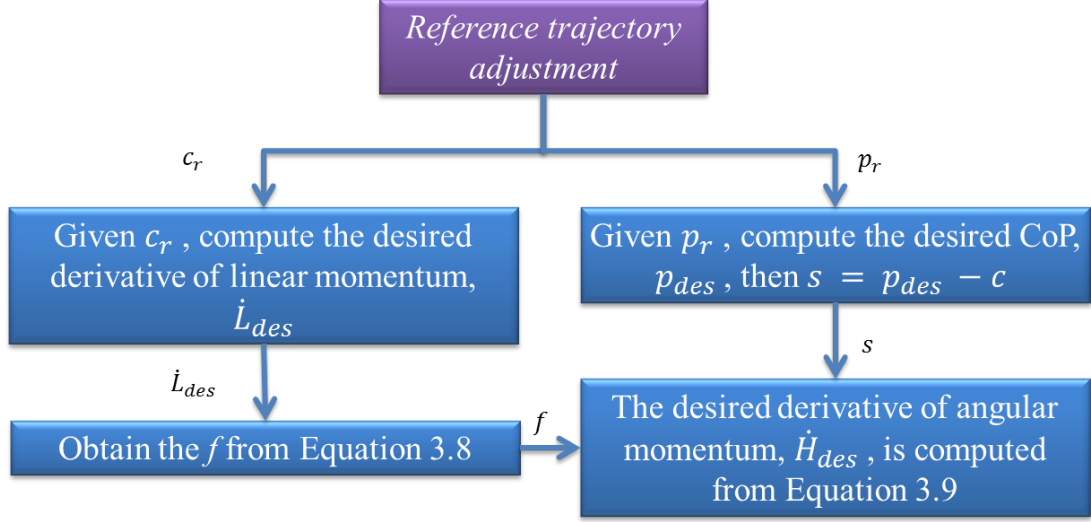


Figure 3.4: Momentum control mechanics.

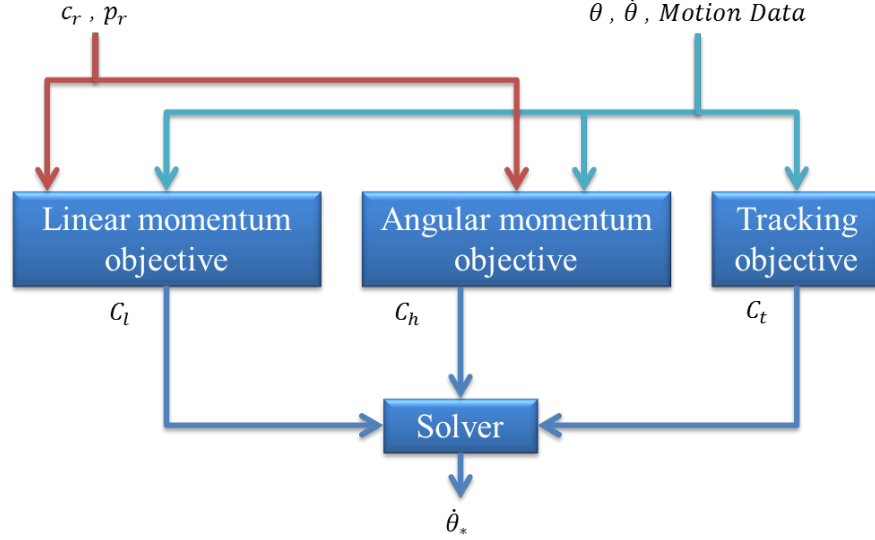
### 3.4 Optimization

The core of our method is the formulation of the optimization problem to allow the simulated character to maintain balance. The optimization process is sketched in Figure 3.5. After we obtain the reference CoM and the reference CoP from section 3.2, the multi-objective optimization solves the optimal joint velocities  $\dot{\theta}_*$  at each time step. Our objective function consists of the linear momentum  $C_l$ , angular momentum  $C_h$  and tracking terms  $C_t$ . While the linear and angular momentum terms govern the positions of the projected CoM and the CoP to maintain balance, the tracking term tries to follow the reference motion as closely as possible.

We formulate the multi-objective optimization problem as

$$\begin{aligned} \min_{\dot{\theta}} \quad & \beta_l C_l + \beta_h C_h + \beta_t C_t \\ \text{subject to : } & J_{sup}(\theta) \dot{\theta} = \hat{v}_{sup}, \end{aligned} \quad (3.12)$$

where  $\hat{v}_{sup}$  is the spatial velocities of the supports, usually chosen to be zero, and  $J_{sup}$  is the Jacobian matrix related to the support ( $J_{sup}$  is the sub-matrix of Jacobian matrix,  $J$ , pertaining to the supporting body). We set the constraints to prevent the supporting feet from accelerating.  $\beta_l$ ,  $\beta_h$  and  $\beta_t$  are the objective weights of the linear momentum, angular momentum and tracking terms, respectively. We describe each term for detail in following parts.



**Figure 3.5:** Layout of the optimization.

### 3.4.1 Linear momentum objective

Our goal is to control the position of the CoM so that the projected CoM is close to the projection of the reference CoM (which is described in section 3.2). The desired acceleration of the CoM is defined as follows

$$\ddot{c}_{des} = k_l (c_r - c) + d_l (\dot{c}_r - \dot{c}), \quad (3.13)$$

where  $k_l$  is the proportional gain that provides an instantaneous response to the control error, and  $d_l$  is the derivative gain that acts on the change of the control error.  $\dot{c}$  is the temporal derivative of  $c$ .  $c_r$  is the reference CoM from section 3.2, and  $\dot{c}_r$  is the derivative of the reference CoM over time steps. Similar to the method described by Macchietto et al. [MZS09], we only interest in the two orthogonal axes perpendicular to the gravitational axis in Equation 3.13.

Comparing with [MZS09], we can find the differences. First, they choose the  $\dot{c}_r$  to be zero because they only track a single posture or the motion which the CoM does not move. Second, they choose the reference CoM as the center of the supporting polygon, their approach may not be able to track the motions in which the trajectories of the CoM are movable. Therefore, their approach can only track the motions in which the supporting area of simulated character are regular at the same place. Since most motions are movable, we improve the method by adjusting the reference CoM at each time step

### 3.4 Optimization

described in section 3.2. Due to the change of the reference CoM, the  $\dot{c}_r$  will not be zero at each time step.

As we described in section 3.3, it is the same as controlling the linear velocity of the CoM if we can control the linear momentum efficiently. The desired linear momentum can be expressed in

$$L_{des} = m\dot{c}_{des}, \quad (3.14)$$

where  $\dot{c}_{des}$  is obtained by integrating  $\ddot{c}_{des}$  over time.

For optimization, the linear momentum objective  $C_l$  is expressed in

$$C_l = \|L_{des} - L\|^2 = \|L_{des} - MJ\dot{\theta}\|^2, \quad (3.15)$$

where  $L_{des}$  is computed from Equation 3.14, and  $J$  is the Jacobian matrix, defined as

$$J(\theta) = \begin{bmatrix} \frac{\partial \hat{x}_1}{\partial \theta_1} & \cdots & \frac{\partial \hat{x}_1}{\partial \theta_k} \\ \vdots & \ddots & \vdots \\ \frac{\partial \hat{x}_n}{\partial \theta_1} & \cdots & \frac{\partial \hat{x}_n}{\partial \theta_k} \end{bmatrix} \quad (3.16)$$

where  $\hat{x}_i \in \mathbb{R}^3$  is the CoM of  $i$ th body, and  $k$  is the total degrees of freedom of all joints. The joint velocities  $\dot{\theta}$  are chosen by optimizer.  $M$  is a  $3 \times 3n$  matrix for  $n$  bodies, and it is defined as

$$M = [M_1 \ \dots \ M_n], \quad (3.17)$$

where  $M_i$  is a  $3 \times 3$  diagonal matrix,

$$M_i = \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{bmatrix}. \quad (3.18)$$

#### 3.4.2 Angular momentum objective

While the linear momentum objective optimization controls the CoM, we use the angular momentum to control the CoP. Similar to Equation 3.13, we state the desired acceleration

### 3.4 Optimization

to control the CoP as the following:

$$\ddot{p}_{des} = k_h (p_r - p) + d_h (\dot{p}_r - \dot{p}), \quad (3.19)$$

where  $k_h$  and  $d_h$  are the proportional and derivative gain constants.  $p_r$  and  $p$  are the reference CoP and current CoP, respectively.

As we described in Equation 3.9, the desired derivative of the angular momentum is computed by

$$\dot{H}_{des} = (p_{des} - c) \times (\dot{L}_{des} - mg), \quad (3.20)$$

where  $p_{des}$  is the desired CoP, obtained by integrating  $\ddot{p}_{des}$  over time.  $\dot{L}_{des}$  is the desired derivative of linear momentum obtained from Equation 3.13 by multiplying  $m$ ,

$$\dot{L}_{des} = m\ddot{c}_{des} \quad (3.21)$$

Similar to the linear momentum term, we formulate the angular momentum objective  $C_h$  in

$$C_h = \|H_{des} - H\|^2 = \|H_{des} - PJ\dot{\theta}\|^2, \quad (3.22)$$

where  $H_{des}$  is derived by integrating  $\dot{H}_{des}$  (Equation 3.20) over time. The joint velocities  $\dot{\theta}$  are chosen by optimizer. The calculation of  $P$  can be expressed as

$$P = \begin{bmatrix} U & V \end{bmatrix}. \quad (3.23)$$

Let

$$U = \begin{bmatrix} m_1 [r_1]_{\times} & \dots & m_n [r_n]_{\times} \end{bmatrix} \quad (3.24)$$

$$V = [I_1 \dots I_n], \quad (3.25)$$

### 3.4 Optimization

the cross product operator  $[\cdot]_{\times}$  transforms the operand into  $3 \times 3$  skew-symmetric matrix (i.e.  $[a]_{\times} b = a \times b$ ),  $m_i$  is the mass of the  $i$ th body,  $r_i$  is the vector from the  $i$ th body to the CoM, and  $I_i$  is the  $3 \times 3$  inertia matrix of the  $i$ th body.

#### 3.4.3 Tracking objective

The tracking objective optimization attempts to track the trajectory of motion data. In our implementation, motion data obtained from the CMU Mocap database. While the linear and angular momentum objectives govern the balance rules, the tracking objective is used to reproduce the style of motion data. Based on the tracking control described by Abe et al. [AdSP07], the desired accelerations are expressed in

$$\ddot{\theta}_{des} = k_t (\theta_m - \theta) + d_t (\dot{\theta}_m - \dot{\theta}) + \ddot{\theta}_m, \quad (3.26)$$

where  $k_t$  and  $d_t$  are tracking stiffness and damping. We usually set  $d_t = 2\sqrt{k_t}$  similar to [AdSP07].  $\theta_m$ ,  $\dot{\theta}_m$  and  $\ddot{\theta}_m$  are the positions, velocities and accelerations from motion data. We can use the same equation to track a single posture by setting  $\dot{\theta}_m$  and  $\ddot{\theta}_m$  to zero.

We can set the stiffness value according to tracking different motions. High stiffness value is used to closely track motion data despite external forces. This may produce unnatural motion. Low stiffness value produce more compliant and usually realistic motions, especially when the simulation environment is different to the environment from motion data. The tracking objective  $C_t$  is defined as

$$C_t = \|W_t (\dot{\theta}_{des} - \dot{\theta})\|^2, \quad (3.27)$$

where  $W_t$  is a user-specified diagonal weighting matrix. It is used to allow for additional tracking emphasis or de-emphasis on particular joints.  $\dot{\theta}_{des}$  is the desired velocities obtained by integrating  $\ddot{\theta}_{des}$  over time. For example, to adapt to the uneven ground, we can reduce the weighting of the ankle joints so that it tracks motion data weakly than others. The joint velocities  $\dot{\theta}$  are chosen by optimizer.

### 3.4.4 Multi-objective optimization

In our implementation,  $C_l$ ,  $C_h$  and  $C_t$  are represented in the quadratic form:

$$\beta_l C_l = \beta_l \|L_{des} - MJ\dot{\theta}\|^2 \quad (3.28)$$

$$\beta_h C_h = \beta_h \|H_{des} - PJ\dot{\theta}\|^2 \quad (3.29)$$

$$\beta_t C_t = \beta_t \|W_t (\dot{\theta}_{des} - \dot{\theta})\|^2. \quad (3.30)$$

We now demonstrate how to use a system of linear equation to solve the multi-objective optimization problem. Let

$$A_l = -MJ, \quad B_l = L_{des} \quad (3.31)$$



$$A_h = -PJ, \quad B_h = H_{des} \quad (3.32)$$

$$A_t = -W_t, \quad B_t = W_t \dot{\theta}_{des}, \quad (3.33)$$

Equation 3.28, Equation 3.29 and Equation 3.30 can be rewritten as

$$\beta_l C_l = \beta_l \left( \dot{\theta}^T A_l^T A_l \dot{\theta} + \dot{\theta}^T A_l^T B_l + B_l^T A_l \dot{\theta} + B_l^T B_l \right) \quad (3.34)$$

$$\beta_h C_h = \beta_h \left( \dot{\theta}^T A_h^T A_h \dot{\theta} + \dot{\theta}^T A_h^T B_h + B_h^T A_h \dot{\theta} + B_h^T B_h \right) \quad (3.35)$$

### 3.4 Optimization

$$\beta_t C_t = \beta_t \left( \dot{\theta}^T A_t^T A_t \dot{\theta} + \dot{\theta}^T A_t^T B_t + B_t^T A_t \dot{\theta} + B_t^T B_t \right). \quad (3.36)$$

As we transform each objective term into the quadratic form, our multi-objective optimization problem (described in Equation 3.12) is equivalent to solve the linear equation, which ensures the sum of the derivative of each quadratic form being zero. Thus, we solve  $\dot{\theta}$  in the linear equation, expressed in

$$2 \left( \beta_l A_l^T A_l + \beta_h A_h^T A_h + \beta_t A_t^T A_t \right) \dot{\theta} + 2 \left( \beta_l A_l^T B_l + \beta_h A_h^T B_h + \beta_t A_t^T B_t \right) = 0, \quad (3.37)$$

derived from the sum of the derivatives of Equation 3.34, Equation 3.35 and Equation 3.36. Let

$$A = 2 \left( \beta_l A_l^T A_l + \beta_h A_h^T A_h + \beta_t A_t^T A_t \right), \quad B = -2 \left( \beta_l A_l^T B_l + \beta_h A_h^T B_h + \beta_t A_t^T B_t \right), \quad (3.38)$$

the constraint optimization problem described in Equation 3.12 can be solved by solving the following system of linear equations,

$$\begin{bmatrix} A \\ J_{sup}(\theta) \end{bmatrix} \dot{\theta} = \begin{bmatrix} B \\ \hat{v}_{sup} \end{bmatrix}. \quad (3.39)$$

Comparing our structure of optimization to the one described by Abe et al. [AdSP07], our formulation can be executed more efficiently due to the simplified linear equation system and the fewer constraints since they use the inequality constraints which requires a more complicated and iterative quadratic programming (QP) method to solve it.



## 4 Experimental Results

In this chapter, we demonstrate our results of comparison with the approach proposed by Macchietto et al. [MZS09] and simulation of various environments. Comparing our results to [MZS09] which regulates the projection of the CoM at the center of the supporting region, the results show that our method is better for tracking motions (section 4.1). In addition, our work can achieve simulation for different motions in varying environments (section 4.2). In section 4.3, we demonstrate the effects by changing the objective weights for optimization and the PD gains described in each objective term. Finally, we discuss the limitations of our approach in section 4.4.

Our simulation is performed by the Open Dynamics Engine (ODE) version 0.10.1. Figure 4.1 shows the parameters used in ODE simulation. All the experiments were run on a 2.93 GHz Intel Core2 Duo CPU E7500 computer with 3.25GB of RAM. The computation of each simulation loop is 0.038 seconds.

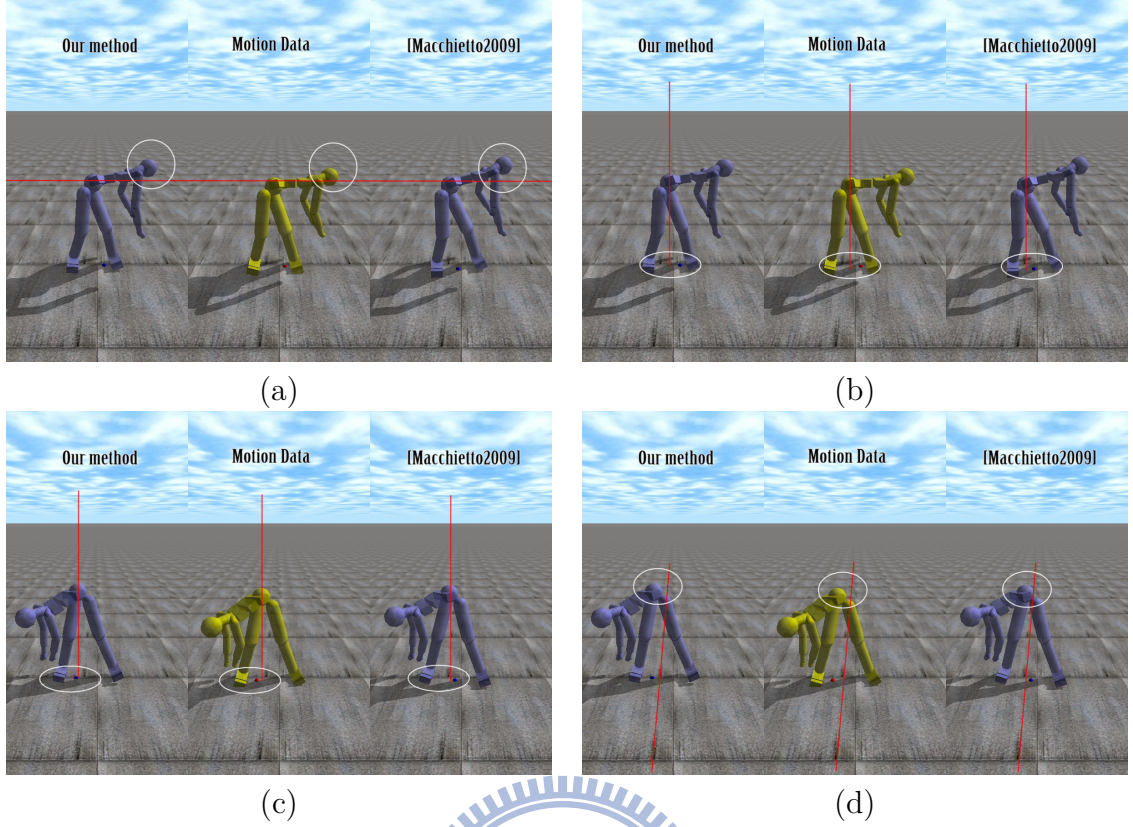
ODE parameters	value
Frictional coefficient	0.8
Time step	0.017 (s)
Gravity	9.8 (m/s <sup>2</sup> )

**Figure 4.1:** The parameters used in ODE simulation.

### 4.1 Comparison

Since our method is similar to the one proposed by Macchietto et al. [MZS09], we first compare our results to theirs in this section. Figure 4.2 shows the results using our method (left) and theirs (right) while tracking the same bending motion (middle). Because their

## 4.1 Comparison

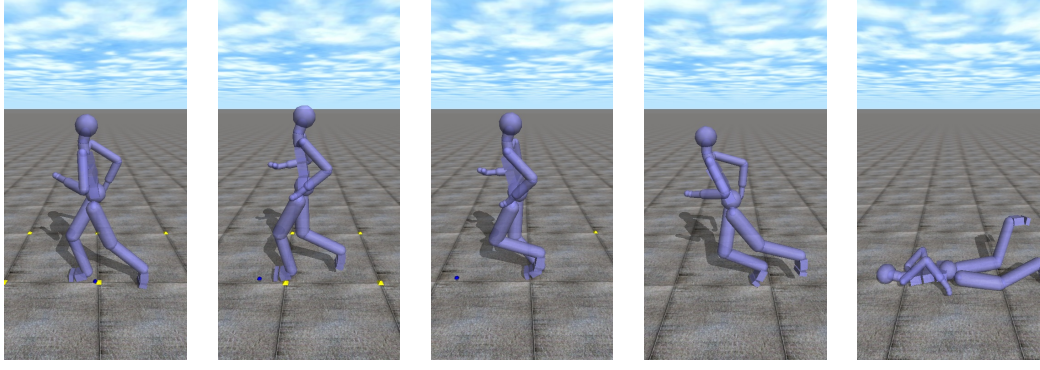


**Figure 4.2:** Comparing our method to those methods which the projection of the reference CoM is fixed at the center of the supporting region, each figure shows that our resulting motion is closer to the reference motion in frames.

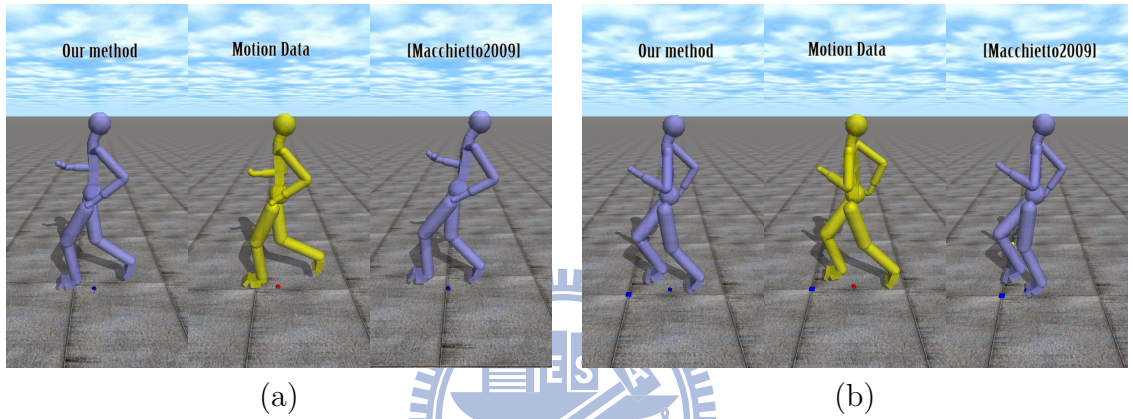
projection of the reference CoM is regulated at the center of the supporting region, their results deviate from the reference motion when the projection of the CoM in motion data is far away from the center of the supporting region. Without external forces, our resulting motion is closer to the reference motion than theirs.

In addition, to simulate motions for locomotion, it is important to control the reference CoM, which is an indicator for the location of a simulated character. Directly following the trajectory of the reference CoM from motion data usually does not work due to model and environment differences (see Figure 4.3). Regulating the projection of the reference CoM at the center of supporting region would limit the movement of the simulated character, and may produce unnatural motions since the CoM in motion data is varying (even outside the supporting region). Figure 4.4 shows a comparison of our results, original motion, and Macchietto et al. results. One can observe that our method can make the simulated character move forward, while theirs limits the character nearly in the original place.

## 4.2 Simulation in different environments



**Figure 4.3:** Directly following the trajectory of the reference CoM from motion data would cause the character tipping over.



**Figure 4.4:** The results are obviously different between the methods for locomotion simulation. Since the CoM in motion data is usually in front of the center of the supporting region in order to achieve a running motion, fixing the reference CoM may limit the character nearly in the original place.

## 4.2 Simulation in different environments

Since the environments of simulation and motion data are usually different, it is difficult for a controller to reproduce motions in different environments. Through testing the ability of our character to simulate on different planes and enable external forces, our experimental results show extreme balance reactions. Given a motion capture or key-frame data, the standing character can actively respond to external perturbations and simultaneously track the reference motion in different environments. Figure 4.5 and Figure 4.6 show a simulated character can keep balance on a floating box or on a dynamically shaking plane while tracking the reference posture. Tracking the reference frames from the CMU Mocap database Figure 4.7 (a), Figure 4.7 (b)-(d) are the simulation results in different environments. Each result shows that the simulated character can remain balance to

## 4.2 Simulation in different environments

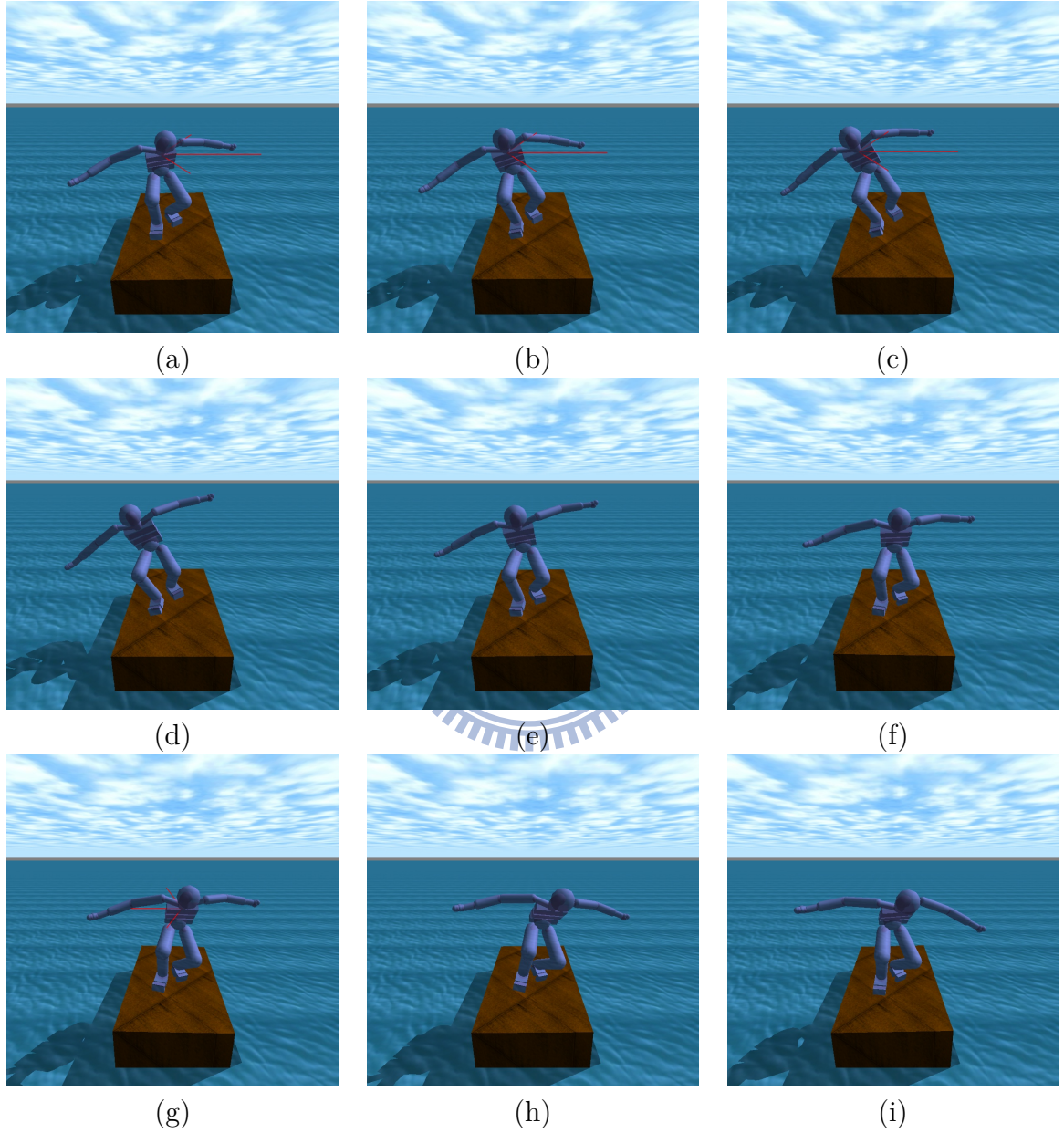
achieve the specified motion even in a dynamic environment.

Unlike standing balance, simulating for locomotion needs to move the reference CoM outside of the supporting region temporarily. Therefore, the simulated character probably falls without carefully control the reference CoM. Given a motion of walking or running, we further show the results by following the adjusted reference trajectories of the CoM and the CoP. Figure 4.8 shows the simulation results of walking, and Figure 4.9 shows the result that the simulated character can walk on the slope plane. In addition, to simulate a motion of locomotion which the movement of the CoM is fast, such as running, it needs a robust control scheme to determine the adjusted reference trajectory of the CoM, or it would be suddenly fail after time steps due to the large difference between the reference CoM and the current CoM. Figure 4.10 shows the simulation result for running on the slippery ground.



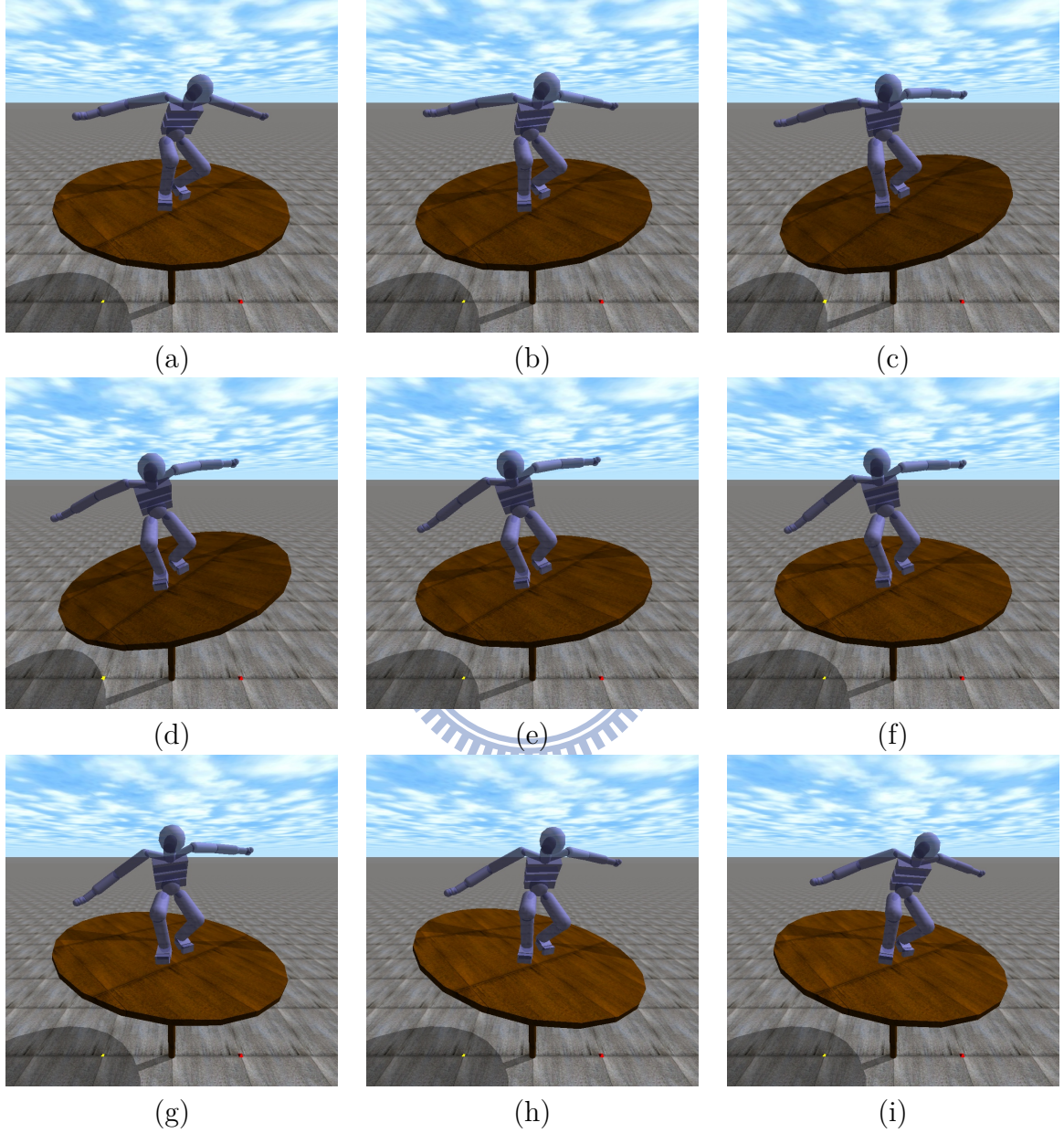


## 4.2 Simulation in different environments



**Figure 4.5:** A result of maintaining balance on the floating box. The simulated character, whose thorax is pushed by external forces ((a)-(c) and (g)), can correct imbalance.

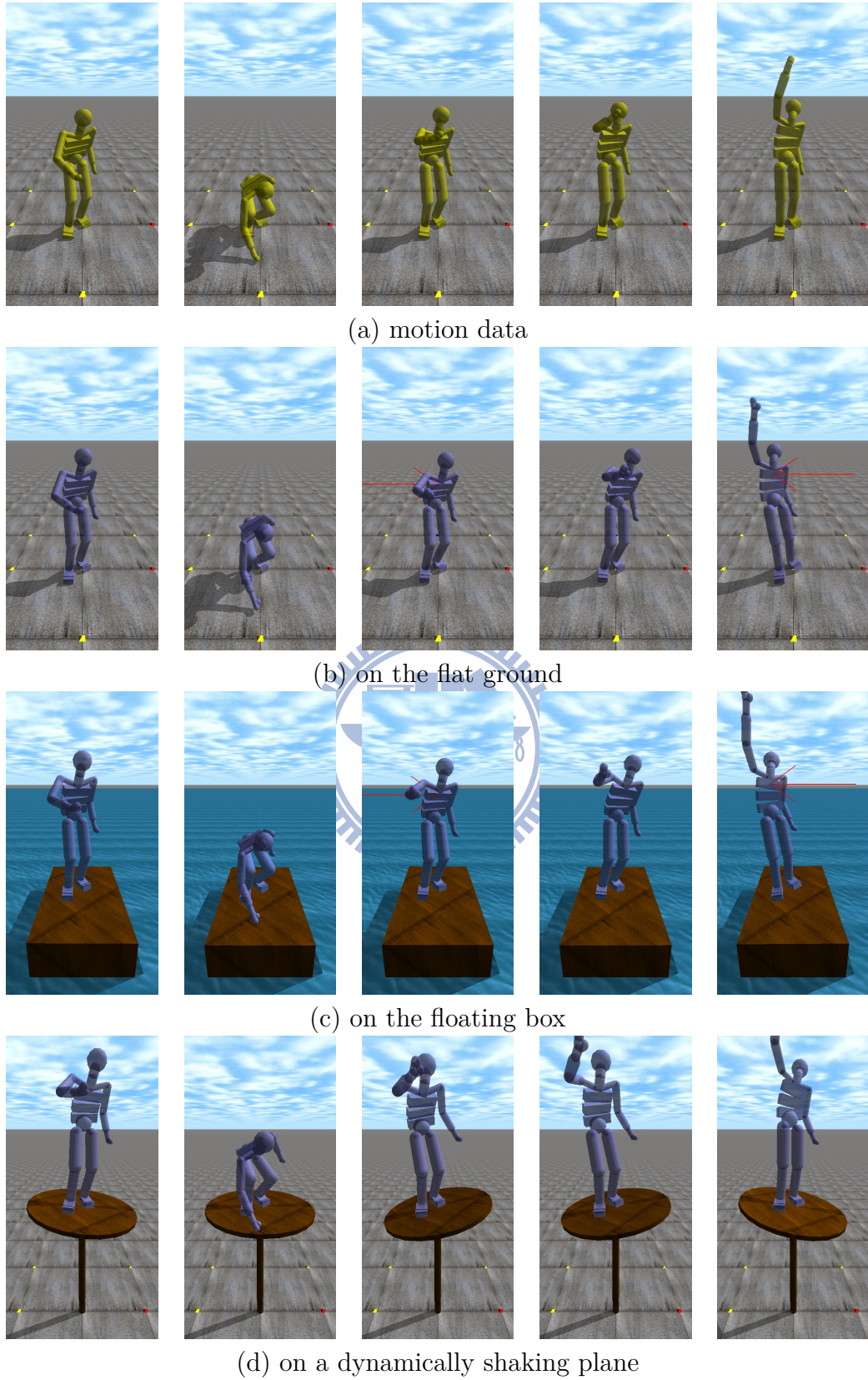
## 4.2 Simulation in different environments



**Figure 4.6:** A result of maintaining balance on a dynamically shaking plane, which dynamically changes the inclination (between  $-20$  and  $+20$  degrees) during simulation time steps.

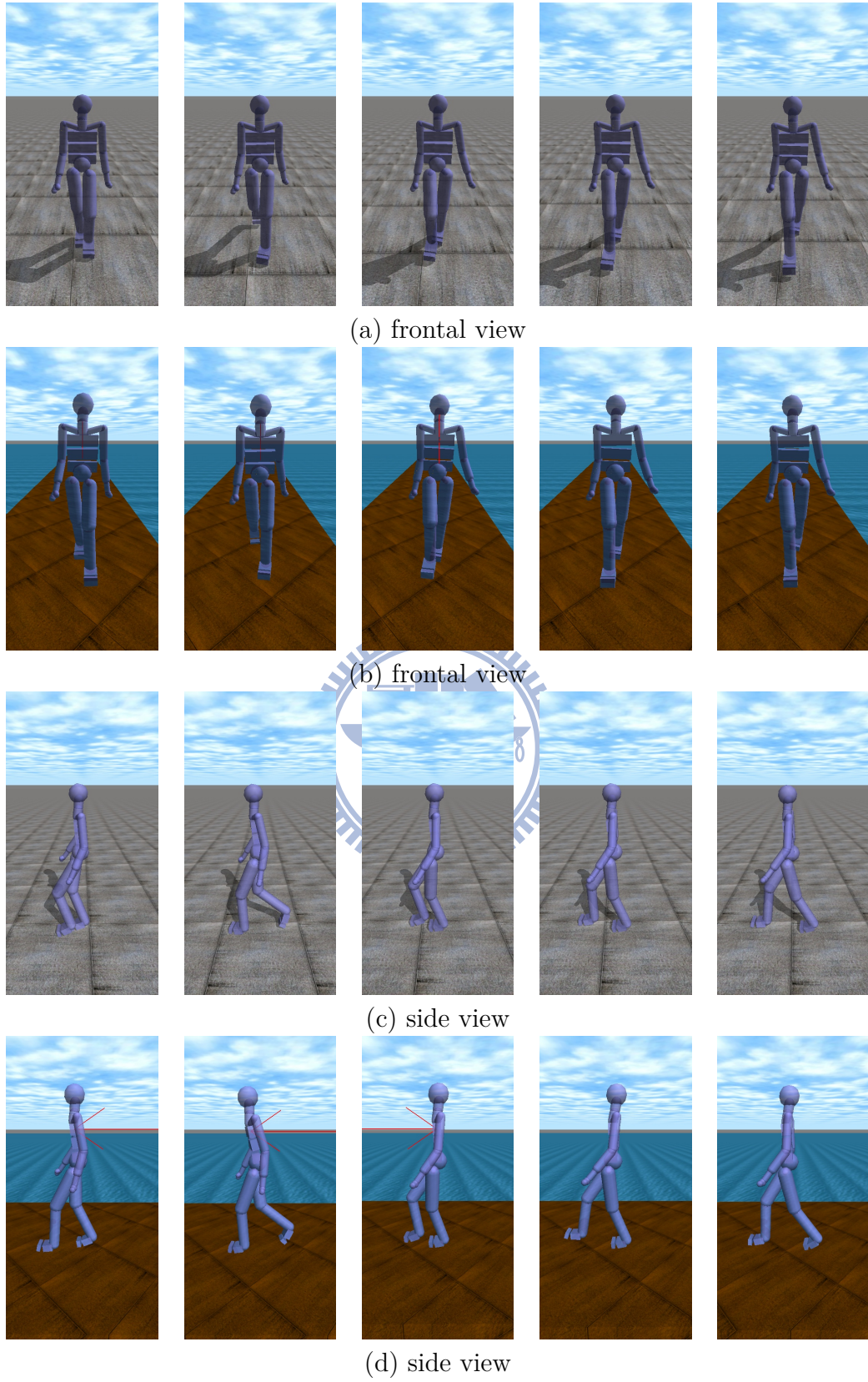


## 4.2 Simulation in different environments



**Figure 4.7:** Given the reference frames from MOCAP data (a), the bottom three rows show the results that the simulated character performs repeated squatting motions with/without pushing by external forces in different environments.

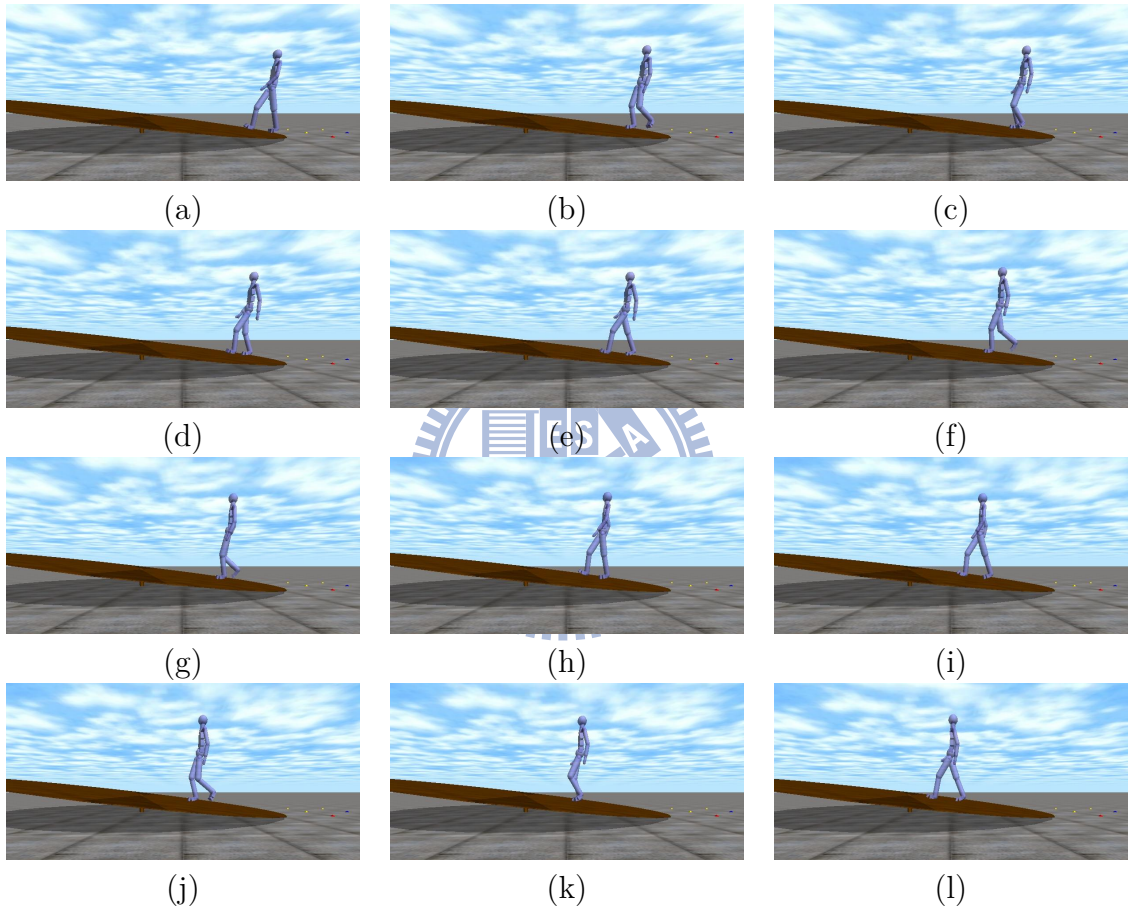
## 4.2 Simulation in different environments



**Figure 4.8:** The simulation results of walking on the flat ground ((a) and (c)) and on the floating box while the external forces push into the thorax of the character ((b) and (d)).

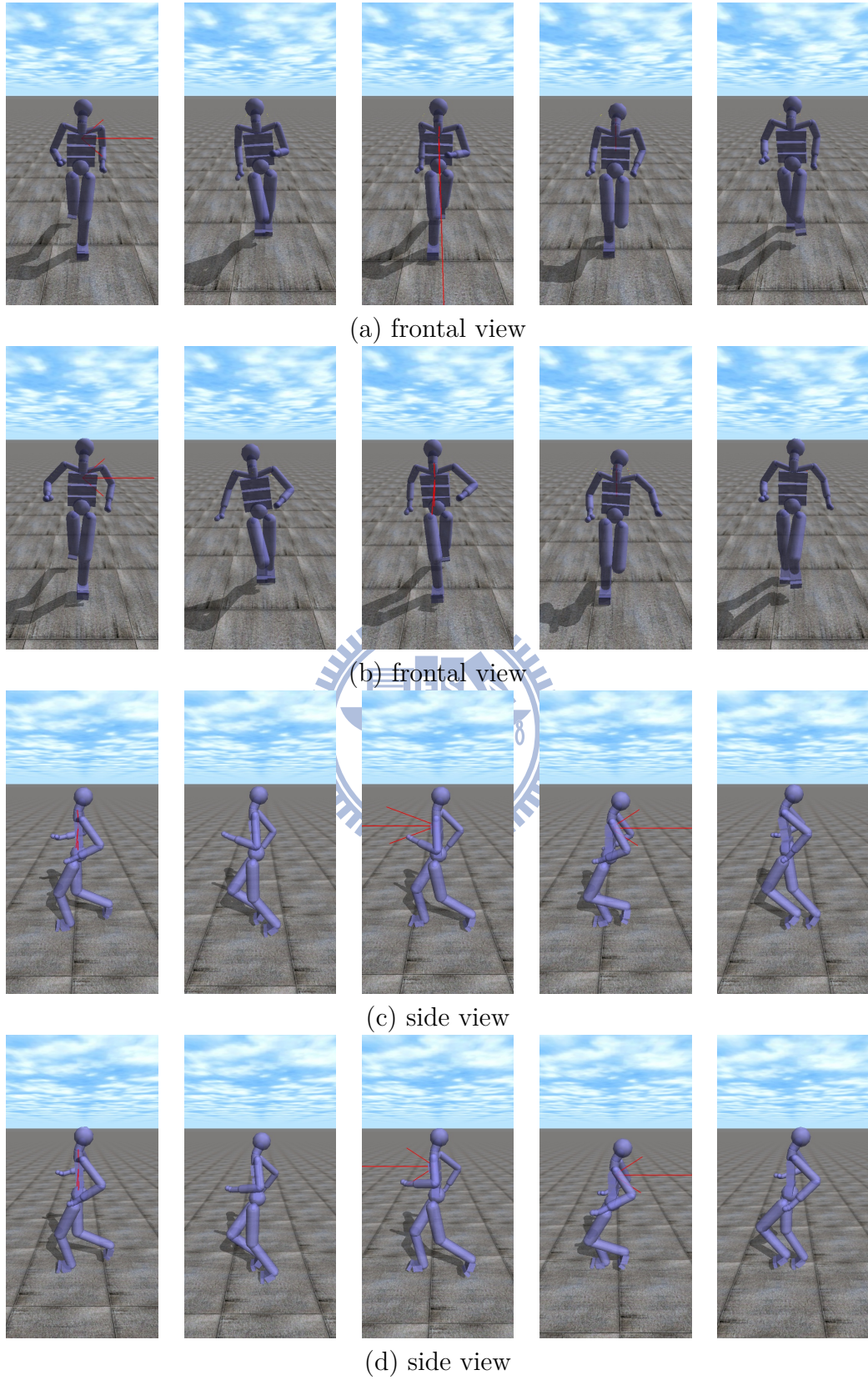


## 4.2 Simulation in different environments



**Figure 4.9:** A simulation result of walking on the slope plane (15 degrees).

## 4.2 Simulation in different environments



**Figure 4.10:** The simulation results of running on the flat ground,  $\mu = 0.8$  ((a) and (c)) and on the slippery ground with the low-friction,  $\mu = 0.08$  ((b) and (d)).

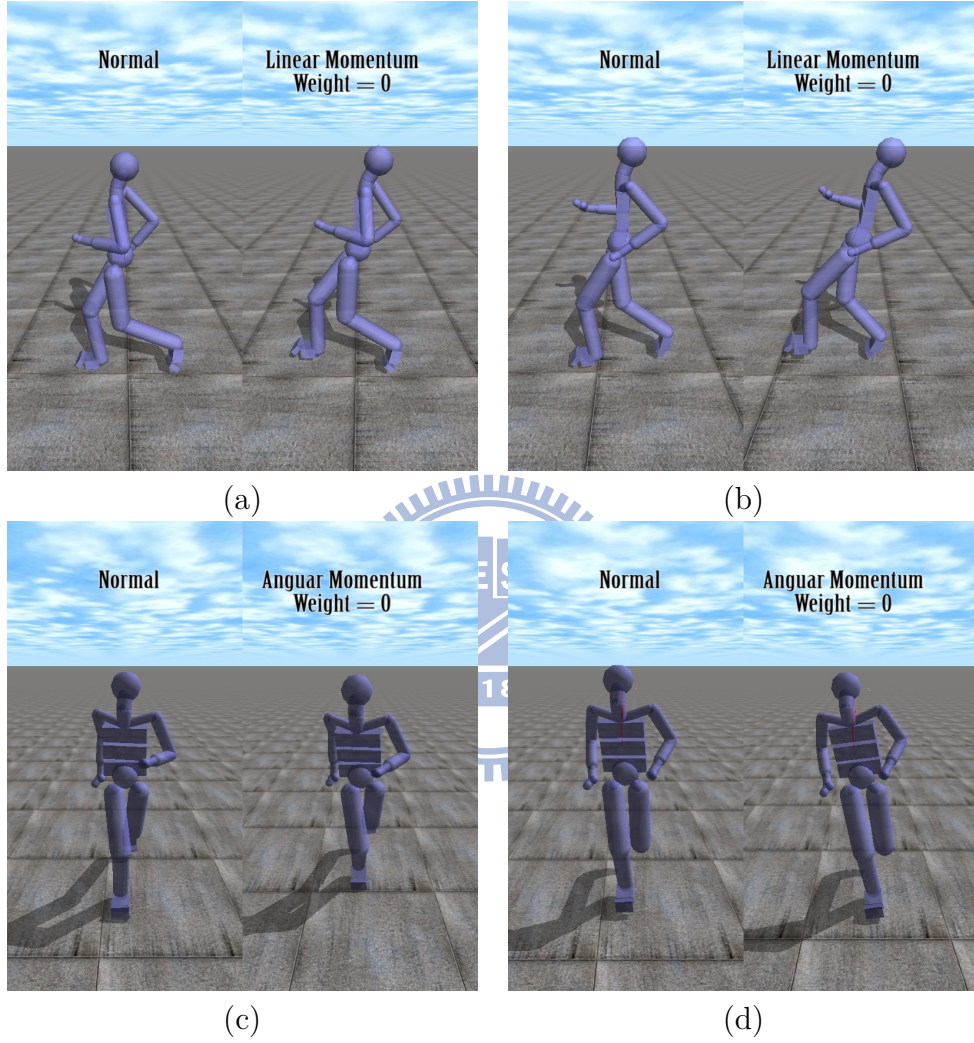
### 4.3 Effects with the objective weights and the PD gains

There are some parameters we can change, including the objective weights for optimization and the PD gains described in each objective term. We demonstrate the effects without the objective weight of the linear and angular momentum separately. As we use the linear momentum term to control the CoM, the simulated character could not move forward and produce unnatural motions by setting  $\beta_l = 0$ . Figure 4.11 (a) and (b) show the effects without linear momentum. While simulating for locomotion, this result is similar to section 4.1, it is difficult to reproduce the reference motion since the character can not move the CoM for tracking motions. In addition, this would loss the ability for maintaining balance because it is more difficult to revise the location of the CoM within the supporting region while the CoM is unexpectedly outside the supporting region. By setting  $\beta_h = 0$ , it is similar to those methods which simply control the CoM for balance; however, the character losses the ability to react against sudden rotation caused by external forces. Figure 4.11 (c) and (d) show the effects without angular momentum term, the character without the angular momentum term is more difficult to maintain the upper body upright.

In order to produce physically plausible motions, we demonstrate the effects with the PD gains for the desired accelerations of tracking (described in Equation 3.26). Since too low proportional gain may not work for tracking a motion, we focus on the effects with the derivative gain which results in different reactions in presence of external forces. By regulating the proportional gain, which is set  $k_t = 15$  in our experience, Figure 4.3 shows the effects with the derivative gains for the desired accelerations of tracking. By observing the right arm of the simulated character in the figures, lower derivative gain ( $d_t = 2$ , see Figure 4.3 (a)) is more likely to be influenced by external forces. Lower derivative may reproduce more physically plausible motions in presence of the external forces; however, this could loss the ability to react against external forces. According to Abe et al. [AdSP07], they propose  $d_t = 2\sqrt{k_t}$  for producing physically plausible motions. Thus, we use the PD gains ( $k_t = 15$ ,  $d_t = 8$ ) for the desired accelerations of tracking to make sure that the simulated character can respond to external perturbations with physically plausible motions (see Figure 4.3 (c)).

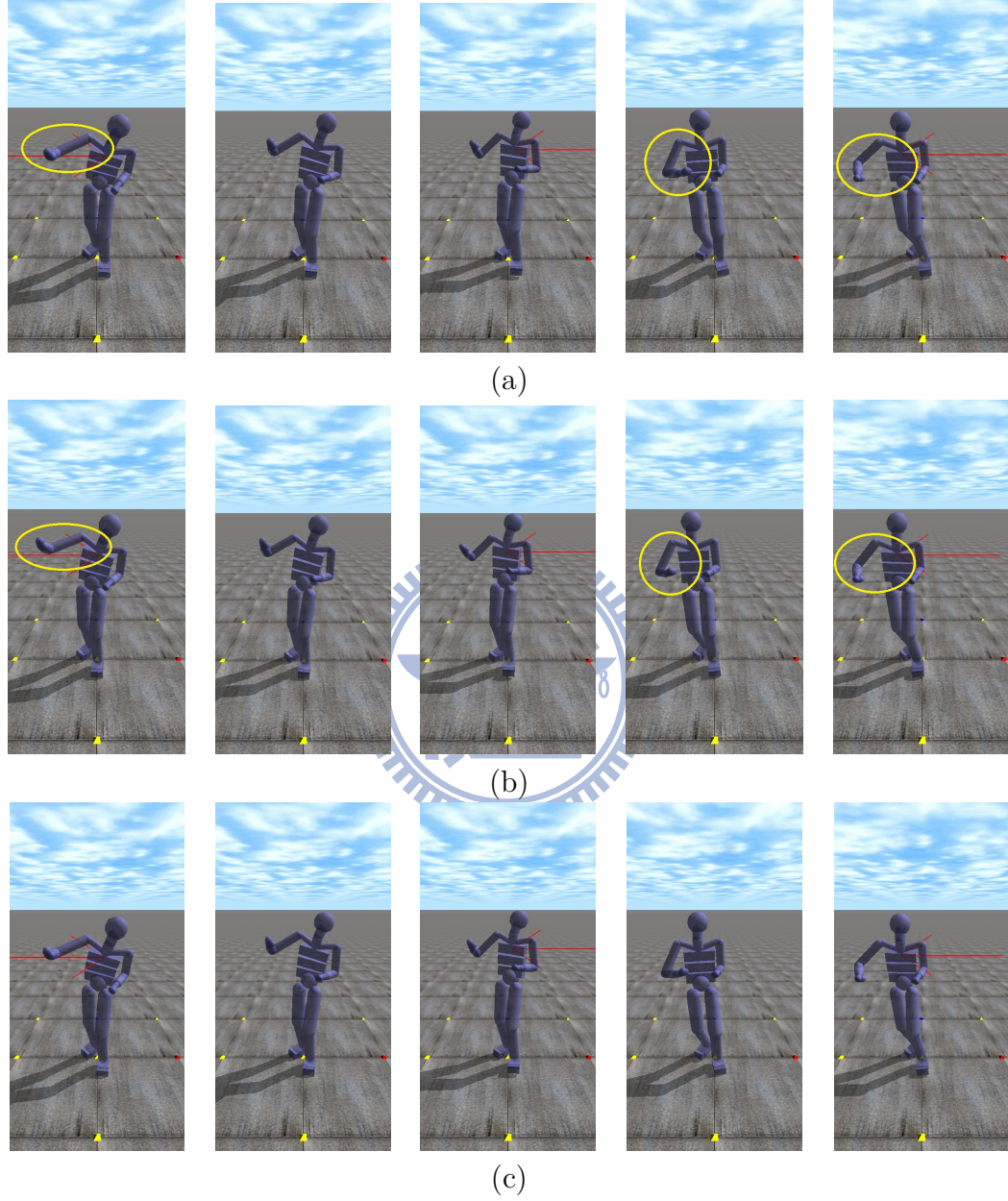


### 4.3 Effects with the objective weights and the PD gains



**Figure 4.11:** (a) and (b) show that the character loses the ability for maintaining balance and produces unnatural motions for locomotion without linear momentum term. Without angular momentum term, (c) and (d) show the character losses the ability to react against sudden rotation caused by external forces and the controller is more difficult to maintain the upper body upright.

### 4.3 Effects with the objective weights and the PD gains

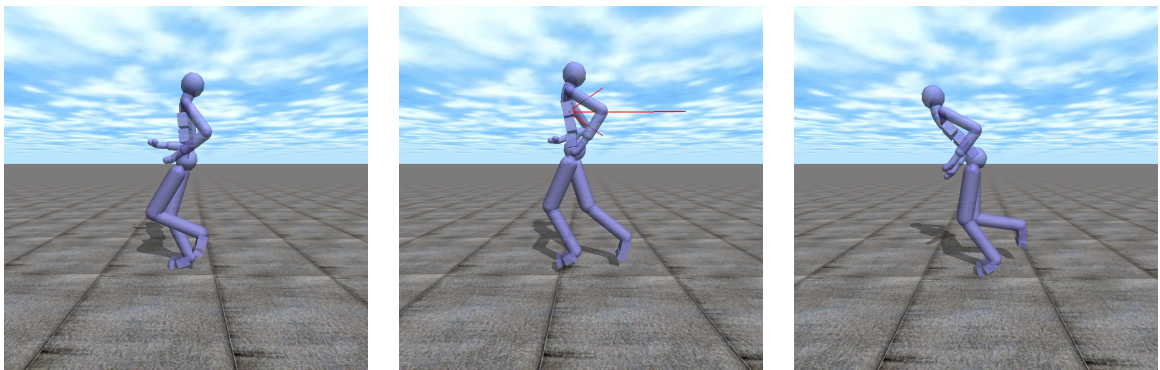


**Figure 4.12:** Comparing the results between (a) the PD gains ( $k_t = 15$ ,  $d_t = 2$ ) and (b) the PD gains ( $k_t = 15$ ,  $d_t = 30$ ) for the desired accelerations of tracking, lower derivative gain (a) is more likely to be influenced by external forces. Our experimental result uses the PD gains ( $k_t = 15$ ,  $d_t = 8$ ) for the desired accelerations of tracking (c).

## 4.4 Discussion and limitation

We demonstrate the effectiveness of our method by simulations of character balance, tracking motions and responding to external forces. Our results demonstrate that our approach can maintain a character's balance for standing, walking and running in different environments. Even when the inclination and friction of the flat ground varies, the simulated character can still track the reference motion and maintain balance under external perturbations. If the simulated character is pushed by forces in the air, it will correct imbalance after its supporting feet contacts with the ground (see Figure 4.13). In our implementation, we set the same parameters for different motions; however, the simulated character can generate different reactions by changing the parameters. In computer animation, it is a issue to find better parameters for simulating physically plausible motions.

There are some limitations in this thesis. First, our method could not achieve jumping motions since we can not control the ground reaction forces to keep the character in the air for a long time. Second, the simulated character probably kicks the slope plane or takes a false step during locomotion because the reference trajectories of the supporting feet are not adjusted. Last, because we do not have the technique of inverse dynamics, which transforms the optimal accelerations into joint torques for simulation, it probably loses some realism for physically plausible motions by using optimized joint velocities.



**Figure 4.13:** If the simulated character is pushed by forces in the air, it will correct imbalance after its supporting feet contacts with the ground.

## 5 Conclusion and Future Work

In order to maintain balance, we use a momentum-based approach to control the CoM and the CoP simultaneously. Based on the ground reaction forces, our method can dynamically adjust the reference trajectory of the CoM. Using the analytic relationship between the CoM and the CoP described by Winter et al. [WPF<sup>+</sup>96], we determine the location of the reference CoP according to the adjusted reference CoM. Comparing to the method described by Macchietto et al. [MZS09], we ameliorate the momentum-based controller for better motion tracking, and our method further achieve locomotion simulation. Unlike most previous balance controllers which need the non-linear optimization [AdSP07, dSAP08a, DSAP08b], we can simplify our multi-objective optimization problem into a system of linear equation to solve the optimal joint velocities quickly and efficiently. Without any precomputation techniques, our multi-objective optimization problem can be solved at each time step and our simulation can be executed in real-time. We show the results that our simulated character can remain balance and correct for imbalance while tracking a reference key-frame or motion capture data in presence of disturbances. Our control scheme is capable of motion simulation in varying environments, including the ground plane, the slope plane and the slippery ground.

In the future work, we would like to adjust the reference trajectories of the swing feet for locomotion to take the correct step by considering the information of the supporting plane. Besides, we can also incorporate motion planning into our approach to achieve user-specified tasks.

# Bibliography

- [AdSP07] Y. Abe, M. da Silva, and J. Popović. Multiobjective control with frictional contacts. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 249–258, 2007.
- [ALP06] Y. Abe, C.K. Liu, and Z. Popovic. Momentum-based parameterization of dynamic character motion. volume 68, pages 194–211, 2006.
- [CBvdP09] S. Coros, P. Beaudoin, and M. van de Panne. Robust task-based control policies for physics-based characters. In *ACM Transactions on Graphics (TOG)*, volume 28, page 170, 2009.
- [dLMH10] M. de Lasa, I. Mordatch, and A. Hertzmann. Feature-based locomotion controllers. In *ACM SIGGRAPH 2010 papers*, pages 1–10, 2010.
- [dSAP08a] M. da Silva, Y. Abe, and J. Popović. Interactive simulation of stylized human locomotion. In *ACM SIGGRAPH 2008 papers*, pages 1–10, 2008.
- [DSAP08b] M. Da Silva, Y. Abe, and J. Popović. Simulation of human motion data using short-horizon model-predictive control. In *Computer Graphics Forum*, volume 27, pages 371–380, 2008.
- [GK04] A. Goswami and V. Kallem. Rate of change of angular momentum and balance maintenance of biped robots. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, pages 3785–3790, 2004.
- [JYL09] S. Jain, Y. Ye, and C.K. Liu. Optimization-based interactive motion synthesis. volume 28, pages 1–12, 2009.
- [KKI02] S. Kudoh, T. Komura, and K. Ikeuchi. The dynamic postural adjustment with the quadratic programming method. In *Intelligent Robots and Systems*,



2002. *IEEE/RSJ International Conference on*, volume 3, pages 2563–2568, 2002.
- [KKI06] S. Kudoh, T. Komura, and K. Ikeuchi. Stepping motion for a human-like character to maintain balance against large perturbations. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2661–2666, 2006.
- [KKK<sup>+</sup>03] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Resolved momentum control: Humanoid motion planning based on the linear and angular momentum. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1644–1650, 2003.
- [LKL10] Y. Lee, S. Kim, and J. Lee. Data-driven biped control. volume 29, page 129, 2010.
- [LP02] C.K. Liu and Z. Popović. Synthesis of complex dynamic character motion from simple animations. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 408–416, 2002.
- [MZS09] A. Macchietto, V. Zordan, and C.R. Shelton. Momentum control for balance. volume 28, page 80, 2009.
- [PCC<sup>+</sup>09] Z. Pan, X. Cheng, W. Chen, G. Liu, and B. Tang. Real time falling animation with active and protective responses. volume 25, pages 487–497, 2009.
- [PHH04] M. Popovic, A. Hofmann, and H. Herr. Angular momentum regulation during human walking: biomechanics and control. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 3, pages 2405–2411, 2004.
- [PP97] Y.C. Pai and J. Patton. Center of mass velocity-position predictions for balance control. volume 30, pages 347–354, 1997.
- [SCCH09] T. Shiratori, B. Coley, R. Cham, and J.K. Hodgins. Simulating balance recovery responses to trips based on biomechanical principles. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 37–46, 2009.

- [SYLH10] K.W. Sok, K. Yamane, J. Lee, and J. Hodgins. Editing dynamic human motions via momentum and force. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 11–20, 2010.
- [VB04] M. Vukobratovic and B. Borovac. Zero-moment point-thirty five years of its life. volume 1, pages 157–173, 2004.
- [WPF<sup>+</sup>96] D.A. Winter, F. Prince, J.S. Frank, C. Powell, and K.F. Zabjek. Unified theory regarding a/p and m/l balance in quiet stance. volume 75, page 2334, 1996.
- [WPP<sup>+</sup>98] D.A. Winter, A.E. Patla, F. Prince, M. Ishac, and K. Gielo-Perczak. Stiffness control of balance in quiet standing. volume 80, page 1211, 1998.
- [WZ10] C.C. Wu and V. Zordan. Goal-directed stepping with momentum control. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 113–118, 2010.
- [YL10] Y. Ye and C.K. Liu. Optimal feedback control for character animation using an abstract model. volume 29, page 74, 2010.
- [YLvdP07] K.K. Yin, K. Loken, and M. van de Panne. Simbicon: simple biped locomotion control. In *ACM SIGGRAPH 2007 papers*, pages 105–es, 2007.
- [ZH02] V.B. Zordan and J.K. Hodgins. Motion capture-driven simulations that hit and react. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 89–96, 2002.