Embodiment Approaches to Humanoid Behavior

–Energy efficient walking and visuo-motor mapping–

(ヒューマノイドの行動生成への身体性によるアプローチ

–歩行エネルギーの効率化と視覚運動マッピング–)

By

Masaki Ogino

A THESIS SUBMITTED TO OSAKA UNIVERSITY

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ADAPTIVE MACHINE SYSTEMS

JANUARY 2005

Thesis Supervisor:	Minoru Asada
Thesis Committee:	Minoru Asada
	Hiroshi Ishiguro
	Koh Hosoda
	Masao Ikeda
	Kouichi Osuka

Produced in LaTeX 2ε.

Acknowledgments

I would like to thank my advisor, Prof. Minoru Asada, for providing me a chance to step in this exciting research area, robotics. I am very impressed and have respect for his serious attitude in encouraging quite a novel field in robotics, and his serious efforts in conveying the interesting and exciting accomplishments and futures in robotics to the public.

I am also grateful to my another advisor, Prof. Koh Hosoda, for leading me to this exciting, interesting and strange research area, the ”embodiment approach” in robotics. Because this area has short history from its birth, we often have had hard discussions on fundamental methodology. Those discussions give good opportunities for training the logical explanations.

Special thanks to my students for their great efforts, Masaki Haruna, Masaaki Kikuchi, Kazuhito Tawara, Issei Tsukinoki, Jun’ichiro Ooga, Shigeo Matsuyama, Yutaka Kato, Masahiro Aono, Hideki Toichi, Sou-san, Daisuke Shibuya. I am happy to work with many unique and excellent students.

Thanks to Norbert Mayer for reading through this thesis and giving me helpful advice. Thanks to Yasutake Takahashi for supporting in technological aspects in this thesis. Some researches are impossible without his image processing programs. And congratulations on your marriage! Thanks to Takashi Minato for helping me in various aspects. I was often helped by him when I was starting to send my life in Asada Lab. Thanks to Noriaki Mitsunaga for giving various advices from his rich experiences and knowledge about technologies. Thanks to Yukie Nagai for giving me cups of tea. Although her research field is not related to mine directly, I am very impressed for her active attitude to explore unique ideas in the unexplored field, developmental cognitive robotics. Thanks to Yasunori Tada for his various implicit supports. He always takes on humble jobs in Asada Lab. that I cannot do. I hope his subtle and careful works come to fruition in the very near

future. Thanks to Takashi Takuma for drinking with me. I have really enjoyed drinking with him and this often relieves my stress. I hope he accomplishes natural walking in a real biped robot that I couldn’t do in my research.

Thanks to Shizu Okada for supporting my life in Asada Lab.. She had been always annoyed with and bearing the consequences of my mistakes on office procedures.

Thanks to other all members in Asada Lab. for their supporting.

I would like to thank my parents, brothers, family, teachers, and friends. Their mental and financial supports encourage me through these long student lives.

Abstract

This dissertation presents embodiment approaches for realizing humanoid behaviors. In generating humanoid behaviors without a precise model of a humanoid, it is important to adopt an appropriate initial controller so that adequate parameter sets can be explored with avoiding falling down. The phase resetting is a plausible feature that such an initial controller should have because it enables a humanoid to walk in a wide range of parameters due to its strong entrainment to a stable limit cycle. In this dissertation, with walking controllers utilizing the phase resetting, two problems involved in humanoid walking are attacked: exploring energy efficient walking and accomplishing soccer tasks with learning visuo-motor mapping. The appropriate parameters to solve each problem are designed so as to make it easier to accomplish each task. For the first problem, we bring in the parameters to the phase reset walking controller which determines ballistic motion of the swing leg. Exploring those parameters, the same energy efficiency walking with that of human is acquired in two dimensional simulation. For the second problem, we bring in the parameters which modifies the standard walking and enables a robot to walk in various directions with various step length. These parameters are related to visual information by reinforcement learning and visuo-motor mapping, and various soccer tasks can be realized in a real humanoid robots.

Contents

Abstract **iii**

1 Introduction **1**

1.1 The embodiment approach 1

1.2 Neural basis for locomotion in biology 2

1.3 Phase reset 2

1.4 Overview 3

I Energy efficient walking of humanoid **7**

2 Related works in biped walking **9**

2.1 Model Based Approaches 10

2.1.1 The ZMP based approaches 10

2.1.2 Inverted pendulum model 11

2.2 Dynamics based approach 12

2.2.1 Passive Dynamic Walking 12

2.2.2 Central Pattern Generator 14

2.2.3 Ballistic walking 15

2.3 Approaching to the energy efficient walking from ballistic walking with Learning . 19

3 Acquiring Passive Dynamic Walking Based on Ballistic Walking **21**

3.1 Layered controller for ballistic walking 21

3.2 Simulation results 24

3.2.1 Result on a biped without knees . . . 25
3.2.2 Results on a biped with knees . . . 27
3.3 Discussion and future works . . . 28

4 Learning Energy Efficient Walking with Ballistic Walking 29
4.1 Ballistic walking with state machine . . . 29
4.2 Energy minimization with a learning module . . . 33
4.3 Comparing with human data . . . 36
4.4 Discussion . . . 38

II Visuo-motor mapping in humanoid walking 41

5 Related works in visuo-motor mapping 43
5.1 The model-based approach . . . 43
5.2 The non-model-based approach . . . 46
5.3 Learning visuo-motor mapping in humanoid behavior generation . . . 50

6 Reinforcement Learning of Humanoid Rhythmic Walking Parameters based on Visual Information 51
6.1 A RHYTHMIC WALKING CONTROLLER . . . 52
6.1.1 A biped robot model . . . 52
6.1.2 A rhythmic walking controller based on CPG principle . . . 52
6.2 REINFORCEMENT LEARNING WITH RHYTHMIC WALKING PARAMETERS 55
6.2.1 The principle of reinforcement learning . . . 55
6.2.2 Construction of action space based on rhythmic parameters . . . 57
6.2.3 Reinforcement learning with visual information . . . 57
6.3 EXPERIMENTS . . . 60
6.3.1 A robot platform and environment set-up . . . 60
6.3.2 Experimental results . . . 62
6.4 DISCUSSION . . . 63
6.5 CONCLUSION . . . 64

7 Visuo-Motor Learning for Behavior Generation of Humanoids 65

7.1 Task, Robot, and Environment 66

7.1.1 Robot platforms 66

7.1.2 Visuo-motor learning 66

7.1.3 Task and Assumptions 67

7.2 Module Learning Based on Optic Flow Information 68

7.2.1 Ball Approaching 68

7.2.2 Ball Kicking to the Opponent 72

7.2.3 Ball Trapping 75

7.3 Integration of the Modules for Face-to-face Pass 76

7.4 Discussion 77

7.5 Conclusions 78

References 81

A Planning the reference trajectory around the pitch axis 91

Chapter 1

Introduction

Human infants are able to learn to walk within about 2 years. At the beginning of walking, infants often fall down easilly. However, within two months, they walk smoothly. At the same time, infants learn not only how to walk, but also how to control the direction where they like to go. In this process, human infants acquire two principle skills. One is to make their walking more skillful. The other is to adjust their walking so that they can walk to the intended position that can be monitored through the visual information. The following sections show an approach in which a humanoid robot achieves walking skills.

1.1 The embodiment approach

Although many methods for generating humanoid motions have been proposed, they are mainly categorized into two groups. One is the model based approach, and the other is the model-free approach. In the former, a designer precisely constructs a physical model of the target system and builds a controller based on the precise model. In the latter, it is more important to make use of the intrinsic dynamics of a robot or to associate the sensor information with motions, instead of using the controller based on a physical model. For that reason, this approach has been attracting many researchers and it is called ”the embodiment approach” [1], or ”dynamics based approach” [2, 3]. However, there are few studies concerning about whole body movements of a humanoid. We suppose there are two difficulties in generating humanoid behaviors that a wheel type robot does not have. The first one is the large number of degrees of freedom (DoF). This causes the

explosion of combinations of motions that should be examined. The other is non-linear motion and intrinsic instability of a body balance. This makes it difficult to search without falling down in exploration space.

1.2 Neural basis for locomotion in biology

Here, let us see how these problems are solved in human beings. It is also thought to be a mystery how humans solve the problem of the number of DoF (the curse of dimensions). It is found that general movements observed in early infants are fundamental and germ for the following motion developments [4]. Human infants show the walking-like movements even at the very early developing stage of two days after their birth. And it is supposed that the special neural circuits, called central pattern generator (CPG) [5], takes an important and principal role in that movement [6].

From these observations, one assumption can be drawn. That is, in human development, the neural basis is predetermined for locomotion, and the parameters in the innate neural basis are learned adaptively through the developmental process. This may help to avoid exploring the useless parameter sets for a higher learning layer. In other words, CPG can be regarded as an interface between the lower neural circuits for motion primitives and the higher learning nervous systems (cerebellum or cerebrum).

Apart from the biological system, as an engineering solution for realizing bipedal walking by learning, it is plausible to prepare the appropriate interface of walking parameters for higher learning module so as to avoid exploring in the extravagant parameter space. Thus, it is important how to construct the walking controller as the interface for learning.

1.3 Phase reset

In biomechanics, some CPG models have been proposed to explain how humans walk. They use the non-linear equations to model the neural activities. Thus, they demonstrate that all the dynamics of a robot, a controller, and the environment are drawn into a stable limit cycle in computer simulations. However, from a view point of a humanoid controller, non-linear oscillator has some

inappropriate aspects with respecto to the design of the controller because the resulting behaviors of non-linear equations when the value of the parameters were changed are hard to predict.

To design a humanoid behavior, instead of directly using the CPG model, it is plausible to adopt a controller with fewer parameters, of which the physical meaning are tractable. Moreover, if we can extract the properties from the CPG model that stabilizes a biped walking, it becomes possible to design a controller with tractable parameters.

The basic idea in this thesis is to design a simple controller for stable bipedal walking that has appropriate parameters for the higher learning module. This walking controller should have the following properties;

- Even though the number of parameters are limited, it must at least realize the intended motions.
- The walking is sufficiently stable during the exploration of the parameter values.

In this study, we adopt the phase reset algorithm as the control basis for realizing stable bipedal walking. Briefly, the controller changes its walking mode from a swing phase to a support phase and vice versa when the swing leg contacts with the ground. The role of the phase reset to stabilize the walking was studied in detail by Yamasaki et al. [7]. The most attractive features of the phase reset are to enable a robot to walk in a broad area of the parameter space, to make it possible to explore the desired parameter and to design various modified motions (Fig. 1.1). The first feature is used for exploring the energy efficient walking, and the second is used for making the various walking motions that are related to visual information.

1.4 Overview

This thesis deals with two problems on the generation of humanoid behaviors using embodiment approaches. The first problem concerns the realization of energy efficient walking. The second concerns the visuo-motor mapping. For both problems, we adopt walking controllers that stabilize a biped walking based on the phase resetting. The appropriate parameters to solve each problem are designed so as to make it easier to accomplish the corresponding task. For the first problem, we bring in the parameters to the phase reset walking controller which determines ”ballistic” motion

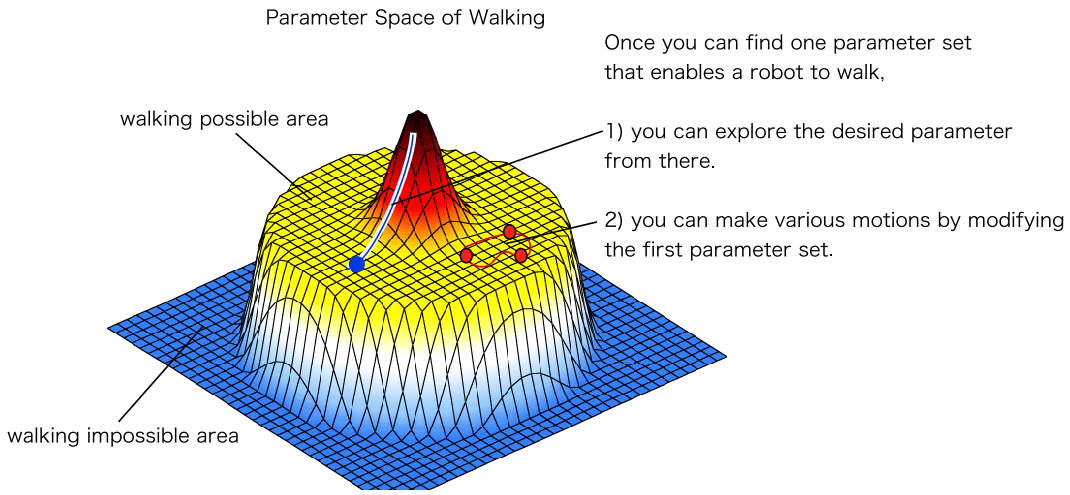


FIGURE 1.1: The advantage of using phase reset controller

of the swing leg. For the second one, we bring in the parameters which modifies the standard walking and enables a robot to walk in various directions with various step length.

Part I presents the controller of the energy efficient walking based on ballistic walking.

Chapter 2 introduces the background of biped walking.

Chapter 3 presents the proposed controller of a simple biped model consisting of two links without a torso, and shows that passive dynamic walking (PDW) can be accomplished without falling down. Then, the biped model with knees but without a torso is examined

Chapter 4 demonstrates that the walking that has the same efficiency as human walking can be achieved in the more realistic human model in two dimensional simulation.

Part II deals with the problem of the generation of humanoid behaviors utilizing the sensori-motor mapping between visual information and motor commands.

Chapter 5 introduces the existing studies of sensori-motor mapping.

Chapter 6 presents the application of reinforcement learning to the humanoid walking by finding the correlation between rhythmic walking parameters and visual information in order to approach to a ball for kicking.

Chapter 7 proposes a controller in which a robot learns sensori-motor maps in each motion module as the forward and inverse relationships between optic flows in the robot's view and motion parameters to realize a face-to-face pass between two humanoids.

Part I

Energy efficient walking of humanoid

Chapter 2

Related works in biped walking

Recently many robotics research groups have developed humanoids in academic institutes [8–10], and in commercial companies [11–13]. However, the walking realized in these studies looks awkward and seems ”unnatural” compared with human walking. What causes this difference? It is essential to consider the scientific definition of ”natural look” of walking. The movements of living creatures are optimized energetically and make best use of gravitation and inertia. We are so accustomed to seeing such a movement that our brain has a model of locomotion and feels that typical robot-like movements are strange.

The purpose of *P*art I is to propose controllers that realize energy efficient walking with making best of natural dynamics as it is in human walking. For that purpose, in this chapter, an overview of the methods of biped walking in robotics and a human walking model are presented, to clear what factors in robotics cause the unnatural biped walking and how those problems can be solved. The algorithms for biped walking that so far have been proposed can be categorized into two streams as explained in the previous chapter (Fig. 2.1). One is the *model based approach* in which the model of a robot and the environment are used explicitly during control. The other is the *dynamic based approach* in which the model is not used during control although it can be used to verify the stability of the movement.

In the following, I present the researches done so far in this field.

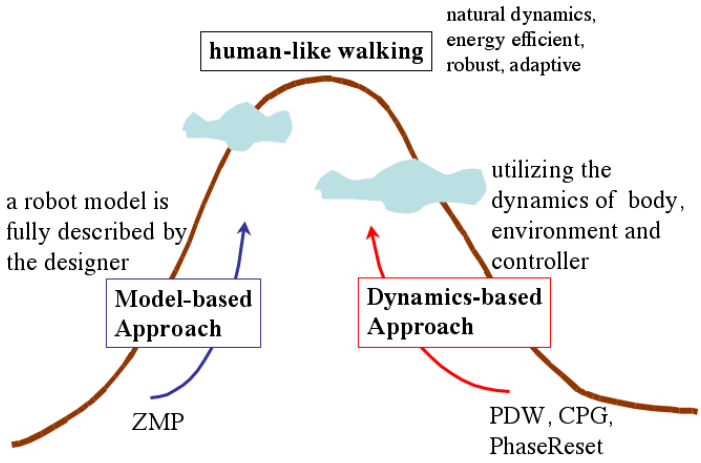


FIGURE 2.1: Two approaches in biped walking algorithms

2.1 Model Based Approaches

One of the common features in this approach is to set the reference trajectories for each joint angles. The basic ideas for calculating the trajectories differs from methods to methods. But broadly categorized, they are classified into three basic concepts based on the core indices, ZMP and inverted pendulum mode. In the followings, the key idea of each method is introduced.

2.1.1 The ZMP based approaches

In bipedal walking, the state such that the foot of the support leg contacts with the ground at one point should be thought to be avoided, because the system becomes uncontrollable. In static walking, in which the walking speed is slow, if the projected point of center of mass (CoM) of a robot exists within the area of the foot of support leg, the foot does not get into such an unstable state. Zero moment point (ZMP) was proposed so that the same idea for the stability as in static walking can be extended to the dynamic walking. ZMP can be defined as the point on the ground where the total moment generated due to gravit yand inertia equals to zero [14–16]. Actually, ZMP can be proved to be the same as the point of center of pressure (CoP) [17].

In the approaches based on ZMP index, the reference trajectories to be followed by high gain

PD controllers are designed so that ZMP always exists within the foot of the support leg or the polygon consisting of the support legs. For example, Takanishi group in Waseda University [9, 14, 18–20], the group that realized the biped walking in real robots based on ZMP since the very early stage of humanoid robot researches, describes the trajectories of the arms, legs and ZMP by the Fourier series and determines its coefficients to ensure the ZMP conditions in simulation. The preset walking pattern is then played back using joint position control during walking with compensated by the trunk. They tried to make use gravitational and inertial force with varying the compliance of the legs, but the realized walking was not so natural as human walking.

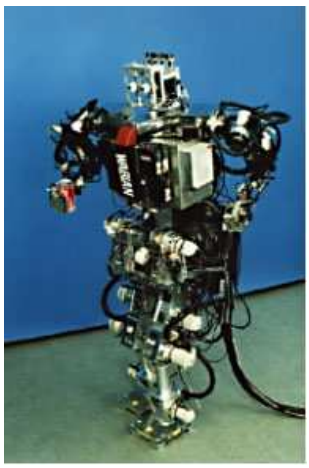


FIGURE 2.2: The humanoid robot in Waseda University: WABIAN

In this approach, once you determine the ZMP trajectory, you can calculate the joint trajectories to realize the ZMP trajectory based on the robot and the environment models. However, it is difficult to determine the trajectories that fit to the natural dynamics as well as that fit to the ZMP conditions. Moreover, you must select the appropriate ZMP trajectory that fits to the natural dynamics among numerous possible walking trajectories.

2.1.2 Inverted pendulum model

In order to determine the reference trajectories that meet with natural dynamics, it is convenient to adopt the simplified model so as to understand the intrinsic dynamics in the system. Inverted pendulum model has been often used for bipedal control because, in the single supported phase,

human walking can be modeled as the inverted pendulum model [21–26].

For example, Kajita et al. [24] modeled a robot with simplified model in which the legs were massless, the upper body was a rigid inertial element and the robot walked with the center of mass at a constant height. These assumptions enables to linearize equations of motion and makes it easy to determine the reference trajectory that fit to the natural dynamics. Recently, they extended their methods from the simplified two dimensional model to the three dimensional model with the inverted pendulum in three dimensions.

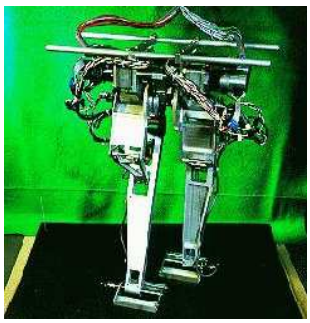


FIGURE 2.3: Kajita's robot: Meltran II

The model of this approach is to relate the simplified low-dimensional model to the real complex high-dimensional robot model. Simplification often needs peculiar assumptions such as short lengths of steps or linear movement of the waist.

2.2 Dynamics based approach

This section presents the studies in the dynamics based approach. They can be categorized into three groups from the view point of the originating idea; PDW, CPG and ballistic approaches. However, the resulting control are quite similar in spite of the difference of their algorithms.

2.2.1 Passive Dynamic Walking

The robot that walks down on a shallow slope without any actuator has been popular since early times. The oldest record I know is the patent about the shape of the feet of the toy [27] (Fig. 2.4).

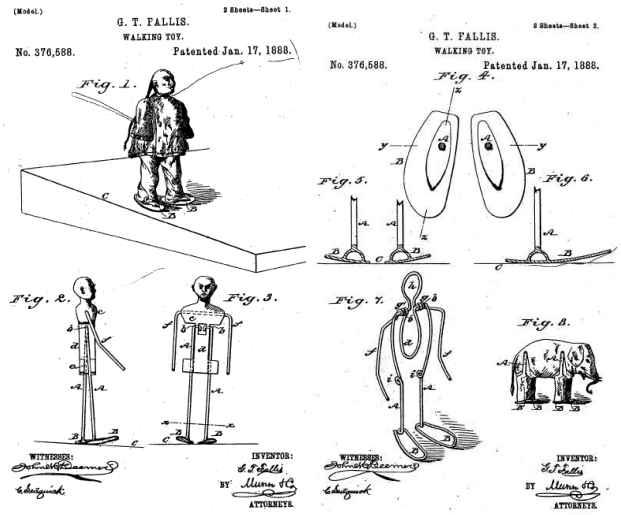


FIGURE 2.4: The oldest record of PDW: [27]

It is McGeer's work [28, 29] that brought the toy to the stage of robotic research field as introduced in the previous chapter. His work encourage many researchers to realize bipedal walking as a mean to make best use of the intrinsic dynamics of a structure of humanoids.

In the early stage of this field, the properties of the dynamics that the simple PDW model has were investigated. Coleman et al. [30] shows that the dynamics of PDW can be thought as the extension of the rimless wheel. It is shown that the chaotic features appears in the boundary area between the stable and unstable walking in simulation [31–34] and in a real robot [35].

The attempts to enhance a PDW robot with small actuation and with less control so that it can walk in a level floor have begun around 2000. Sugimoto and Osuka [37] developed a real robot which has 4 PDW (total 8 legs) and proposed a controller derived from a chaotic control (Fig. 2.6). Asano et al. [38] proposed a controller that applies torque to the joints during walking on the level floor so that the torque are the same as that given by the gravity in PDW. However, their methods might not be categorized in "dynamic based approach" because they use the model of a robot to derive the necessary torque.

FIGURE 2.5: A robot which can realize three dimensional PDW :[36]

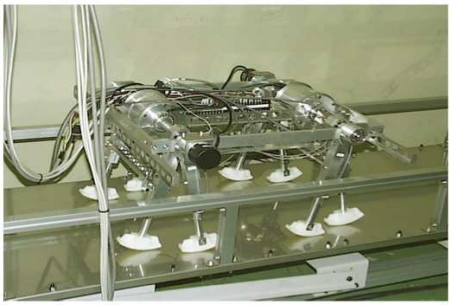


FIGURE 2.6: PDW robot of Osuka and Sugimoto: Quartet III

2.2.2 Central Pattern Generator

Vertebrate animals are thought to have neural basis for locomotion in spinal chord, named central pattern generator (CPG) [5, 39]. Matsuoka proposed a mathematical model of CPG and demonstrated that the combinations of simple neural models can generate the neural activities for locomotion [40]. Taga applied the CPG model proposed by Matsuoka to a human model and realized stable biped walking in simulation [41]. He also showed his CPG model realizes very stable walking on a shallow slope and against sudden external forces. Taga suggests that this robust limit cycle is the result of entrainment of the dynamics among the mechanical properties of a human model, the CPG controller and the environment [42, 43].

Although various CPG models have been proposed and studied (for example, for locomotion of

insects [44], for swimming and locomotion of lamprey [45]) or salamander [46], here I'll introduce the simple Matsuoka's model as typical CPG model.

Kimura et al. [47–50] realized 4-legged locomotion with Matsuoka type of CPG model in real robots. They used the outputs of the CPG neurons as the desired value of the joint angles instead of torque value. For biped walking, there have been few applications yet. The one example is Miyakoshi's study on stepping motion of humanoid model in three dimensional simulation [51].

One of the difficulties in application of CPG model to real robots is to determine the coefficients of neural connections. As explained in Matsuoka's model, CPG models includes many neurons and they are connected mutually. This is the main reason why genetic algorithm have been often used to solve this problem in general [52–57]. Another difficulty is that almost all robots employs the proportional and derivative control for motor control in which the desired angle and angle velocity should be given, whereas, in human model, the outputs of the CPG neurons are used as torque.

Recently, CPG models have begun to be used as the rhythmic oscillator which can entrain with the external inputs, and applied to various movements other than walking [58–60].

2.2.3 Ballistic walking

Ballistic walking was originated from the simple human walking model proposed by Mochon and McMahon [61–63]. They got the idea from the observation of human walking in which the muscles of the swing leg are activated only at the beginning and the end of the swing phase. Although their model is simple which starts with certain initial value at the beginning of the swing phase and afterwards moves only under the gravity (This is why this model is named "ballistic".), it can elucidate the observed data of human walking period.

Here, "ballistic walking family" is broadly defined as the walking controllers in which the movement of the swing leg is governed by the gravitational and the inertial force in the middle of the swing phase. Because ballistic walking originated as a human walking model, it is difficult to distinguish clearly ballistic walking controllers from CPG controllers. However, it can be said that CPG controllers use the intrinsic features of non-linear equations, whereas ballistic controllers only mimic the apparent properties of human walking.

In the following, we will introduce three main approaches of ballistic walking.

Delft Biped Laboratory of Delft University in Holland has developed real robots with pneumatic actuators. Pneumatic actuators are intrinsically difficult to use for positional control, but they can be used passive factors when the internal pressure is very low. Their group utilizes this characteristics of pneumatic actuators very well. Their controller uses the actuators only for moving the swing leg forward, instead of using for positional control [64–67]. There are two points for keeping stability of walking in their control. The first point is to build a robot body so that it can realize passive dynamic walking. The second is to determine the timing of turning on the pneumatic actuators depending on the contact of the swing leg with the floor. In spite of this simple controller, they have realized the two dimensional and three dimensional waking without torso, and three dimensional walking with torso [68, 69].

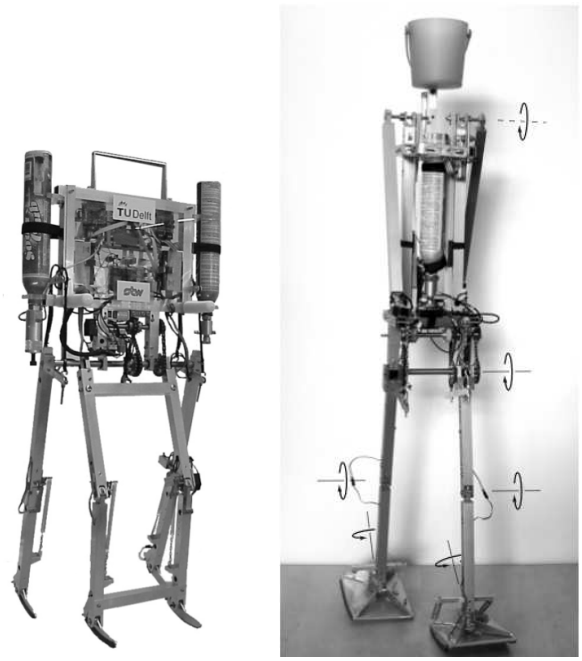


FIGURE 2.7: Max (left) and Denise (right) in Delft University

Pratt et al. [70, 71] of Leg laboratory in Massachusetts Institute of Technology realized robust walking in a two dimensional real robot with the controller in which no control is applied during the middle of the swing phase. Even though they use electric motors, they achieve the passive phase with the series elastic actuators that enable a robot to vary the compliance of the joints. The

controller they adopt is a state machine algorithm which transits its states depending on the sensor information at the contact of the swing leg with the floor and time. However, it does not change the trajectories of joint angles, but change only the timing of the transition of control state. With this simple controller, they realize the stable walking with 1.25 [m/sec].

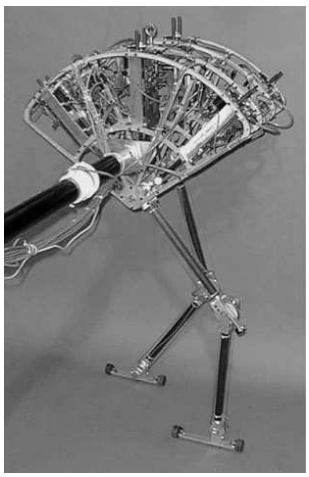


FIGURE 2.8: Spring Flamingo in MIT

Ono et al. in Tokyo Institute of Technology propose the controller that applies torque to the hip joint proportional to the bending angle of the knee joint of the swing leg. They realize a stable walking in simulation [72, 73] and in real robots [74] and insist that with this controller the whole system makes a stable self-excited limit cycle. Their robots have actuators only in the hip joint. The knees are fully passive during the swing phase and fixed by solenoids or magnetic brakes during the support phase. In their controller, only the knee joint angle is used for sensor information, with which it can detect the contact with the floor and changes the control mode from swing phase to support one and vise verse. With this minimal configurations, they realize a stable planar walking.

It is important and interesting to think about what properties ballistic controllers have in common. The common features in the controllers mentioned above are stated as follows;

- In the beginning of the swing phase, the torque to make a swing leg move forward is applied.
- In the support phase, the torque to avoid bending of the knee joint of the support leg is applied.

FIGURE 2.9: biped Robot 3 of Ono et al. [74]

- The ground contact of the swing leg turns the control mode from the support one to the swing one and vise verse.

It is meaningful to note that CPG controllers have similar features as these items. In CPG models, the neurons are activated so that the torque is applied in the same way as mentioned above. Furthermore, the ground contact information takes an important role in turning the support mode and the swing mode in a controller. It is also interesting that the newer version of Taga's CPG model to aim the more robust walking adopts "global states" as the input of the CPG neurons that helps the CPG network to work in the same way as state machines such as Pratt's model.

I believe the features mentioned above are fundamental to make a stable limit cyle for biped walking, and the controller with those features can realize the stable walking, whatever algorithms are adopted for implementing those features.

- Among these features, the third feature is regarded as the basis of the stable limit cycle. The feature is named *phase reset* and the dynamics it affects to the system are investigated in detail. Actually, Tsuchiya et al. propose a biped walking algorithm applying this feature that does not have passive phase in control, and they demonstrate the stable walking with a real small bipedal robot [75, 76].

2.3 Approaching to the energy efficient walking from ballistic walking with Learning

The purpose of my researches in chapter I is to realize energy efficient walking with exploiting natural dynamics. Needless to say, PDW is the most efficient walking. However, PDW has some difficulties for application to walking in a normal environment. The initial conditions that converge to PDW are difficult to find, and the entrainment to the stable limit cycle occurs within the limited small area. Although energy efficient walking has a limit cycle, it is hard to reach it. How does a robot reach such a stable limit cycle? And how can it be accomplish adaptively without any knowledge of a robot and the environment, a priori?

The studies that follow in the next two chapters are one of the possible solutions for that problem. As explained in the previous section, ballistic walking model provides the foundations for stable walking against any control model. This means ballistic walking controller has larger area for stable walking in control parameter space. Our aim is to go from ballistic walking to energy

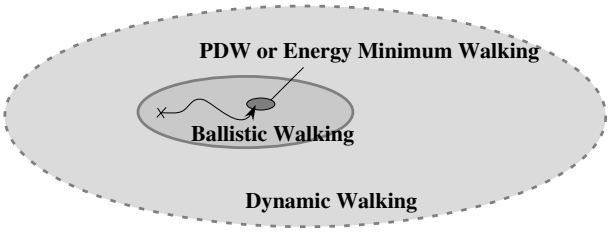


FIGURE 2.10: Approaching from ballistic walking area to the energy efficient walking

efficient walking. Our controller consists of two layeres; the lower layer stabilizes the ballistic walking, while the upper layer tries to realize the minimal energy walking. In the followings, this type of controller is applied to from a simple biped model to a more realistic human model.

Chapter 3 presents the application of the proposed controller to the simplest biped model consisting of two links, without a torso and shows that PDW can be accomplished successively. Then, the biped model with knees but without a torso is examined. Finally, it is shown that the walking that has the same efficiency as human walking can be achieved in the more realistic human model in Chapter 4.

Chapter 3

Acquiring Passive Dynamic Walking Based on Ballistic Walking

Abstract

In this chapter, we propose a layered controller which enables the biped robot to walk adaptively in minimal energy or passive dynamic walk, if possible. This controller consists of two layers: the lower layer stabilizes the walking while the upper layer realizes the minimal energy walking. The torque is applied to the robot in short time after the the free leg leaves the ground, and so the walking is ballistic. Simulation results show that the proposed controller can realize passive dynamic walking successively, starting from ballistic walking in the simple robot model.

3.1 Layered controller for ballistic walking

Taking the Poincaré section at heel contact in phase space of walking motion, the change of the walking in each step is described as the transposition of the point on the Poincaré section.

In the passive dynamic walking in which no external force is applied to the robot, the state of the (n+1)-th step depends only the state of the n-th step;

$$s_{n+1} = f(s_n) \tag{3.1}$$

Many studies on ballistic walking have derived the conditions for stable walking from the condition

that $s = f(s)$ on Poincaé section. This condition is quantified by the multi-dimensional partial derivative (or Jacobian) Δf. If all eigenvalues of Δf lie within the unit circle, walking is stabilized [65] [31].

In the ballistic walking, the condition for stable walking is analyzed in the same way. Suppose the torque, τ_w, is applied in short time after the free leg leaves the ground in the following way,

$$\tau_w = \begin{cases} A_n & (0 < t < t_0) \\ 0 & (t_0 < t < T) \end{cases} . \tag{3.2}$$

Here, t is the elapsed time after the free leg left the ground, t_0 is small time as compared with walking cycle T. A_n is constant during the n-th step.

In this case, the displacement equation on Poincaré section becomes

$$s_{n+1} = f(s_n, A). \tag{3.3}$$

In this ballistic walking in which torque is applied during short time, contrary to passive dynamic walking, the walking stability is controlled to some extent because the next state, s_{n+1} can be changed by A_n. Formal'sky derived the stable walking condition by formulating the equation (3.3) as a boundary-value problem like $s = f(s, A)$ when t_0 is infinitesimal (the torque A is applied like δ-functions.) [77, 78]. Linde has found the condition for stable walking using the Jacobian of f [66].

The method mentioned above is effective when all parameters of the robot model are known a priori. But if the parameters are unknown, how robot can attain the minimal energy gait adaptively using its own sensor inputs?

To assure the walking stability, it is necessary to give the appropriate torque to the robot according to the current state s_n, so that s_n is a cyclic solution. For that, the inverse relation of the equation (3.3) is useful:

$$A_n = g(s_n, s_{n+1}). \tag{3.4}$$

If this function is available, the controller to realize the desired state can be built as follows:

$$A_n = g(s_n, s_d). \tag{3.5}$$

Under this control, the desired state to realize the gait with minimum energy can be searched safely

by the following equation.

$$A_{min} = \min_{s_d} g(s_d, s_d) \tag{3.6}$$

We propose a controller which implements the equation (3.5) as the lower layer and the equation (3.6) as the upper layer as shown in Fig 1. The lower layer of the controller regulates the torque so as to keep the intersectional point on the Poincaré plane, s_n, coincident to the desired state, s_d. The upper layer of the controller receives the information of current magnitude of the torque applied to the waist, τ_w, from the lower layer and gives the new desired state to be realized in the lower layer.

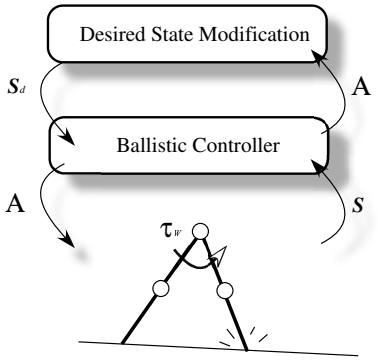


FIGURE 3.1: A proposing layered controller for biped walking

In the lower layer, the controller should implement the equation (3.5). But it is impossible to find the function f without knowing the physical parameters of the robot. If the model of the robot is simple, it is possible to prepare the simple feedback controller that works as the function f by getting the relationship of torque, A_n, and the resultant state, s_{n+1} from observing several trial walks. An example of this implementation will be shown in section 3.3. For more complicated robot models, it is difficult to obtain the relationship from observations of trial walks. Thus we introduce a neural network that maps the causality of the torque and the resultant state. This neural network implements the equation (3.4) from the several trials in the learning phase. The simulation relult of this implementation will be shown in section 3.3.

In the upper layer, to implement the equation (3.6), the desired state, s_d, that enables the robot to walk with minimum energy is searched by stochastic hill-climbing method, as follows.

$$if(|s - s_d| < \delta)$$

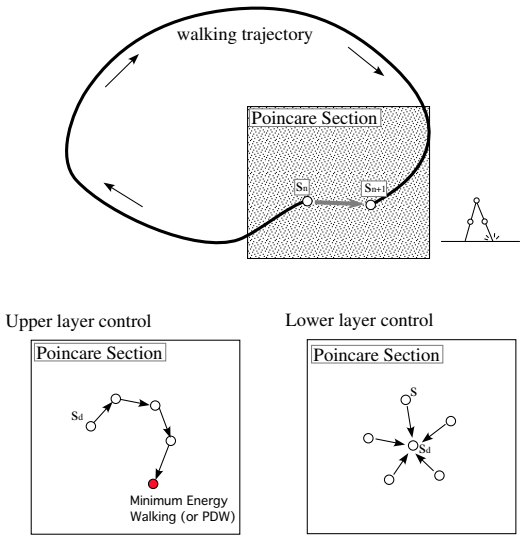


FIGURE 3.2: Poincaré section is taken at heel strike. On this plane, the lower layer of the controller tries to keep the state at heel contact in the same area as the desired one which is given by the upper layer. The upper layer of the controller tries to search the desired state which realize walking mode with minimum energy.

$$
\begin{aligned}
&if(A_{min} > A) \\
&\qquad A_{min} = A \\
&\qquad s_{d0} = s_d \\
&s_d = s_{d0} + random\ perturbation
\end{aligned}
$$

The upper layer of the controller modifies the desired state s_d. While ballistic torque produced by the controller varies from time to time, the upper layer does not change the desired state. When the torque is settled, which is observed by the error $|s - s_d| < \delta$, the layer will modify the desired value in a random manner, and find the gradient to reduce the ballistic torque. In this way, the layered controller makes the biped learn the passive dynamic walking by itself.

3.2 Simulation results

Several simulations are performed to show the effectiveness of the proposed layered controller. The first experiment shows that the proposing controller can realize passive dynamic walking stably on

a simple robot without knees. In the second experiment, it is shown that introducing a neural network in the lower layer controller can successfully suppress the limping gait of a biped robot with knees and then a gait with minimal energy can be realized.

In the following, each leg of a biped without knee is 1.5 [kg] and 0.6 [m]. For a biped model with knees, the mass of thigh and shank is 1.0 [kg] and 0.5 [kg], respectively, and the length of them is 0.3 [m]. The contact between the feet and the floor is modeled with spring-dumper model. The coefficients of spring and dumper of the floor model is 20000.0 [N/m] and 100.0 [N sec/m], respectively. The iteration time is 0.2 [msec]. The dynamics simulation is performed by articulated body method [79].

3.2.1 Result on a biped without knees

In this experiment, to stabilize the ballistic walking, constant torque is applied to the hip joint during 0.1 [sec] after the free leg leaves the ground. As the first trial to implement "ballistic controller", we adopted the simple feedback algorithm,

$$A_{n+1} = A_n - \alpha|s - s_d|, \tag{3.7}$$

and as the representational variables of walking state, we adopt the angular velocity of waist joint, $\dot{\theta}_w$, so above equation becomes

$$A_{n+1} = A_n - \alpha(\dot{\theta}_w - \dot{\theta}_{wd}), \tag{3.8}$$

where $\dot{\theta}_{wd}$, and α denote the desired value of $\dot{\theta}_w$ provided from the upper layer, and feedback gain, respectively.

The upper layer of the controller modifies the desired joint angle $\dot{\theta}_{wd}$. While ballistic torque produced by the controller varies from time to time, the upper layer does not change the desired angle. When the torque is settled, which is observed by the error $|\Delta\dot{\theta}_w| < \delta$, the layer will modify the desired value in a random manner, and find the gradient to reduce the ballistic torque. In this way, the layered controller makes the biped learn the passive dynamic walking by itself, if possible.

A simulation result with a biped without knees is shown in Fig. 3.4 . Figs. 3.4 (a) and (b) show the angular velocities of the waist joint at the impact of each step, and the torque magnitude applied to the waist joint, respectively. At the beginning, the torque is not settled, therefore the

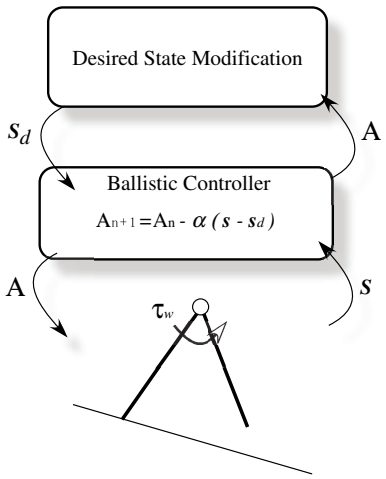
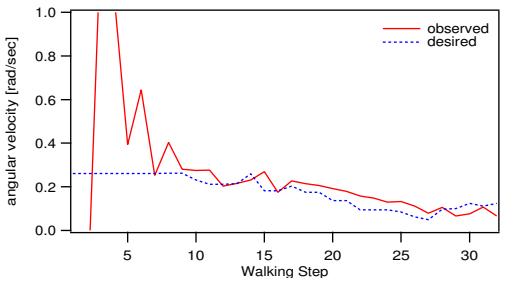
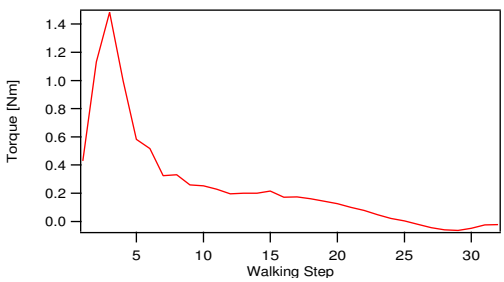


FIGURE 3.3: Layered controller with simple feedback controller

desired waist velocity is not changed. Nevertheless, the biped continued to walk thanks to the ballistic controller. After 10 steps, the torque is settled, and the upper layer begins to search for a better desired value that makes the torque smaller. Afterward, the torque is continuously decreased and passive dynamic walking is realized within 30 steps.



(a) the angular velocity of hip joint at the moment of heel strike in each step



(b) the magnitude of output torque at each step

FIGURE 3.4: Walking mode converges to PDW under the proposing controller when the slope is inclined 1 [deg]

3.2.2 Results on a biped with knees

When the feedback control proposed above is applied to a robot with knees, ballistic learner cannot make the walking mode approach to the desired one because the walking mode is easily fell into the 2-cycled mode. To let walking mode converge to 1-cycled mode, the ballistic learner is enhanced with neural network as shown in Fig. 3.5.

This neural network calculates the magnitude of the torque when the state at heel contact and the difference between the current state and the desired one are given. Because the appropriate torque value is unknown a priori, the function g of the equation (3.6) is realized by neural network. In training phase during which random torques are given to biped robot, the neural network is trained by means of backpropagation algorithm so that the A_n at the n-th step can be obtained when the state at n-th step and one at the $(n+1)$-th step are given. The algorithm of the upper layer is the same as that described in the previous section.

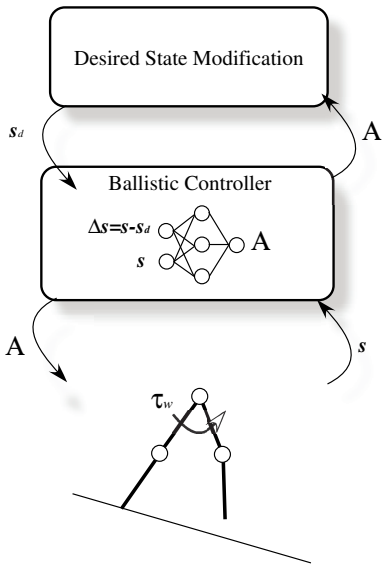
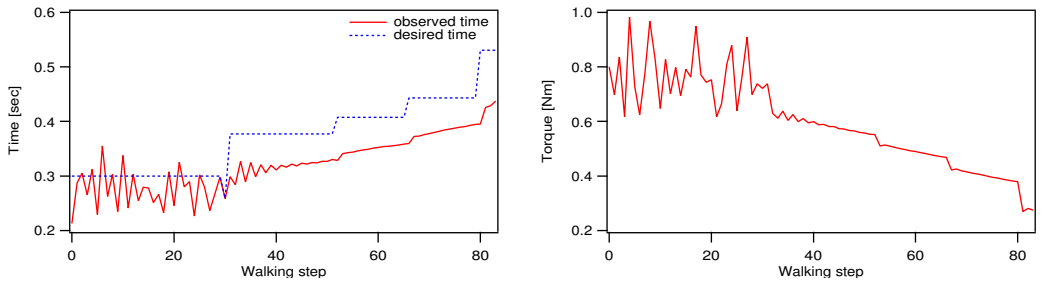


FIGURE 3.5: Layered controller with neural network

Fig. 3.6 shows simulation results. In this experiment, the time from the moment the knee of free leg is straight to the heel contact, is adopted as the representational variable of walking state. Until the 30th step, random torque is given for training the neural network of ballistic learner. When the ballistic learner works after the 30th step, the walking mode is stabilized by ballistic

learner and converges to one cycled walking within 10 steps. And after about the 50th step, the upper layer controller is activated and begins to search better desired value of the state.



(a) the time during which the knee of free leg is straight

(b) the magnitude of output torque at each step

FIGURE 3.6: Applying layered controller to robot with knees

3.3 Discussion and future works

Here, we discuss about two problems of the method presented in this chapter.

The first problem is about the representative variables of the state. In this chapter, as the representative variables of the state, we adopted the angular velocity of hip joint for the biped model without knees, and the time during which the knee of free leg is straight, for the biped model with knees. However, currently we do not know what kind of variables should be chosen. Other variables or some combinations of them might be plausible. How to choose the appropriate variables is the next challenge.

The second problem is the prediction of falling down. The upper layer tries to search the desired state which enables the robot to walk with minimum energy, without knowing the safety zone of states on Poincaré section. So, if the robot cannot realize passive dynamic walking, the robot inevitably falls down in the final stage of searhing. We are now trying to add another module which evaluates the stability of walking.

Chapter 4

Learning Energy Efficient Walking with Ballistic Walking

Abstract

This chapter presents a method for energy efficient walking of a biped robot with a layered controller. The lower layer controller has a state machine for each leg. The state machine consists of four states: First, constant torque is applied to hip and knee joints of a swing leg. Second, no torque is applied so that the swing leg can move in a ballistic manner. Third, a PD controller is used so that the certain posture can be realized at the heel contact, which enables a biped robot to walk stably. Finally, as a support leg, hip and knee joints are servoed to go back and the torque to support the upper leg is applied. With this lower layer controller, parameters that enable robot to walk as energy efficiently as human walking can be searched by the upper layer controller without paying any attention to avoid falling down.

4.1 Ballistic walking with state machine

Here, we use a robot model consisting of 7 links: a torso, two thighs, two shanks and two foots as shown in Fig. 4.1.

The state machine controller at each leg consists of four states, as shown in Fig. 4.2: the

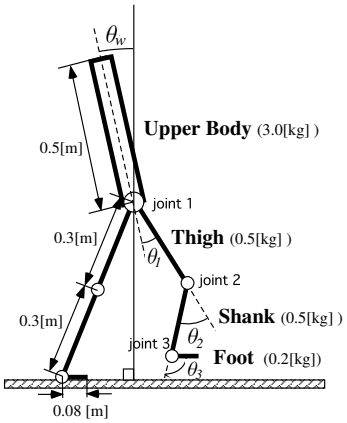


FIGURE 4.1: Robot model

beginning of the swing phase (*swing I*), the middle of the swing phase (*swing II*), the end of the swing phase (*swing III*) and the support phase (*support*).

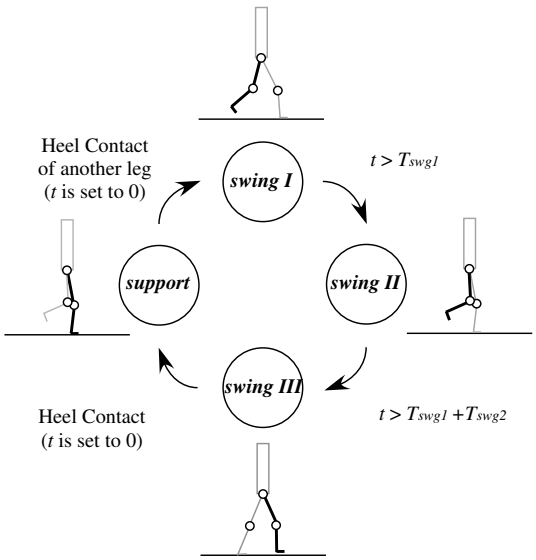


FIGURE 4.2: A state machine controller consisting of four states

In the support phase, the hip joint is servoed by a proportional derivative (PD) controller so that the torso stands up and the support leg goes back. To the knee joint, torque is applied so that the knee joint becomes straighten during the support phase. Therefore, the torque applied to the

hip and waist joints are given by the following equations;

$$\tau_1 = -K_p(\theta_1 - \theta_{1d}) - K_v(\dot{\theta}_1 - \dot{\theta}_{1d}) - K_{wp}\theta_w - K_{wv}\dot{\theta}_w, \tag{4.1}$$

$$\tau_2 = -K_p(\theta_2 - \theta_{2d}) - K_v(\dot{\theta}_2 - \dot{\theta}_{2d}). \tag{4.2}$$

The reference trajectory for the above PD controllers are described with the simple sinusoidal functions which connect the angle of the beginning of the state to the desired one to be realized at the end of the state,

$$\theta_{1d}(t) = \begin{cases} \frac{(\theta_{1e}-\theta_{1s})}{2}(1 - \cos\frac{\pi t}{T_{spt}}) + \theta_{1s} & (t < T_{spt}) \\ \theta_{1e} & (t \geq T_{spt}) \end{cases}, \tag{4.3}$$

$$\dot{\theta}_{1d}(t) = \begin{cases} \frac{\pi(\theta_{1e}-\theta_{1s})}{2T_{spt}} \sin\frac{\pi t}{T_{spt}} & (t < T_{spt}) \\ 0 & (t \geq T_{spt}) \end{cases}, \tag{4.4}$$

$$\theta_{2d}(t) = \begin{cases} \frac{(\theta_{2e}-\theta_{2s})}{2}(1 - \cos\frac{\pi t}{T_{spt}}) + \theta_{2s} & (t < T_{spt}) \\ \theta_{2e} & (t \geq T_{spt}) \end{cases} \tag{4.5}$$

and

$$\dot{\theta}_{2d}(t) = \begin{cases} \frac{\pi(\theta_{2e}-\theta_{2s})}{2T_{spt}} \sin\frac{\pi t}{T_{spt}} & (t < T_{spt}) \\ 0 & (t \geq T_{spt}) \end{cases} \tag{4.6}$$

where θ_{*s} indicates the angle at the moment when the controller enters the support phase (the moment of contact of the swing leg with the ground), and θ_{*e} indicates the desired angle that should be realized at the end of the support phase. t is the time since the controller enters to the support phase and T_{spt} is the desired time when the support phase ends. In this simulation, the control gains are set as $K_p = 300.0$ [Nm/rad], $K_v = 3.0$ [Nm sec/rad], $K_{wp} = 300.0$ [Nm/rad] and $K_{wv} = 0.3$[Nm sec/rad], and the desired angles of the end of the support phase are set as $\theta_{1e} = 20.0$ [deg] and $\theta_{2e} = 0.0$ [deg].

The swing phase is separated to three states; *swing I* (the beginning phase), *swing II* (the middle phase), and *swing III* (the end phase). In *swing I*, the controller applies constant torque to both the hip and knee joints. After the certain time passes, the control state changes to *swing II*, in which no torque is applied to the hip and knee joints. Therefore, in *swing II* the swing leg moves in a fully

passive manner. After the swing time passes T_{swg2}, the control state changes to *swing III*, in which the joints are servoed using PD controllers so that the desired posture at the end of the swing phase can be realized. By taking a certain posture at the moment of ground contact, a certain degree of walking stability can be assured. The state of the controller transits to the state *support* when the swing leg contacts with the ground. The output torque can be summarized as the following equations,

$$\tau_1 = \begin{cases} A & (t < T_{swg1}) \\ 0 & (T_{swg1} \leq t < T_{swg1} + T_{swg2}) \\ -K_p(\theta_1 - \theta_{1d}) - K_v(\dot{\theta}_1 - \dot{\theta}_{1d}) & (T_{swg1} + T_{swg2} \leq t) \end{cases} \tag{4.7}$$

and

$$\tau_2 = \begin{cases} -B & (t < T_{swg1}) \\ 0 & (T_{swg1} \leq t < T_{swg1} + T_{swg2}) \\ -K_p(\theta_2 - \theta_{2d}) - K_v(\dot{\theta}_2 - \dot{\theta}_{2d}) & (T_{swg1} + T_{swg2} \leq t) \end{cases} \tag{4.8}$$

where the reference trajectory in *swing III* is given in the same manner as the support phase, eqs. 4.3-4.5. In our study, the desired angles of the hip and knee joints at the end of the swing phase are set as $\theta_{1e} = -20$ [deg] and $\theta_{2e} = 0$ [deg], respectively. T_{swg1} and T_{swg2} are set to 0.2 [sec], and 0.05 [sec].

Throughout walking, a PD controller with the small gains ($K'_p = 3.0$ [Nm/rad] and $K'_v = 0.3$ [Nm sec/rad]) is used to the ankle joints,

$$\tau_3 = -K'_p(\theta_3 - \theta_{3d}) - K'_v(\dot{\theta}_3 - \dot{\theta}_{3d}). \tag{4.9}$$

The desired angle of the ankle joint is always fixed to 90 [deg]. Therefore, the ankle joint works as a spring is attached.

The simulation result of the controller is shown in Fig. 4.3, in which the resultant torque curves are shown with control mode during one period (two steps). In this figure, the control modes 1, 2, 3 and 4 correspond to *swing I, swing II, swing III and support*, respectively. In Fig. 4.3, large torque is observed at the end of the swing phase and the beginning of the support phase. This torque might be caused by too large or too small torque applied at the beginning of the swing phase. If the

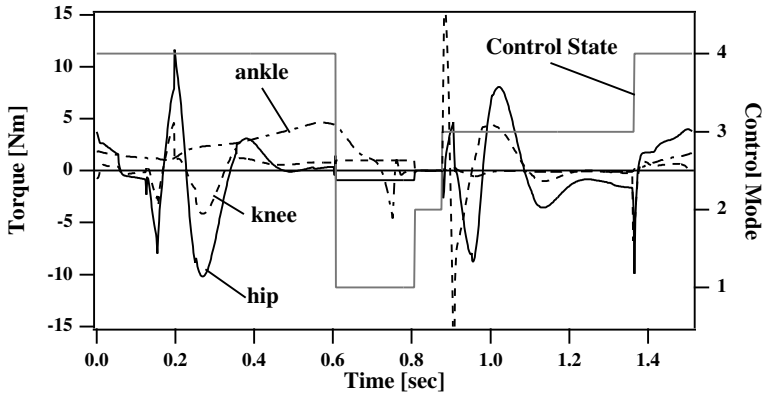


FIGURE 4.3: State machine mode and torque during one period

appropriate torque is applied in *swing I* (at the beginning of the swing phase), this feedback torque might be lessen and the more energy-efficient walking could be realized. In the next section, the optimization of this torque is attempted by adding a learning module.

4.2 Energy minimization with a learning module

To realize the energy efficient walking, a learning module which searches appropriate output torque in *swing I* is added to the controller described in the previous section (Fig. 4.4). Besides torque, the learning module searches the appropriate value of control parameter which determines the end of the duration of passive movement, T_{swg2}. It is noted that these parameters are not related to the PD controller which stabilizes walking. For the evaluation of energy efficiency, we use the average of all the torque which is applied during one walking period (two steps),

$$Eval = \frac{1}{T_{step}} \int_0^{T_{step}} \sum_{i=1}^{3} \tau_i dt \tag{4.10}$$

Using this performance function, the appropriate values of the parameters are searched in the probabilistic ascent algorithm as follows.

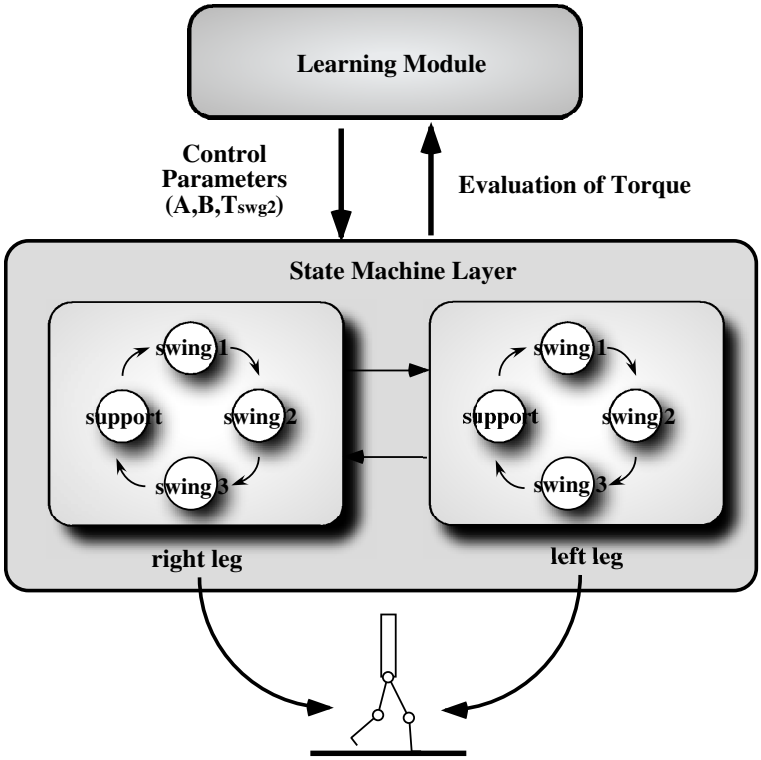
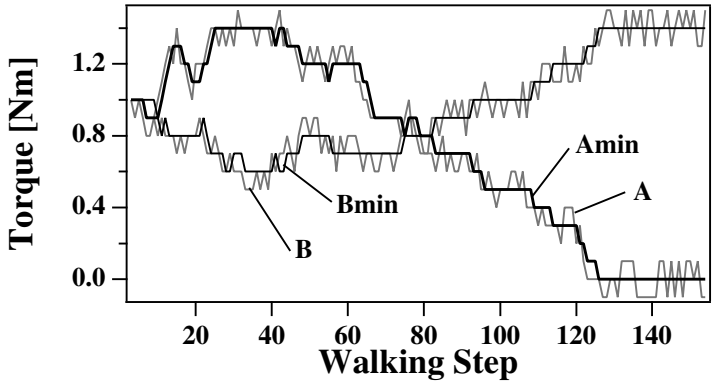


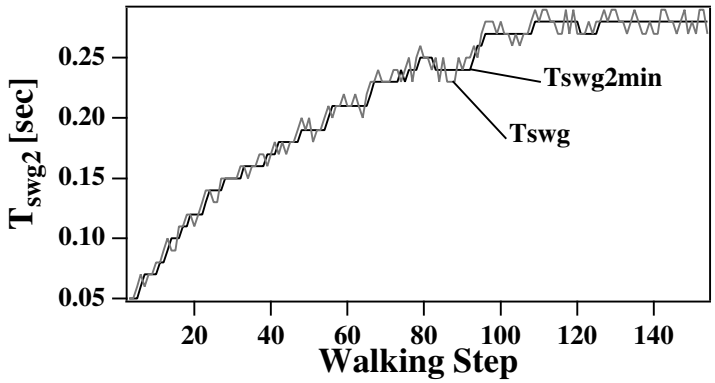
FIGURE 4.4: Ballistic walking with learning module

$if(Eval < Eval_{min})$
    $A_{min} = A$
    $B_{min} = B$
    $T_{swg2min} = T_{swg2}$
$A = A +$ random perturbation
$B = B +$ random perturbation
$T_{swg2} = T_{swg2} +$ random perturbation

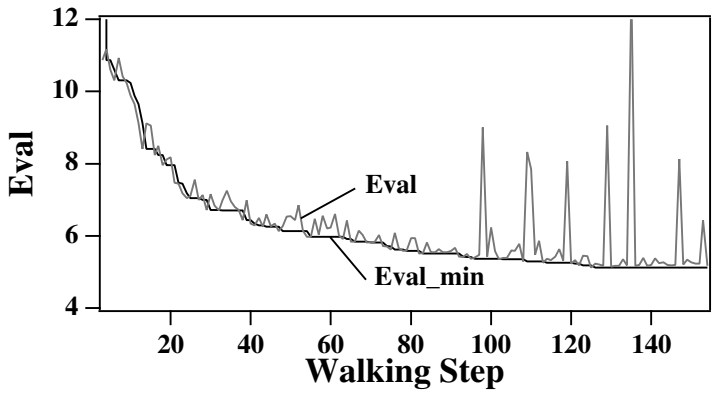
The simulation results are shown in Fig. 4.5. Figures. 4.5 (a), (b) and (c) show the time courses of the output torque applied to the hip and knee joints in *swing I*, A, B, and the passive time, T_{swg2}, and the average of total torque, $Eval$, respectively. Even though the input torque changes variously, the PD controller in *swing III* which keeps the posture at ground contact constant realizes a stable walking.



(a) torque



(b) T_{swg2}



(c) Average of Total Torque

FIGURE 4.5: Learning curve of control parameters and total torque

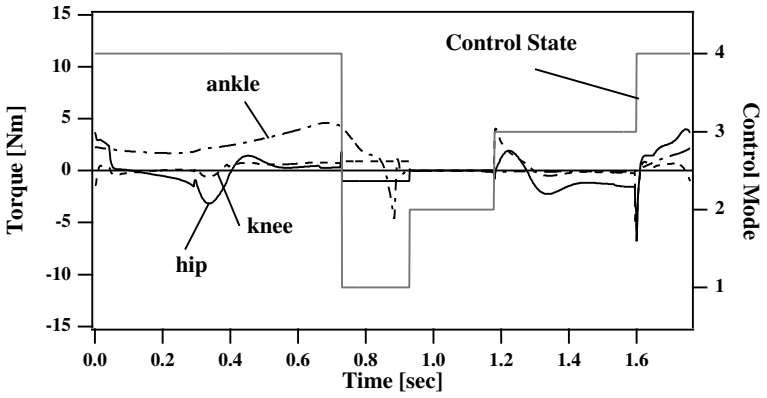


FIGURE 4.6: State machine mode and torque by a state machine controller with a learning module

Comparing the first step with the 80th one, the average of total torque decreases (Fig. 4.5 (c)), even though the output torque of the beginning of the swing phase at the 80th step is almost the same as the first step (Fig. 4.5 (a)), whereas the passive time, T_{swg2}, increases (Fig. 4.5 (b)). The total torque of walking, therefore, depends more on the passive time than the magnitude of the feedforward torque which is given in the beginning of the swing phase.

Furthermore, in the final stage of learning, after the 120th step, the output torque of the hip joint at the beginning in the swing phase becomes zero while the torque of the knee joint increases. This result might be strange because many researchers have applied torque to hip joint in swing phase. In this stage, the large energy output appears among weak ones (Fig. 4.5 (c)). This may be because robot walk on a wing and a prayer on the subtle balance between dynamics and energy. Once the balance is lost, the PD controller compensates stability with large torque.

Fig. 4.6 is the time-course of the torque in around the 80th step. Comparing the torque appeared in Fig. 4.6 with those in Fig. 4.3, the total torque are reduced about 1/10 in the hip and knee joints, whereas the torque profile at the ankle joint is almost the same.

4.3 Comparing with human data

In this section, we apply the proposed controller to the model which has the same mass and length of links as human, and the torque and angle of each link are compared with the observed data in

	Mass [kg]	Length [m]	Inertia [kg m^2]
HAT	46.48	0.542	3.359
Tigh	6.86	0.383	0.133
Shank	2.76	0.407	0.048
Foot	0.89	0.148	0.004

Table 4.1: Mass and length of human model links

	Human	Simulation
Support : Swing [%:%]	60:40	60:40
Walking Rate [steps/sec]	1.9	1.3
Walking Speed [m/sec]	1.46	0.46
Walking Step [m]	0.76	0.36
Energy Consumption [cal/m kg]	0.78	0.36

Table 4.2: Characteristics of simulation and human walking

human walking.

For the parameters of the human model, we use the same model as that of Ogihara and Yamazaki [52], which is shown in Table 4.1. The control gains at hip and knee joints are set as $K_p = 6000.0$ [Nm/rad], $K_v = 300.0$ [Nm sec/rad], $K_{wp} = 6000.0$ [Nm/rad] and $K_{wv} = 100.0$ [Nm sec/rad]. The desired angles at the end of the swing and support phases are the same as in Section 2.

The time course of angle and torque of the simulation results are shown in Figs. 4.7 with human walking data (from [80]). The horizontal axis is normalized by the walking period.

At the hip joint, while the time course of joint angle is almost same as human, that of torque is different, especially in around 80% and 30% walking periods in which strong effects of PD controllers appears (Fig. 4.7 (b)). At the knee joint, the pattern of the time course of joint angle roughly resembles human data in shape except at around the end of the swing phase and the beginning of the support phase, in which the knee joint of human data becomes straighten but that of simulation data not. Moreover, the torque pattern is quite different from human data. At the ankle joint, it is surprising that the torque pattern shares common traits with human data, even though

the ankle joint is modeled as simple spring joint. Fig. 4.7 (f) shows that, although the control state after the support phase is named "*swing I*", it works as double support phase. The rate of swing phase to support phase is the same as human data (40:60).

Table 4.2 compares characteristic features of walking in the simulation result with that in human data ([81]). It shows that the simulation algorithm succeeds in finding the parameters which enable the human model to walk with 45% less energy consumption. But this walk may not necessarily mean the energy efficient walking because the walking speed (and the walking rate) is much slower than human walking. This may be because the proposed controller uses the ankle joint only passively, and only the energy consumption is taken into consideration in the evaluation function (eq. 10). Acquiring fast walking is our future issue.

4.4 Discussion

Our controller has a state machine on each leg which affects each other by sensor signals. Even this simple controller enables a biped robot to walk stably. There are two reasons. First, PD controllers at the end of the swing phase ensure that a biped touches down on the ground with the same posture. This prevents a swing leg from contacting with too shorter or too longer step length because of inadequate forward torque given at the beginning of the swing phase. But this stabilization does not always work well. It mainly depends on the posture at ground contact. How this posture is determined is the issue we should attack next.

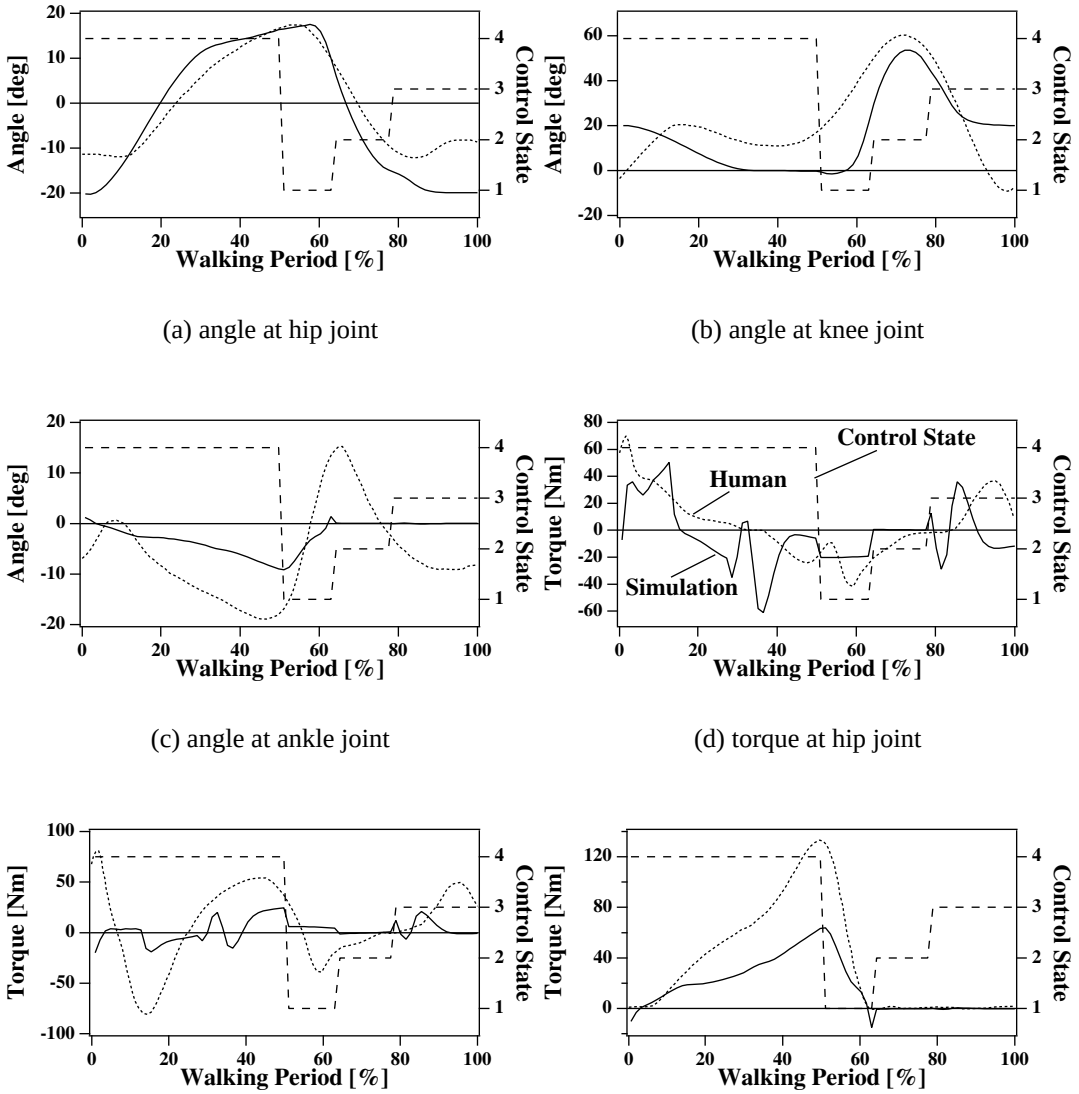
The second reason for stable walking is that the controller has some common features to CPG (Central Pattern Generator). In CPG model, the activities of neurons are affected by sensor signals (or environment), and as a result global entrainment between a neural system and the environment takes place [42] [43]. Our proposed controller doesn't have a walking period explicitly. The period of the controller is strongly affected by the information from touch sensors, which determine the state transition of a state machine in each leg. It can be said that our controller has some properties like global entrainment between the state machine controller and the environment.

The walking mode realized in this chapter is much slower than human walking as shown in Table 2. Presumably the reason of this slow walking is due to the passive use of the ankle joint. To realize fast walking, it is necessary to shorten the walking period and to make the step length

longer. They are closely related to the ankle joint setting because the speed of falling forward of the support leg is largely affected by the stiffness of the ankle joint, and the step length can be longer if the support leg rotates around the toe. Controlling the walking speed is another issue to be attacked.

Acknowledgments

This study was performed through the Advanced and Innovational Research program in Life Sciences from the Ministry of Education, Culture, Sports, Science and Technology, the Japanese Government.



(a) angle at hip joint

(b) angle at knee joint

(c) angle at ankle joint

(d) torque at hip joint

(e) torque at knee joint

(f) torque at ankle joint

FIGURE 4.7: Comparing with human walking data

Part II

Visuo-motor mapping in humanoid walking

Chapter 5

Related works in visuo-motor mapping

In part II, we will treat a sensori-motor mapping problem when generating humanoid behaviors, especially the mapping problem between visual information and various motions. This part is also related to the the problem on how to generate purposive behaviors with the dynamic walking.

This chapter introduces the existing approaches that accomplish purposive locomotive behaviors using visual information in a legged robot. This research area can also be categorized into two contrast approaches; model-based approaches and non-model-based approaches. Interestingly, these approaches are strongly correlated with the approaches to realize the biped walking. This is partly because a designer in the model-based approach tries to accomplish humanoid behaviors after modeling a robot, a controller and the environment completely, whereas a designer in the non-model-based approach regards the unpredictable emergence of entrainment between the dynamics of a robot and the environment. In the followings in this chapter, first the typical researches in the model-based approach are introduced. Next the attempts in the non-model-based approach are presented. Finally, the motivation and the rough idea of our approach are explained.

5.1 The model-based approach

Supposing that all information concerning the interaction between a robot and the environment is available in advance, a designer can predict the dynamics of the whole system and build a controller based on prediction. Conversely, a designer processes sensor information so that he/she can comprehend the whole system. It can be said that the model-based approach is the approach

that uses the frame of reference of a designer (who lives outside an agent).

Take an example task of approaching to the goal that can be captured in a robot camera, and see how sensor information are described in the model-based approach. In the first step, the two dimensional coordination of the goal in camera image is converted to the three dimensional data in the coordinate system that centers on a camera using camera parameters. In the next step, the coordinates are converted to that in the coordinate system that centers on a robot compensating the orientation of a head and eyes of a robot. Using this information about the distance and the direction of the goal, the robot motion to approach to the goal is planned. The trajectories of each joint to realize the planned motion is calculated, and finally those trajectories are realized based on the high gain feedback (Fig. 5.1).

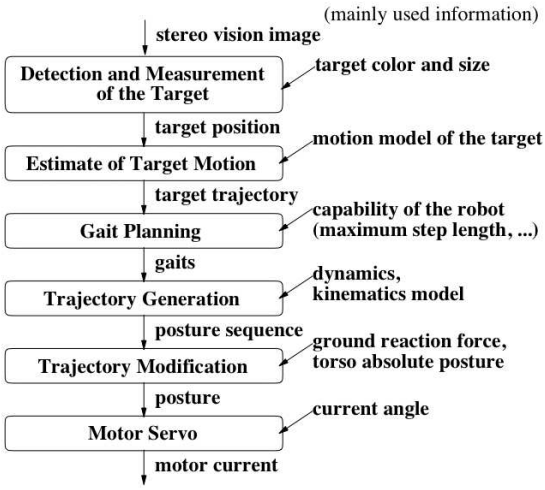


FIGURE 5.1: The typical controller in the model-based controller: from [82]

This control flow is the typical series-type controller that Brooks has criticized for its slow execution. However, the recent progress in computer technology improves the calculation performance and enables this approach. Almost all humanoid projects adopt this approach.

Nishiwaki et al. [82] achieve the approaching of a humanoid to the moving ball. Their controller describes the ball position in the global coordinate system and predicts its motion there. The walking motion is planned based on the prediction. Kagami et al. [83] propose the 2.5 dimensional description of the floor (Fig. 5.3). In their system, the stereo images of a humanoid are converted

to the texture in the global coordinate system using the camera parameters. The resultant map helps a humanoid to plan walking for avoiding obstacles.

FIGURE 5.2: A humanoid robot can chase a moving ball by predicting its motion.: from [82]

One of the serious problems in the model based approach is the error that arises from various reasons (e.g. the disagreement between the planned motion and the real motion, between the model parameters and those in a real robot). Seara et al. [84, 85] propose the gaze control system for a humanoid based on information criterion. Their system determines the motor command for gazing so that the uncertainty arising from the errors are minimized. However, their system is limited to gaze control in moving along the pre-determined course and does not refer to the problem how to determine humanoid walking motions.

The common features in this approach can be summarized as follows,

- Visual information is converted and described in the coordinate system that a designer can understand.
- The perception system and the controlling system for motion are developed separately and

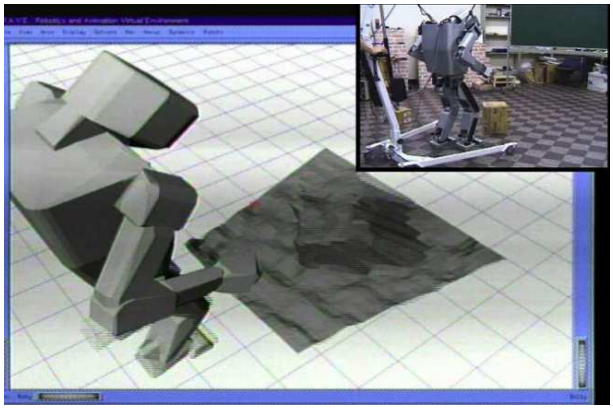


FIGURE 5.3: 2.5D terrain modeling: from [83]

connected based on a designer's interpretation.

Under the known and controlled circumstances, that is when all the information about a robot and the environment are available, this approach is powerful and effective. However, in such a system, any uncertainty causes the collapse of the whole system. How can a robot coordinate their motion behaviors by itself? The non-model-based approach is an approach to cope with that problem.

5.2 The non-model-based approach

When the information about a robot and the environment is not available, it is difficult to convert the information about motions and sensors to the frame of a designer's reference. In stead of calibrating those data for converting to the designer's view, it is necessary to calibrate the motion data with the sensor data and vise verse. This correlating task without a designer's view is sensori-motor mapping.

The controllers in the non-model-based approach has been proposed in a wheel-type robot. Optical flow has been used to learn the sensorimotor mapping for obstacle avoidance planned by the learned forward model [86] or by finding obstacles that show different flows from the environments using reinforcement learning [87]. However, it is difficult to directly apply those methods to a legged robot, because there are much more degrees of freedom and sensors in a

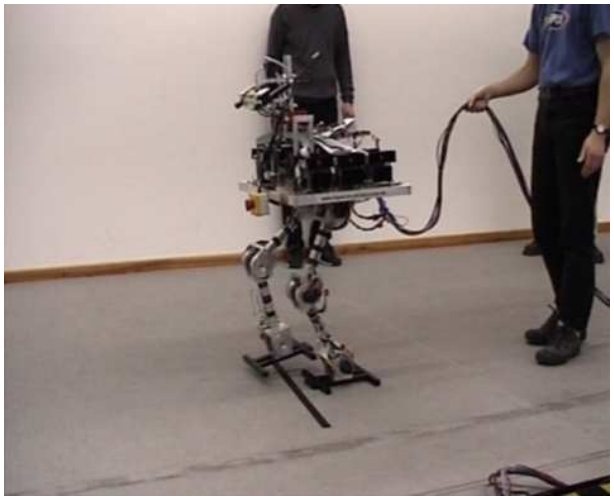


FIGURE 5.4: The robot traverses a obstacle by recognizing and localizing an obstacle: from [84]

legged robot, and motion commands are not so simple as in a wheel robot. In a legged robot, as the first step, it is important to think about how to construct the motion primitives that is preferable to make a sensori-motor mapping with sensors.

Miyashita et al. [88, 89] proposed a system which realizes the stable visual tracking in a 4-legged robot based on the various kinds of reflective motions for avoidance of falling down. They construct the motion repertories from the reflective motions that are generated in the visual tracking, and take the correlations with the visual information with those repertories. This methodology is very unique in that even the motions units are not given by a designer in advance.

As biological-inspired model, CPG is supposed to be the motion unit for locomotion, and several models are proposed to correlate the parameters of CPG neuron models and visual information.

Ijspeert and Arbib [90] propose the CPG model of locomotion and swimming for a salamandar. They showed that the simple input to the CPG neurons can change the direction of locomotion. Using this property, they demonstrate that the lamprey model can chase the given target in a computer simulation (Fig. 5.6).

For a humanoid model, Taga [91] shows that the temporal input to the CPG neurons in certain phase of walking can activate the motions for obstacle avoidance by changing the step length and the height of the foot. de Rugy et al. [92] showed that CPG model can adjust walking automatically so that a human model can step on a target point when the estimated time for the foot to reach the

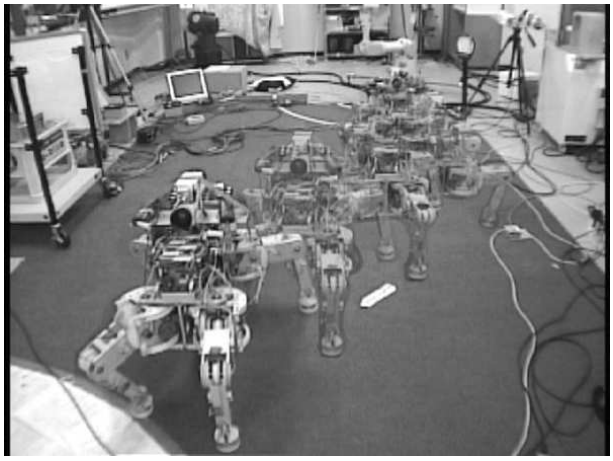


FIGURE 5.5: Quadruped walking with reflexes: from [89]

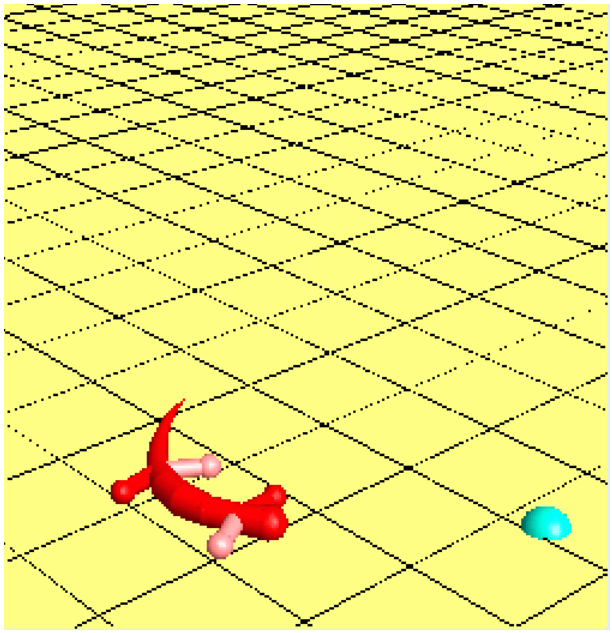


FIGURE 5.6: A Salamandar model chases a randomly moving target: from [90]

given target point is input to the CPG neurons.

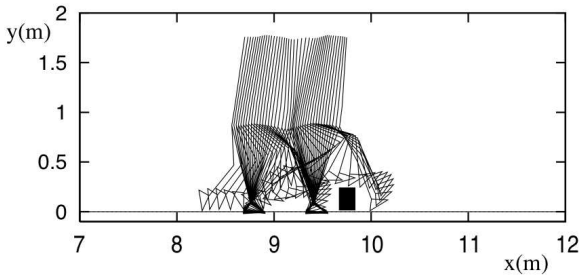


FIGURE 5.7: A human model can change its gait with simple modification of inputs to CPG neurons.: from [91]

The proposed models for CPG above mentioned are all demonstrated in computer simulation. The application of CPG to a robot is conducted by Kimura et al. [93]. They realize to avoid obstacles or climb over a step utilizing visual information in the same way as the methodology of Taga.

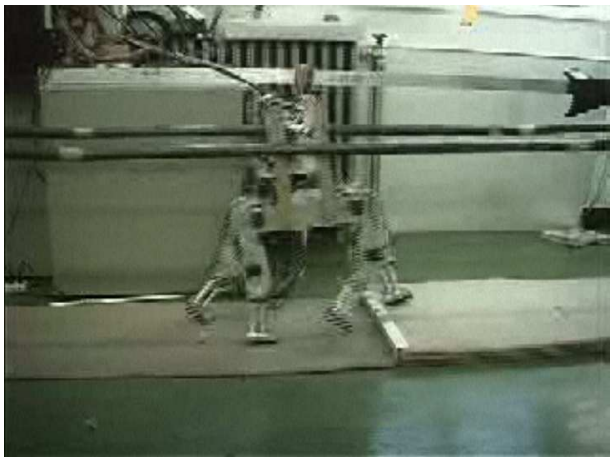


FIGURE 5.8: A quadruped robot can walk over a step by changing the CPG oscillation pattern with visual information.: from [91]

In the non-model-based approach, in spite of the possible performance, the research stage is not so adaptive that an agent adjusts automatically and adaptively the parameters of motions depending on its own experience. In the CPG-based approach, although it is shown that the sensor input can

change the locomotive motion, Currently, it seems that the adjustments of the sensor input is given by the designer.

5.3 Learning visuo-motor mapping in humanoid behavior generation

The purpose of my researches in part II is to realize humanoid motions based on learning visuo-motor mapping.

Chapter 6 introduces a layered controller, in which the lower-layer controller realizes rhythmic walking based on the controller proposed by Tsuchiya et al. [76] and the upper-layer controller learns the parameters of the lower-layer controller based on visual information. There are three points in learning the upper-layer controller: (1) in the first stage, it learns the feasible parameters of the lower-layer controller that enable a robot to walk, (2) to accelerate the learning process, the upper-layer controller learns the model of the world: the relationship between the control parameters given to the lower rhythmic walking controller and the change of the visual sensor information, and (3) the upper-layer controller learns which parameters should be given to reach a goal by reinforcement learning.

Chapter 7 presents a method of visuo-motor learning for behavior generation of humanoids, and, as an example task, passing a ball between two different humanoids (face-to-face pass) [94] is realized based on the sensorimotor mappings of motion primitives. The task is decomposed into three basic motion modules: *trapping a ball*, *approaching to a ball* and *kicking a ball to the opponent*. Each motion module can be further decomposed into several motion primitives, each of which has motion parameters to control the motion trajectory. The sensorimotor mapping is learned as the forward and inverse relationships between these motion parameters and optic flow information in each motion. The acquired sensorimotor maps are used to select the appropriate motion primitive and its parameters to realize the desired pathway or destination in the robot's view given by the planner.

Chapter 6

Reinforcement Learning of Humanoid Rhythmic Walking Parameters based on Visual Information

Abstract

This chapter presents a method for learning the parameters of rhythmic walking to generate purposive humanoid motions. The controller consists of two layers: rhythmic walking is realized by the lower layer, which adjusts the speed of the phase on the desired trajectory depending on sensory information, and the upper layer learns (1) the feasible parameter sets that enable stable walking, (2) the causal relationship between the walking parameters here to be given to the lower-layer controller and the change in the sensory information, and (3) the feasible rhythmic walking parameters by reinforcement learning so that a robot can reach to the goal based on visual information. The experimental results show that a real humanoid learns to reach the ball and to shoot it into the goal in the context of the *RoboCupSoccer* competition, and the further issues are discussed.

6.1 A RHYTHMIC WALKING CONTROLLER

6.1.1 A biped robot model

Fig. 7.1 shows a biped robot model used in the experiment which has a one-link torso, two four-link arms, and two six-link legs. All joints rotate with a single degree-of-freedom (DoF). Each foot has four force-sensing-resistor (FSR) sensors to detect reaction force from the floor, and a CCD camera with a fish-eye lens is attached at the top of the torso.

6.1.2 A rhythmic walking controller based on CPG principle

We build a lower-layer controller based on one proposed by Tsuchiya et al. [76]. The controller consists of two sub-controllers: *a trajectory controller* and *a phase controller* (see Fig. 6.2). The trajectory controller outputs the desired trajectory of each limb depending on the phase that is given by the phase controller. The phase controller consists of four oscillators, each of which is responsible for the movement of each limb (see Fig. 6.3). Each oscillator changes its speed depending on the touch sensor signal, and the effect is reflected on the oscillator in each limb. As a result, the desired trajectory of each joint is adjusted so that the global entrainment of dynamics between the robot and the environment takes place. In the following, the details of each controller are given.

Trajectory controller

The trajectory controller calculates the desired trajectory of each joint depending on the phase given by the corresponding oscillator in the phase controller. Four parameters characterize the trajectory of each joint as shown in Fig. 6.4. For joints 3, 4 and 5, which coincide with pitch axis, the desired trajectory is determined so that in the swing phase the foot trajectory draws a ellipse that has the radii h in the vertical direction and β in the horizontal direction, respectively. For joints 2 and 6, which coincide with roll axis, the desired trajectory is determined so that the leg tilts from $-W$ to W relative to the vertical axis. The amplitude of the oscillation, α, determines the desired trajectory of joint 1. The desired trajectories are summarized by the following functions:

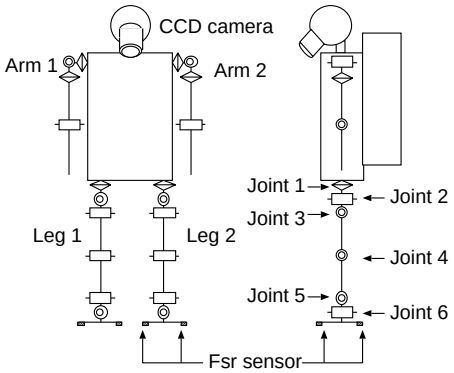


FIGURE 6.1: A model of biped locomotion robot

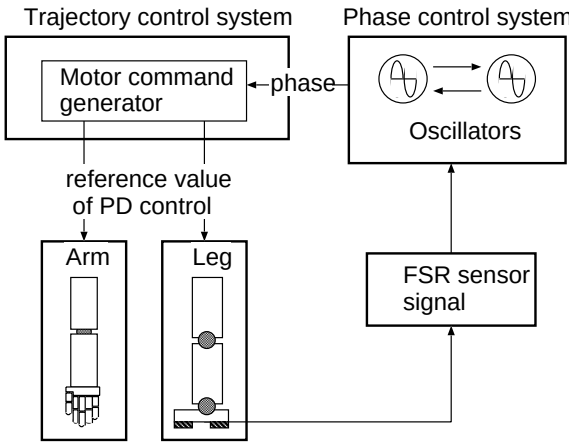


FIGURE 6.2: A walking control system

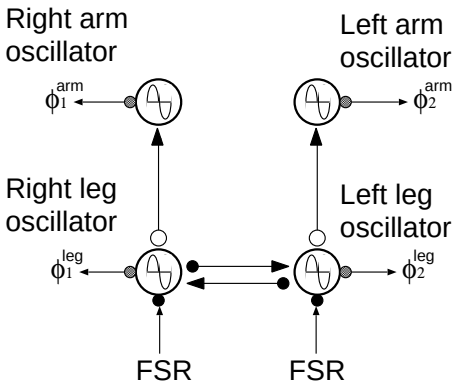
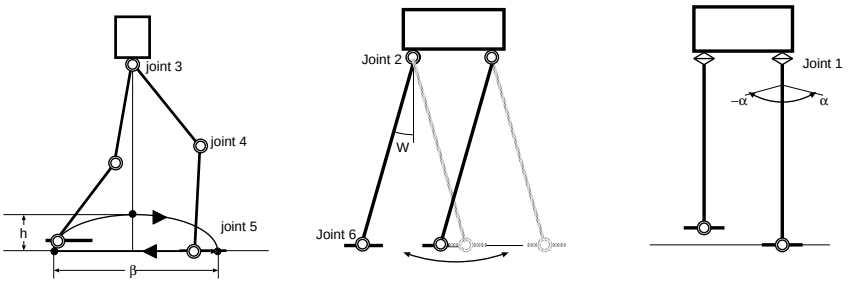


FIGURE 6.3: A phase control system



(a) Joint angles around pitch axis

(b) Joint angles around roll axis

(c) Joint angle around yaw axis

FIGURE 6.4: Joint angles

$$\theta_1 = \alpha \sin(\phi), \tag{6.1}$$

$$\theta_2 = W \sin(\phi), \tag{6.2}$$

$$\theta_i = f_i(\phi, h, \beta), \qquad (i = 3, 4, 5) \qquad \text{and} \tag{6.3}$$

$$\theta_6 = -W \sin(\phi). \tag{6.4}$$

The detail of f_i is explained in the Appendix. Among the four parameters described above, α, which determines the walking step length, and β, which determines the walking direction, are selected as rhythmic parameters of walking. Although these parameters characterize approximate direction and step length, they do not determine the resultant walking as precisely because of slippage between the support leg and the ground. These parameters are learned in the upper-layer learning module, which is described in **3**.

Phase controller

The phase controller sets the phase that determines the desired value of each joint. The phase controller consists of two oscillators, ϕ_R and ϕ_L, for the right and left leg, respectively. The dynamics of each oscillator is determined by the basic frequency, ω, the interaction term between

two oscillators, and the feedback signal from the sensory information,

$$\dot{\phi}_L = \omega - K(\phi_L - \phi_R - \pi) + g_L \tag{6.5}$$

$$\dot{\phi}_R = \omega - K(\phi_R - \phi_L - \pi) + g_R. \tag{6.6}$$

The second term on the RHS in the above equations ensures that the oscillators have opposite phases. The third term, feedback signal from sensor information, is given as follows:

$$g_i = \begin{cases} K' Feed_i & (0 < \phi < \phi_C) \\ -\omega(1 - Feed_i) & (\phi_C \leq \phi < 2\pi) \end{cases} \quad i = \{R, L\}, \tag{6.7}$$

where K', ϕ_C and $Feed_i$ denote feedback gain, the phase when the swing leg contacts with the ground, and the feedback sensor signal, respectively. $Feed_i$ returns 1 if the FSR sensor value of the corresponding leg exceeds a certain threshold value, otherwise 0. The third term in (5) and (6) ensures that the mode switching between the swing phase and the support phase happens appropriately according to the ground contact information from the FSR sensors. The values of parameters are set as follows: $\phi_C = \pi$ [rad], $\omega = 5.23$ [rad/sec], $K = 15.7$ and $K' = 1$.

6.2 REINFORCEMENT LEARNING WITH RHYTHMIC WALKING PARAMETERS

6.2.1 The principle of reinforcement learning

Reinforcement learning has been receiving increased attention as a method of robot learning with little or no *a priori* knowledge and a higher capability for reactive and adaptive behaviors. Fig. 6.5 shows a basic model of robot-environment interaction [95], in which a robot and environment are modelled by two synchronized finite state automatons interacting in a discrete time cyclical processes. The robot senses the current state $s_t \in \boldsymbol{S}$ of the environment and selects an action $a_t \in A$. Based on the state and action, the environment makes a transition to a new state $s_{t+1} \in \boldsymbol{S}$ and generates a reward r_{t+1} that is passed back to the robot. Through these interactions, the robot learns a purposive behavior to achieve a given goal. For the learning to converge correctly, the

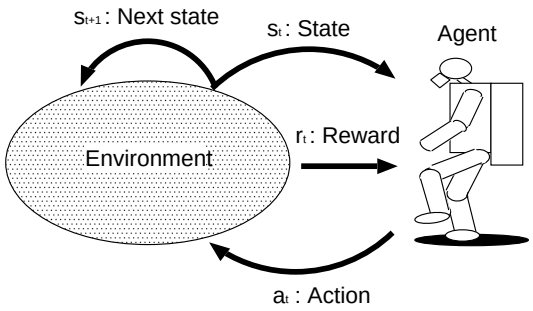


FIGURE 6.5: A basic model of agent-environment interaction

environment should satisfy the Markovian assumption that the state transition depends on only the current state and the action taken. A stochastic function $\boldsymbol{T}$ which maps a state-action pair to the next state ($\boldsymbol{T} : \boldsymbol{S} \times \boldsymbol{A} \rightarrow \boldsymbol{S}$) models the state transition. Using $\boldsymbol{T}$, the state transition probability $P_{s_t,s_{t+1}}(a_t)$ is given by

$$P_{s_t,s_{t+1}}(a_t) = Prob(\boldsymbol{T}(s_t, a_t) = s_{t+1}). \tag{6.8}$$

The reward function gives the immediate reward, r_t, in terms of the current state by $R(s_t)$, that is $r_t = R(s_t)$. Generally, $P_{s_t,s_{t+1}}(a_t)$ (hereafter $\mathcal{P}^a_{ss'}$) and $R(s_t)$ (hereafter $\mathcal{R}^a_{ss'}$) are unknown.

The aim of the reinforcement learner is to maximize the accumulated summation of the given rewards (called $return$) given by

$$return(t) = \sum_{n=0}^{\infty} \gamma^n r_{t+n}, \tag{6.9}$$

where $\gamma\,(0 \leq \gamma \leq 1)$ denotes a discounting factor to give the temporal weight to the reward.

If the state transition probability is known, the optimal policy that maximizes the expected $return$ is given by finding the optimal value function $V^*(s)$ or the optimal action value function $Q^*(s, a)$ as follows. Their derivation can be found elsewhere [95].

$$\begin{aligned} V^*(s) &= \max_a E\{r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a\} \\ &= \max_a \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V^*(s')\right] \end{aligned} \tag{6.10}$$

$$\begin{aligned} Q^*(s,a) &= E\{r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') | s_t = s, a_t = a\} \\ &= \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma \max_{a'} Q^*(s', a') \right] \end{aligned} \tag{6.11}$$

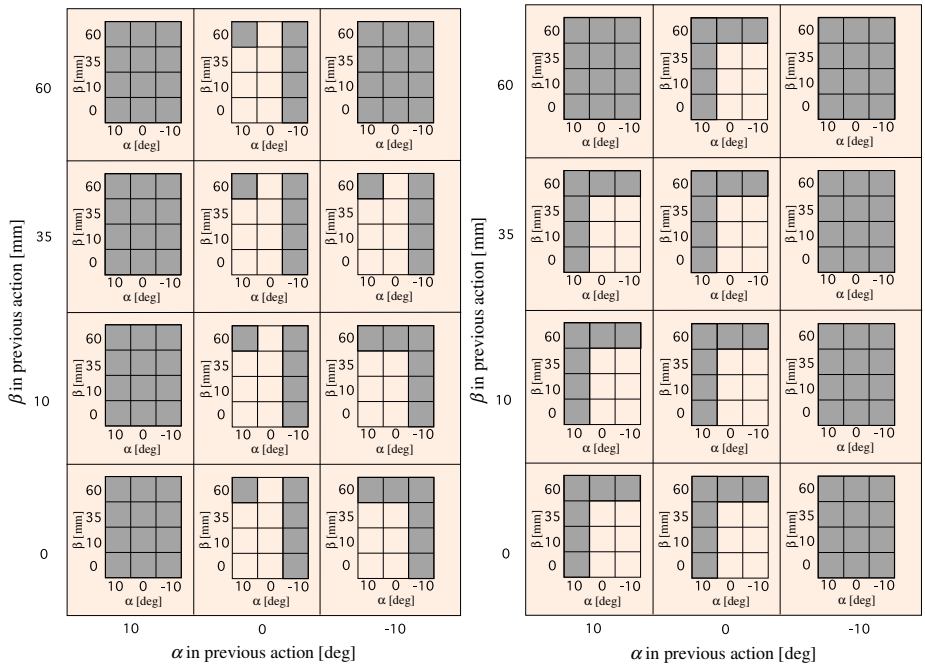
The learning module examines the state transition when both feet are in contact with the ground so that the stable visual information can be obtained (some experiments show the possibility that the human brain may calculate the distance to the obstacle by visual information at the double stance phase [96]). The state space s consists of the visual information s_v and the robot posture s_p, and the action is setting the two parameters of rhythmic walking. Details are explained in the following subsections.

6.2.2 Construction of action space based on rhythmic parameters

The learning process has two stages. The first stage constructs the action space consisting of feasible combinations of two rhythmic walking parameters ($\alpha,\ \beta$). To do that, we prepared the three-dimensional posture space s_p in terms of the forward length β (quantized into four lengths: 0, 10, 35 60 [mm]) and the turning angle α (quantized into three angles: -10, 0, 10 [deg]), which are the previous action command and the leg side (left or right). Therefore, we have 24 kinds of postures. First, we have constructed an action space of the feasible combinations of ($\alpha,\ \beta$) and excluded the infeasible combinations which cause collisions with its own body. Then, various combinations of actions are examined for stable walking in the real robot. Fig. 6.6 shows the feasible actions (empty boxes) for each leg corresponding to the previous actions. Owing to physical differences between the two legs, the constructed action space was not symmetric, although theoretically it should be.

6.2.3 Reinforcement learning with visual information

Fig. 6.7 shows an overview of the whole system, which consists of two layers: adjusting walking based on visual information and generating walking based on neural oscillators. The state space consists of the visual information s_v and the robot posture s_p, and adjusted action a is learned by a dynamic programming (DP) method based on the rhythmic walking parameters ($\alpha,\ \beta$). For the ball shooting task, s_v consists of ball substates and goal substates, which are quantized as shown



(a) left leg (b) right leg

FIGURE 6.6: Experimental result of action rule

in Fig. 6.8. We add two more substates, that is, “the ball is missing” and “the goal is missing” because they are necessary to recover from loosing sight of the ball or goal.

The learning module consists of a planner that determines an action a based on the current state s, a state transition model that estimates the state transition probability $\mathcal{P}^a_{ss'}$ through the interactions, and a reward model (see Fig. 6.9). Based on DP, the action value function $Q(s,a)$ is updated and the learning stops when there are no more changes in the summation of action values.

$$Q(s,a) = \sum_{s'} \mathcal{P}^a_{ss'}[\mathcal{R}_s + \gamma \max_{a'} Q(s',a')], \tag{6.12}$$

where $\mathcal{R}_s$ denotes the expected reward at the state s.

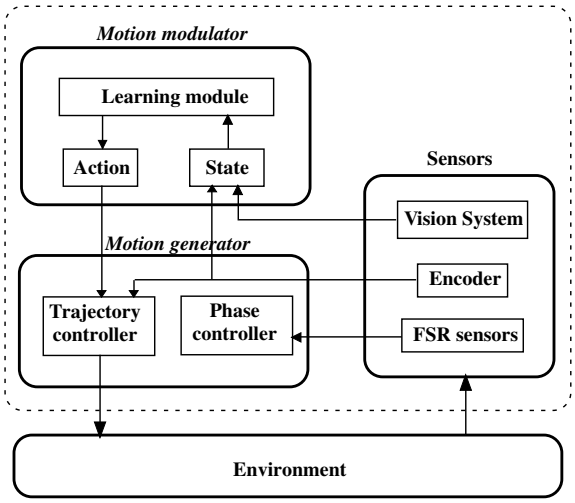
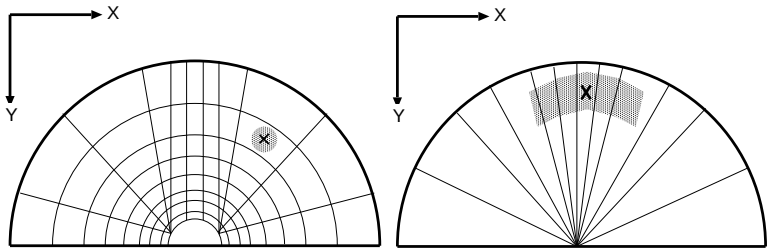


FIGURE 6.7: The biped walking system with visual perception



(a) The state space of ball (b) The state space of goal

FIGURE 6.8: The state space of ball and goal

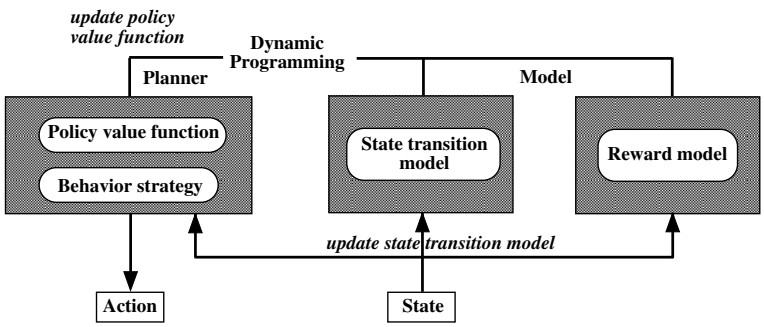


FIGURE 6.9: The learning module

6.3 EXPERIMENTS

6.3.1 A robot platform and environment set-up

We use a humanoid platform HOAP-1 by Fujitsu Automation Ltd. [97] attaching a CCD camera with a fish-eye lens at the head. Figs. 6.10 and 6.11 show a picture and a system configuration, respectively. The height and the weight are about 480 mm and 6 kg, and each leg has six degrees-of-freedom and each arm has four. Joint encoders have high resolution of 0.001 [deg/pulse] and reaction force sensors (FSRs) are attached to the soles. Color image processing to detect an orange ball and a blue goal is performed on the CPU (Pentium Ⅲ 800 MHz) under RT-Linux. Fig. 6.12 shows an on-board image.

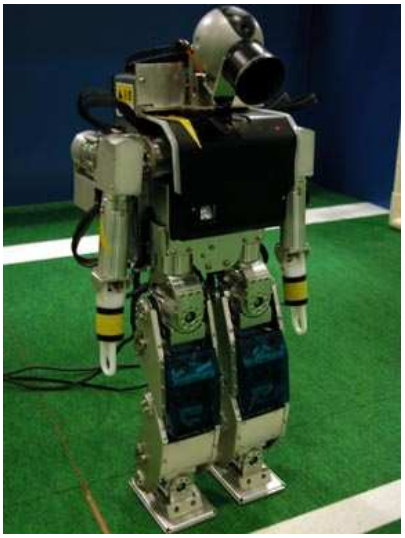


FIGURE 6.10: HOAP-1

The experimental set-up is shown in Fig. 6.13 where the initial robot position is inside the circle whose center and radius are the ball position and 1000 mm, respectively, and the initial ball position is located less than 1500 mm from the goal whose width is 1800 mm and height is 900 mm. The task is to take a position just before the ball so that the robot can shoot the ball into the goal. Each episode ends when the robot succeeds in getting such positions or fails (touches the ball or the pre-specified time period expires).

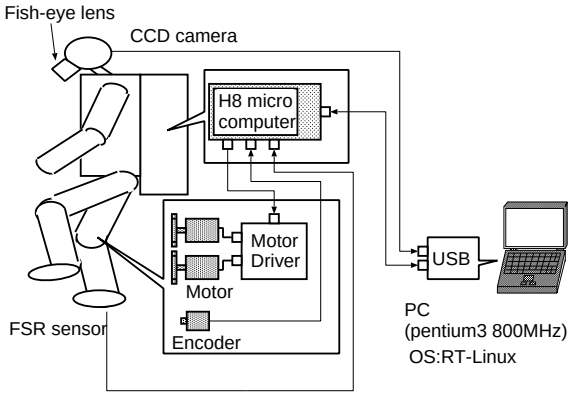


FIGURE 6.11: Overview of robot system

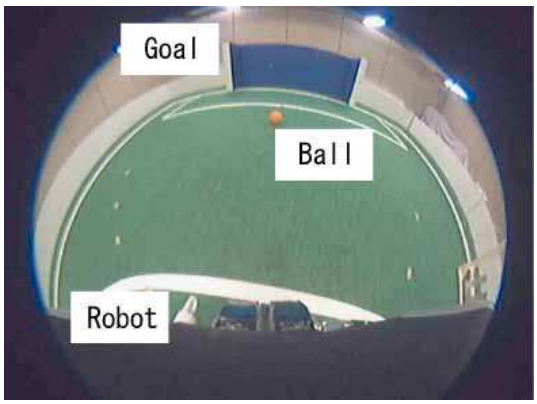


FIGURE 6.12: Robot's view (CCD camera image through fish-lens)

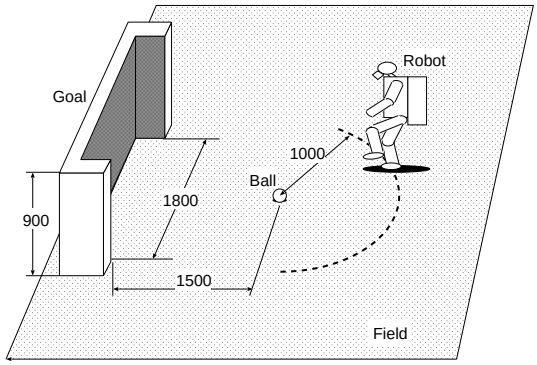


FIGURE 6.13: Experimental environment

6.3.2 Experimental results

One of the most serious issues in applying the reinforcement learning method to real robot tasks is how to accelerate the learning process. Instead of using Q-learning that is most typically used in many applications, we use a DP approach based on the state transition model $\mathcal{P}^a_{ss'}$, which is obtained separately from the learning behavior. Furthermore, we give the instructions to start up the learning: during the first 50 episodes (about half an hour), the human instructor avoids useless exploration by directly specifying the action command to the learner about 10 times per episode. After that, the learner experienced about 1500 episodes. Owing to the state transition model and initial instructions, learning converged in 15 hours, and the robot learned to get to the right position from any initial positions inside the half field.

Fig. 6.14 shows the learned behaviors from various initial positions. In Figs. 6.14 (a)-(e), the robot can capture the image including both the ball and the goal from the initial position while in Fig. 6.14 (f) the robot cannot see the ball or the goal from the initial position.

(a) Result 1 (b) Result 2 (c) Result 3

(d) Result 4 (e) Result 5 (f) Result 6

FIGURE 6.14: Experimental results

6.4 DISCUSSION

This study shows the possibilities for humanoid to correlate its walking parameters and on-board visual information through its experiences based on the so-called model-free approach which does not need very precise model parameters that are usually necessary for the model-based approach.

In our approach, motion commands are directly correlated with camera image without the complicated calibration process. It enables to evaluate the motion commands from the viewpoint of the achievement of the task.

This sort of approach has been already studied in wheeled robots [98]. It is necessary to keep stabilized walking to apply reinforcement learning to humanoid robot. There are two points to realize stable walking in this study. The first point is keeping the walking trajectory continuous when walking parameters are changed. To do that, the planned trajectory before the change is modified so that the effects on walking can be as little as possible. The second is the action rules described in Fig. 6. These rules impose constraints on the selection of action parameters. For example, a robot cannot select left turn command with long step length just after right turn command in the previous step.

There is still much room for improvement in this study as a model-free approach. One of the problems is learning time. In our experiments, although 1500 episodes are examined and convergence is conducted with the state transition probability acquired through those episodes, learning results are not completely optimal. For example, the selected step length is not maximum limit at the place where a robot is far from the goal place. Learning shows good convergence when the experimental setting is simplified: approach to the ball on a straight line. When a robot is far from the goal, the maximum step length is selected. This may be because the number of the states and actions in this simplified experiment is much smaller than that in the experiment of approaching to a ball from the various positions. Therefore, learning acceleration in the complicated environment is one of our future works.

6.5 CONCLUSION

Vision-based humanoid behavior was generated by reinforcement learning with rhythmic walking parameters. Since the humanoid generally has many DoFs, it is very hard to control all of them. Instead of using these DoFs in the action space, we adopted rhythmic walking parameters, which drastically reduce the search space and, therefore, real robot learning was possible in a reasonable amount of time. In this study, the designer specified the state space consisting of visual features and robot postures. State space construction by learning is an issue for future exploration.

Acknowledgments

We would like to thank Karl F. MacDorman for stimulating discussions and suggestions. This study was partially funded by the Advanced and Innovational Research Program in the Life Sciences of the Ministry of Education, Culture, Sports, Science, and Technology of the Japanese Government.

Chapter 7

Visuo-Motor Learning for Behavior Generation of Humanoids

Abstract

Humanoid behavior generation is one of the most formidable issues due to its many degrees of freedom. This chapter proposes a controller for a humanoid to cope with this issue. A given task is decomposed into a sequence of modules. Each of which consists of a set of motion primitives that have control parameters to realize the appropriate primitive motions. Then, these parameters are learned by sensori-motor maps between visual information (flow) and motor commands. The controller accomplishes a given task by selecting a module, a motion primitive in the selected module, and its appropriate control parameters learned in advance. A face-to-face ball pass in a RoboCup context (To the best of our knowledge, this is the first trial.) is chosen as an example task. The corresponding modules are approaching a ball, kicking a ball to the opponent, and trapping a ball coming to the player. In order to show the validity, the method is applied to two different humanoids, independently, and they succeeded in realizing the face-to-face pass more than three rounds. The rest of this chapter is organized as follows. Section 1 introduces an overview of our proposed system. Section 2 provides the details of each module for ”passing a ball” task. Section 3 shows experimental results of the task that need to use integrated modules. Finally discussions and concluding remarks are given.

7.1 Task, Robot, and Environment

7.1.1 Robot platforms

Fig. 7.1 shows biped robots used in the experiments, HOAP-1, HOAP-2, and their on-board views. HOAP-1 is 480 [mm] in height and about 6 [kg] in weight [97]. It has a one-link torso, two four-link arms, and two six-link legs. The other, HOAP-2 (a successor of HOAP-1), is 510 [mm] in height and about 7 [kg] in weight. It has two more joints in neck and one more joint at waist that HOAP-1 does not have. Both robots have four force sensing register (FSRs) in each foot to detect reaction force from the floor and a CCD camera with a fish-eye lens (HOAP-1) or semi-fish-eye lens (HOAP-2).

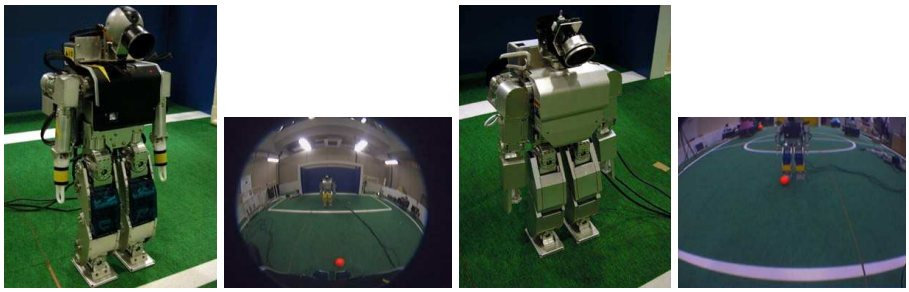


FIGURE 7.1: HOAP-1 with fish-eye lens and HOAP-2 with semi-fish-eye lens

These robots detect objects in the environments by colors. In this experiment, a ball is colored orange, and the knees of the opponent robot are colored yellow. The centers of these colored regions in the images are recorded as the detected positions.

7.1.2 Visuo-motor learning

Let the motion flow vector be $\Delta \boldsymbol{r}(t+1)$ at the position $\boldsymbol{r}(t)$ in the robot's view when a robot takes a motion, a. The relationship between them can be written,

$$\Delta \boldsymbol{r}(t+1) = f(\boldsymbol{r}(t), a(t)), \tag{7.1}$$

$$a(t) = g(\boldsymbol{r}(t), \Delta \boldsymbol{r}(t+1)), \tag{7.2}$$

where $\Delta \boldsymbol{r}(t+1)$ is difference between the current position vector $\boldsymbol{r}(t+1)$ and the previous position vector $\boldsymbol{r}(t)$. The latter function is useful to determine the motion parameters after planning the motion path way in the image. However, it is difficult to determine an unique motion to realize a certain motion flow because different motion primitives can produce the same image flow by adjusting motion parameters. Then, we separate the description of the relationship between the motion parameters in each primitive and the image flow as follows.

$$\boldsymbol{a}^i = (p_1^i, \ldots, p_n^i)^T = g_p^i(\boldsymbol{r}, \Delta \boldsymbol{r}) \tag{7.3}$$

$$\Delta \boldsymbol{r} = f^i(\boldsymbol{r}, \boldsymbol{a}^i), \tag{7.4}$$

$\boldsymbol{a}^i$ is a motion parameter vector of the i-th motion primitive. We use neural networks to learn these relationships.

7.1.3 Task and Assumptions

"Face-to-face pass" can be decomposed into following three modules:

1. approaching to a ball to kick,
2. kicking a ball to the opponent, and
3. trapping a ball which is coming to the player

All these basic modules need the appropriate prediction models between motion parameters and the environment changes. For example, to trap a ball in an appropriate way, the robots have to estimate the arrival time and the position of the coming ball. To approach to a kicking position, the robot should know the causal relationship between the walking parameters and the positional change of the objects in its image. Further, to kick a ball to the opponent, the robot must know the causal relationship between the kicking parameters and the direction into which the kicked ball will go.

Moreover, basic modules to realize these behaviors should be activated at the appropriate situations. Here, the designer determines such situations to switch the behaviors, and we focus on the module learning based on the optic flow information. Fig. 7.2 shows an overview of the proposed system.

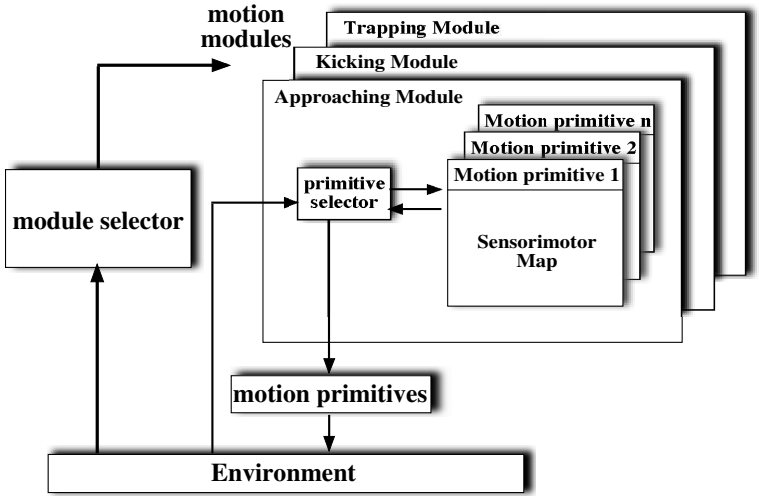


FIGURE 7.2: A system overview

7.2 Module Learning Based on Optic Flow Information

7.2.1 Ball Approaching

Approaching to a ball is the most difficult task among the three modules because this task involves several motion primitives each of which has parameters to be determined. These motions yield various types of image flows depending on the values of the parameters which change continuously. We make use of environmental image flow patterns during various motions to approach to the ball.

We separate the description of the relationship between the motion and the image flow into the relationship between the motion primitive and the image flow, and the relationship between the motion parameters in each primitive and the image flow (Fig. 7.3), as follows:

$$\boldsymbol{m} = g_m(\boldsymbol{r}, \Delta\boldsymbol{r}), \tag{7.5}$$

$$\boldsymbol{a}^i = (p_1^i, p_2^i)^T = g_p^i(\boldsymbol{r}, \Delta\boldsymbol{r}), \tag{7.6}$$

$$\Delta\boldsymbol{r} = f^i(\boldsymbol{r}, \boldsymbol{a}^i), \tag{7.7}$$

where $\boldsymbol{m}$ is a vector whose i-th element indicates the effectiveness of i-th motion primitive to generate the given flow, and $\boldsymbol{a}^i = (p_1^i, p_2^i)^T$ is the motion parameter vector of the i-th motion primitive. In this study, the motion primitives related to this module consists of 6 primitives; *forward walk* (left and right), *curve walk* (left and right), and *side step* (left and right). Each of the

primitives has two parameters which have real values, as shown in Fig. 7.4.

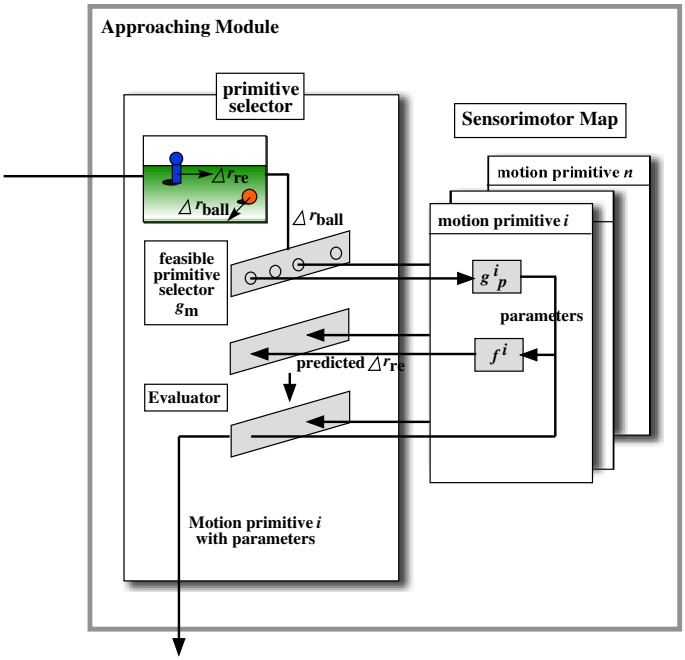


FIGURE 7.3: An overview of the approaching module

Given the desired motion pathway in the robot's view, we can select appropriate primitive by g_m, and determine the motion parameters of the selected motion primitive by g_p^i based on the learned relationships among the primitives, their parameters, and flows. If the desired image flow yields several motion primitives, the preferred motion primitive is determined by an evaluation function.

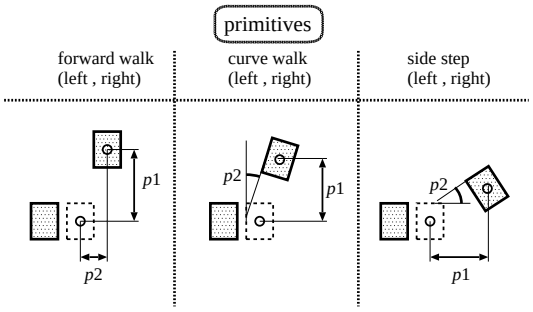


FIGURE 7.4: Motion primitives and parameters for approaching

Images are recorded every step and the image flow is calculated by block matching between the current image and the previous one. The templates for calculating flows are 24 blocks in one image as shown in Fig. 7.5.

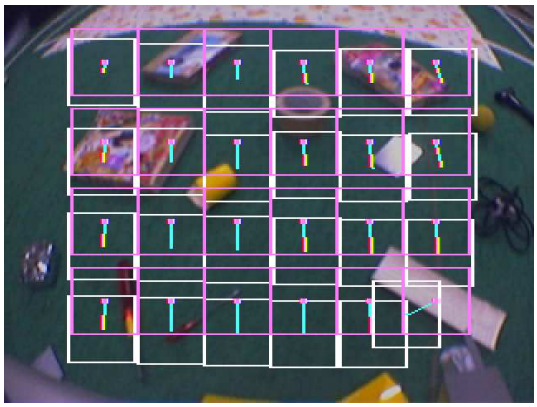


FIGURE 7.5: An example of an optic flow in the robot's view

Sensorimotor mapping between motion primitives and image flows: g_m

All of the data sets of the flow and its positional vector in the image, $(\boldsymbol{r}, \Delta\boldsymbol{r})$, are classified by the self organizing map (SOM), which consists of 225 (15×15) representational vectors. After organizing, the indices of motion primitives are attributed to each representational vector. Fig. 7.6 shows the classified image vector (the figure at the left side) and the distribution of each primitive in SOM. This SOM outputs the index of appropriate motion primitive so that the desired flow vector in the image is realized.

Sensorimotor mapping between the motion parameters and the image flows: f^i, g_p^i

The forward and inverse functions that correlates the relationship between the motion parameters in each primitive and the image flow, f^i, g_p^i, are realized by a simple neural network. The neural network in each primitive is trained so that it outputs the motion parameters when the flow vector and the positional vector in the image are input.

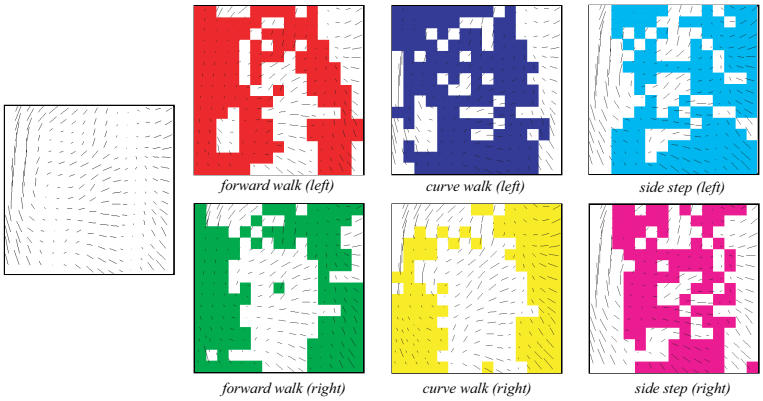


FIGURE 7.6: Distribution of motion primitives on the SOM of optic flows

Planning and evaluation function

In this study, the desired optic flows in the robot's view for the ball and the receiver, $\boldsymbol{s}_{ball}, \boldsymbol{s}_{re}$, are determined as vectors from the current position of a ball to the desired position (kicking position) in the robot's view, and as the horizontal vector from the current position to the vertical center line, respectively. The next desired optic flow of a ball to be realized, $\tilde{\boldsymbol{s}}_{ball}$, is calculated based on these desired optic flows,

$$n_{step} = \|\boldsymbol{s}_{ball}\|/\Delta r_{max}, \tag{7.8}$$

$$\tilde{\boldsymbol{s}}_{ball} = \boldsymbol{s}_{ball}/n_{step}, \tag{7.9}$$

where Δr_{max} is the maximum length of the experienced optic flow. This reference vector is input to the primitive selector, g_m, and the candidate primitives which can output the reference vector are activated. The motion parameters of the selected primitive are determined by the function g_p^i,

$$\boldsymbol{a}^i = g_p^i(\boldsymbol{r}_{ball}, \tilde{\boldsymbol{s}}_{ball}), \tag{7.10}$$

where $\boldsymbol{r}_{ball}$ is the current ball position in the robot's view. When the primitive selector outputs several candidates of primitives, the evaluation function depending on the task, $V(m_i)$, determines the preferred primitive. In this study, robots have to not only approach to a ball but also take an

appropriate position to kick a ball to the other. For that, we set the evaluation function as follows,

$$\begin{aligned} V(m_i) &= \|\tilde{\boldsymbol{s}}_{ball} - f^i(\boldsymbol{r}_{ball}, \boldsymbol{a}^i)\| \\ &\quad +k\|\mathbf{s}_{re} - n_{step} f^i(\boldsymbol{r}_{re}, \boldsymbol{a}^i)\|, \\ P &= \underset{i \in primitives}{\arg\min}\ V(m_i) \end{aligned} \tag{7.11}$$

where k is the constant value, $\boldsymbol{r}_{re}$ is the current position of the receiver in the robot's view, and P is the selected primitive.

Fig. 7.7 shows experimental results of approaching to a ball. A robot successfully approach to a ball so that the hypothetical opponent (a poll) comes in front of it.

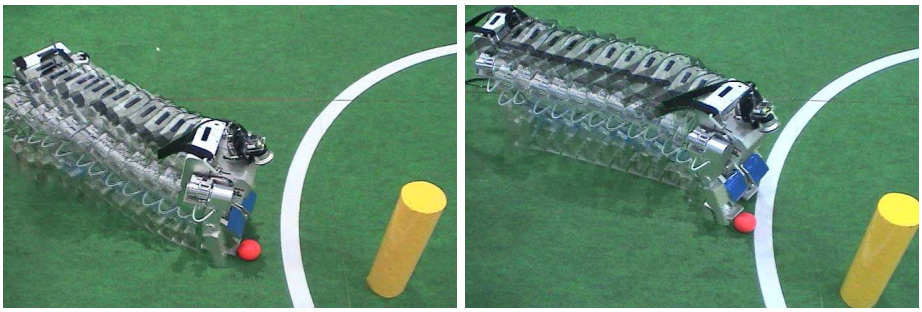


FIGURE 7.7: Experimental results of approaching to a ball

7.2.2 Ball Kicking to the Opponent

It is necessary for the robots to kick a ball to the receiver very precisely because they cannot sidestep quickly. We correlate the parameter of kicking motion with the trace of the kicked ball in the robot's view so that they can kick to each other precisely. Fig. 7.8 shows a proposed controller for kicking.

The kicking parameter is the hip joint angle shown in Fig. 7.9 (a). A quick motion like kicking changes its dynamics depending on its motion parameter. The sensor feedback from the floor reaction force sensors is used for stabilizing the kicking motion. The displacement of the position of the center of pressure (CoP) in the support leg is used as feedback to the angle of the ankle joint of the support leg (see Fig. 7.9 (b)). Fig. 7.9 (c) shows the effectiveness of the stabilization of the kicking motion.

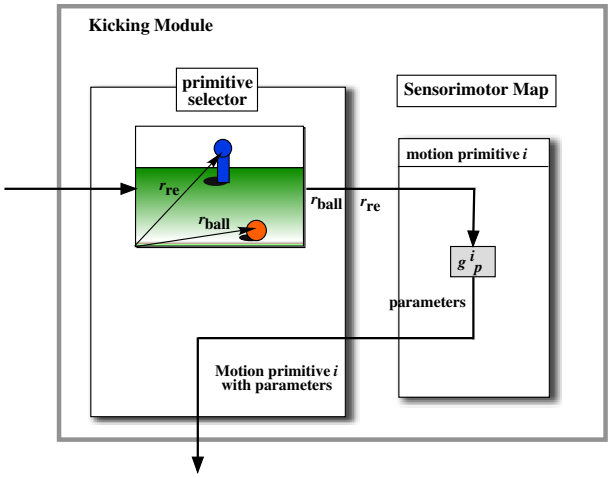
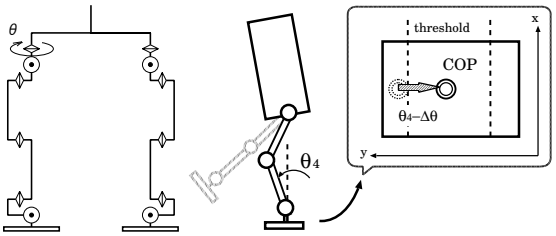
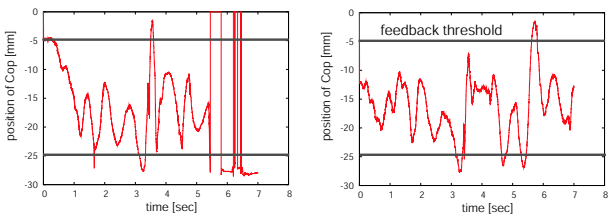


FIGURE 7.8: An overview of the kicking module



(a) Kick parameter

(b) An overview of stabilization of kick motion



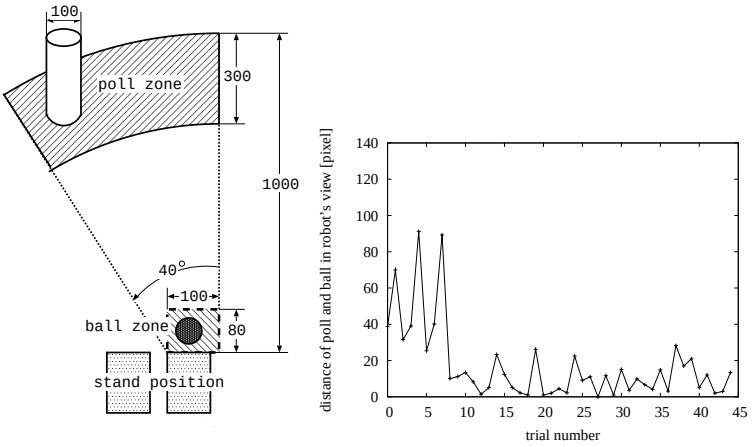
without feedback

with feedback

(c) The trajectories of CoP of the support leg during kicking motion

FIGURE 7.9: The parameter and the stabilization of kicking

The initial ball position and the parameter of the kicking motion affects sensitively the ball trace in the robot's view. To describe the relationship among them, we use a neural network, which is trained in the environment where the poll (10 [cm]) is put about 1 [m] in front of the robot (Fig. 7.10) (a)). The trace of the ball (the effects of the self motion is subtracted) is recorded every 100 [msec], and the weights in the neural network are updated every one trial. Fig. 7.10 (b) shows the time course of error distance between target poll position and kicked ball in the robot's view. It shows that the error is reduced rapidly within 20 [pixel], which is the same size of the width of the target poll. Fig. 7.11 shows the kicking performance of the robot.



(a) The environmental setting

(b) The error of distance during the learning process for kicking

FIGURE 7.10: The environmental setting and the learning curve for kicking

FIGURE 7.11: An experimental result of kicking a ball to the poll

7.2.3 Ball Trapping

Fig. 7.12 shows an overview of trapping module. Robots learn the relationship between the position of the foot in robot's view and the trap parameter which affects the position of the foot, to acquire the skill to trap a coming ball.

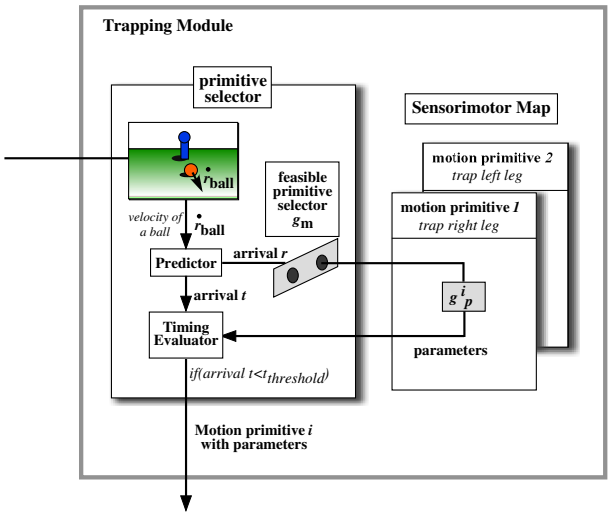


FIGURE 7.12: An overview of the trapping module

Fig. 7.14 shows the trapping motion by HOAP-2 acquired by the method described below. In order to realized such a motion, the robot has to predict the position and the arrival time of a ball from its optic flow captured in the robot view. For that purpose, we use a neural network which learns the causal relationship between the position and an optic flow of the ball in visual image of a robot and the arrival position and time of the coming ball. This neural network is trained by the data in which a ball is thrown to a robot from the various positions. Fig. 7.13 shows several prediction results of the neural network after learning. Δx [pixel] and Δt [sec] indicates the errors of the arrival position and the time predicted at each point (every 0.3 [sec]) in the robot's view. T denotes a duration of the ball rolling. Based on this neural network, the robots can activate the trapping motion primitive with the appropriate leg (right or left) at the appropriate timing (Fig. 7.14).

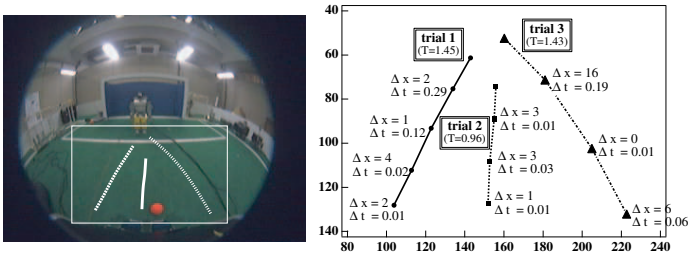


FIGURE 7.13: The prediction of the position and time of a coming ball

FIGURE 7.14: An experimental result of a trapping module

7.3 Integration of the Modules for Face-to-face Pass

To realize passing a ball between two humanoids, the basic modules described in the previous sections are integrated by a simple rule as shown in Fig. 7.15.

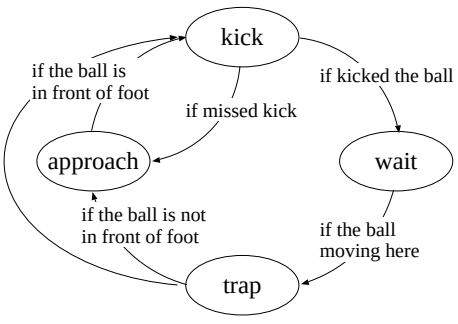


FIGURE 7.15: The rule for integrating motion modules

Fig. 7.16 shows the experimental result. Two humanoids with different body and different camera lens realize the appropriate motions for passing a ball to each other based on their own sensorimotor mapping. The passing lasts more than 3 times.

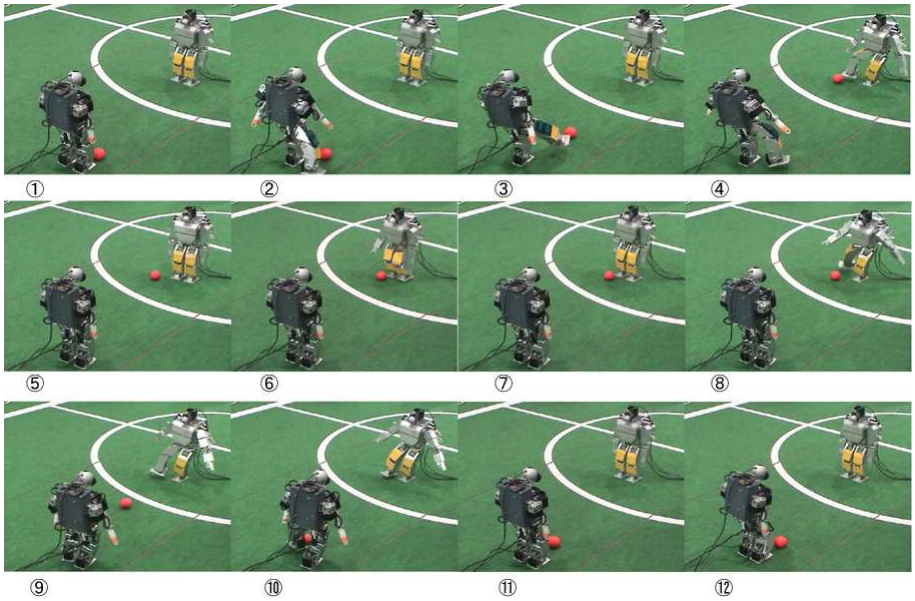


FIGURE 7.16: An experimental result of passes between two humanoids

7.4 Discussion

A humanoid has many degrees of freedom, which makes it more difficult to make motions than in wheel robots. In this chapter, we encapsulated humanoid behaviors into motion modules, each of which consists of multiple motion primitives. This method can be related to the biological neural circuits for locomotion, CPG (Central Pattern Generator) [5, 39]. In cats, neurons for various types of locomotions are found in the midbrain and it is believed that the deviations of one locomotion is performed by the input from the higher central nervous system. Thus, nature seems to solve the problem of DoFs to some extent in the same way as the proposed model in this chapter. The model for locomotion that is more realistic as biological model are proposed by Taga [91]. They demonstrate that the simple input to the CPG circuits makes it possible for a humanoid model to walk over obstacles in the computer simulation environment.

The proposed approach is also plausible as a biological model in the view point of the calibration (calibrating of the motor command by a robot's self sensors and vice versa). This method is more effective and simpler than the calibration method by a designer's view because the relationships between motor commands and sensor values are directly related to the tasks. And

the proposed method is much more efficient than the direct reinforcement learning approach. In the previous chapter, we realized the ball approaching task by reinforcement learning, where the combinations of action and sensor values are directly related to the performance measure. Even though the forward model was used for evaluating the state action evaluation function, the learning process was much more unefficient, because the state for visual information was described by the combination of target goal and the ball position. That caused the problem that a robot must try the all configurations of the positions ball and goal. On the other hand, the proposed method in this chapter describes the forward model of the general object in the view. Furthermore that model is learned by the optic flows of general scene during walking, and a robot does not need to try all the configurations of relative positions of a ball and a goal, which saves much learning time.

However, there remain the harder problems we skip in this chapter as a learning approach. First is module decomposition problem, that is how to determine what are the basic modules for the given task. Second is planning, that is how to organize each motion primitive to achieve the given task. In this chapter, we assume module decomposition and planning are given in advance. Combining the learning in each module level with that in higher level is our future issue.

7.5 Conclusions

In this chapter, the robots learn the sensorimotor mapping between optic flow information and their own motion parameters. Acquiring basic modules for passing a ball is achieved using the sensorimotor mapping. In each module, optic flow information is correlated with the motion parameters. Through this correlation, a humanoid robot can obtain the sensorimotor mapping to realize the desired modules. The experimental results show that a simple neural network quickly learns and models well the relationship between optic flow information and motion parameters of each motion primitive.

List of Publications

Articles in Journals

- 荻野 正樹, 春名 正樹, 俵 和史, 細田 耕, 浅田稔: 受動歩行からヒューマノイド歩行に向けて，バイオメカニズム 16, バイオメカニズム学会編集, 東京大学出版会, pp. 223-230, 2002.

- Masaki Ogino, Yutaka Katoh, Masahiro Aono, Minoru Asada and Koh Hosoda: Reinforcement Learning of Humanoid Rhythmic Walking Parameters based on Visual Information, Advanced Robotics, Vol. 18, No. 7, pp. 677-697, 2004

- Masaki Ogino, Masaaki Kikuchi and Minoru Asada: Visuo-motor learning for face-to-face pass between two different humanoids, Robotics and Autonomous Systems, submitted.

Papers in Proceedings of International Conferences

- Masaki Ogino, Koh Hosoda, and Minoru Asada: Acquiring Passive Dynamic Walking Based on Ballistic Walking. Proceedings of the Fifth International Conference on Climbing and Walking Robots, pp.139-146, 2002.

- Masaki Ogino, Koh Hosoda, and Minoru Asada: Learning Energy Efficient Walking with Ballistic Walking. Proceedings of the 2nd International Symposium on Adaptive Motion of Animals and Machines, CDROM, ThP-I-5, 2003.

- Masaki Ogino, Koh Hosoda, and Minoru Asada: Learning Energy Efficient Walking Based

on Ballistics. Proceedings of SICE Annual Conference 2003 in Fukui, Vol.CDROM, pp.3064-3069, 2003.

- Masaki Ogino, Yutaka Katoh, Masahiro Aono, Minoru Asada, and Koh Hosoda: Vision-Based Reinforcement Learning for Humanoid Behavior Generation with Rhythmic Walking Parameters. Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1665–1671, 2003.

- Masaaki Kikuchi, Masaki Ogino, and Minoru Asada: Visuo-Motor Learning for Behavior Generation of Humanoids, Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 521-526, 2004.

- Masaki Ogino, Masaaki Kikuchi, Jun'ichiro Ooga, Masahiro Aono, and Minoru Asada: Optic Flow Based Skill Learning for A Humanoid to Trap, Approach to, and Pass a Ball. RoboCup 2004 Symposium papers and team description papers, CD-ROM, 2004.

References

[1] R. Pfeifer and C. Scheier. *Understanding Intelligence* (Bradford Books, 2001).

[2] K. Osuka. *From model based control to dynamics based control - 'explict model' and 'implicit model' in robotics*. ISCIE Journal Systems, Control and Information **43**(2), 94 (1999).

[3] K. Osuka. *Dynamics based control of mechanical systems*. In *Proceedings of 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '01)*, pp. 566–570 (2001).

[4] H. F. R. Prechtl and B. Hopkins. *Developmental transformations of spontaneous movements in early infancy*. Early Human Development **14**, 233 (1986).

[5] S. Grillner. *Neurobiological bases of rhythmic motor acts in vertebrates*. Science **228**, 143 (1985).

[6] G. Taga, R. Takaya, and Y. Konishi. *Analysis of general movements of infants towards understanding of developmental principle for motor control*. In *Proceedings of the 1999 IEEE SMC*, pp. 678–683 (1999).

[7] T. Yamaski, T. Nomura, and S. Sato. *Possible functional roles of phase resetting during walking*. Biological Cybernetics **88**, 468 (2003).

[8] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi. *Humanoid robot HRP-2*. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pp. 1083–1090 (2004).

[9] J. Yamaguchi, E. Soga, S. Inoue, and A. Takanishi. *Development of a bipedal humanoid robot -control method of whole body cooperative dynamic biped walking-*. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pp. 368–374 (Detroit, Michigan, 1999).

[10] Y. Okumura, T. Furuta, T. Tawara, and H. Kitano. *Powered-passive dynamics as the third mode of dynamics in humanoid biped locomotion*. In *Proceedings of the 2001 IEEE-RAS International Conference on Humanoid Robots*, pp. 421–423 (2001).

[11] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. *The development of Honda humanoid robot*. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pp. 1321–1326 (1998).

[12] Y. Kuroki, T. Ishida, and J. Yamaguchi. *A small biped entertainment robot*. In *proceedings of IEEE-RAS International Conference on Humanoid Robot*, pp. 181–186 (2001).

[13] *Toyota motor corporation*. In *http://www.toyota.co.jp/jp/special/robot/*.

[14] A. Takanishi, M. Ishida, Y. Yamazki, and I. Kato. *The realization of dynamic wakling by the biped robot WL-10RD*. In *Proceedings of the International Conference on Advanced Robotics*, pp. 459–466 (Tokyo, 1985).

[15] M. Vukobratovic, B. Borovac, and D. Surdilovic. *Zero-moment point — proper interpretation and new applications*. In *Proceedings of the 2nd IEEE-RAS International Conference on Humanoid Robots, Humanoids 2001*, pp. 237–244 (Waseda University, Tokyo, Japan, 2001).

[16] M. Vukobratovic and B. Borovac. *Zero-moment point — thirty five years of its life*. International Journal of Humanoid Robotics **1**(1), 157 (2004).

[17] A. Goswami. *Postural stability of biped robots and the foot rotation indicator (FRI) point*. International Journal of Robotics Research **18**(6) (1999).

[18] J. Yamaguchi, N. Kinoshita, A. Takanishi, and I. Kato. *Development of a dynamic biped walking system for humanoid –development of a biped walking robot adapting to the human's*

living floor–. In *Proc. of 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 232–239 (1996).

[19] J. Yamaguchi and A. Takanishi. *Development of a biped walking robot having antagonistic driven joints using nonlinear spring mechanism*. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, pp. 185–192 (Albuquerque, New Mexico, 1997).

[20] J. Yamaguchi, D. Nishino, and A. Takanishi. *Realization of dynamic biped walking varying joint stiffness using antagonistic driven joints*. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pp. 2022–2028 (Leuven, Belgium, 1998).

[21] F. Miyazaki and S. Arimoto. *A control theoretic study on dynamica. biped locomotion*. Transactions ASME Journal of Dynamic Systems, Measurements and Control **102**(4), 233 (1980).

[22] H. Miura and I. Shimoyama. *Dynamic walk of a biped*. The International Journal of Robotics Research **3**(2), 60 (1984).

[23] J. Furusho and M. Masubuchi. *Control of a dynamical biped locomotion system for steady walking*. Transactions ASME Journal of Dynamic Systems, Measurements and Control **108**(2), 111 (1986).

[24] S. Kajita and K. Tani. *Adaptive gait control of a biped robot based on realtime sensing of the ground profile*. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 570–577 (1996).

[25] S. Kajita, O. Matsumoto, and M. Saigo. *Real-time 3D walking pattern generation for a biped robot with telescopic legs*. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pp. 2299–2036 (Seoul, Korea, 2001).

[26] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. *The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation*. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 239–246 (Maui, Hawaii, USA, 2001).

[27] G. T. Fallis. U.S. Patent No. 376, 588 (1888).

[28] T. McGeer. *Passive dynamic walking*. International Journal of Robotics Research **9**(2), 62 (1990).

[29] T. McGeer. *Passive walking with knees*. In *Proceedings of 1990 IEEE Robotics and Automation Conference*, pp. 1640–1645 (Cincinnati, OH, 1990).

[30] M. J. Coleman, A. Chatterjee, and A. Ruina. *Motions of a rimless spoked wheel: a simple 3d system with impacts*. Dynamics and Stability of Systems **12**(3), 139 (1997).

[31] M. Garcia, A. Chatterjee, A. Ruina, and M. Coleman. *The simplest walking model: Stability, complexity, and scaling*. ASME Journal of Biomechanical Engineering **120**(2), 281 (1998).

[32] M. Garcia, A. Chatterjee, and A. Ruina. *Speed, efficiency, and stability of small-slope 2-d passive dynamic*. In *Proceedings of the 1998 IEEEInternational Conference on Robotics and Automation*, pp. 2351–2356 (Leuven, Belgium, 1998).

[33] A. Goswami, B. Thuilot, and B. Espiau. *A study of the passive gait of a compass-like biped robot: symmetry and chaos*. International Journal of Robotics Research **17**(12), 1282 (1998).

[34] M. Garcia, A. Chatterjee, and A. Ruina. *Efficiency, speed, and scaling of passive dynamic bipedal walking*. Dynamics and Stability of Systems **15**(2), 139 (2000).

[35] K. Osuka and Y. Saruta. *Developoment and control of new legged robot quartet iii —from active walking to passive walking*. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 991–995 (2000).

[36] S. H. Collins, M. Wisse, and A. Ruina. *A three-dimensional passive-dynamic walking robot with two legs and knees*. The International Journal of Robotics Research **20**(7), 607 (2001).

[37] Y. Sugimoto and K. Osuka. *Walking control of quasi-passive-dynamic-walking robot 'quartet iii' based on delayed feedback control*. In *Proceedings of 5th International Conference on Climbing and Walking Robots*, pp. 123–130 (Paris, France, 2002).

[38] F. Asano, M. Yamakita, and K. Furuta. *Virtual passive dynamic walking and energy-based control laws*. In *Proceedings of 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1149–1154 (2000).

[39] S. Grillner. *Neural networks for vertebrate locomotion*. Scientific American pp. 48–53 (1996).

[40] K. Matsuoka. *Mechanisms of frequency and pattern control in the neural rhythm generators*. Biological Cybernetics **56**, 345 (1987).

[41] G. Taga, Y. Yamaguchi, and H. Shimizu. *Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment*. Biological Cybernetics **65**, 147 (1991).

[42] G. Taga. *A model of the neuro-musculo-skeletal system for human locomotion: I. emergence of basic gait*. Biological Cybernetics **73**(2), 97 (1995).

[43] G. Taga. *A model of the neuro-musculo-skeletal system for human locomotion: II. real-time adaptability under various constraints*. Biological Cybernetics **73**(2), 113 (1995).

[44] R. Delcomyn. *Walking robots and the central and peripheral control of locomotion in insects*. Autonomous Robots **7**, 259 (1999).

[45] A. J. Ijspeert and J. Kodjabachian. *Evolution and development of a central pattern generator for the swimming of a lamprey*. Artificial Life **5**, 247 (1999).

[46] A. J. Ijspeert. *A connectionist central pettern generator for the aquatic and terrestrial gaits of a simulated salamander*. Biological Cybernetics **84**(5), 331 (2001).

[47] H. Kimura, K. Sakamura, and S. Akiyama. *Dynamic walking and running of the quadruped using neural oscillator*. In *Proceedings of IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pp. 50–57 (1998).

[48] H. Kimura and Y. Fukuoka. *Adaptive dynamic walking of a quadruped robot on irregular terrain by using neural system model*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. T–MI–8–2 (Takamatsu, Japan, 2000).

[49] H. Kimura, Y. Fukuoka, K. Konaga, Y. Hada, and K. Takase. *Towards 3D adaptive dynamic walking of a quadruped robot on irregular terrain by using neural system model*. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pp. 2312–2317 (Hawaii, USA, 2001).

[50] H. Kimura and Y. Fukuoka. *Biologically inspired adaptive dynamic walking in outdoor environment using a self-contained quadruped robot: 'tekken2'*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 986–981 (Sendai, Japan, 2004).

[51] S. Miyakoshi, G. Taga, Y. Kuniyoshi, and A. Nagakubo. *Three dimensional bipedal stepping motion using neural oscillators –towards humanoid motion in the real world–*. In *Proceedings of 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 84–89 (1998).

[52] N. Ogihara and N. Yamazaki. *Generation of human bipedal locomotion by a bio-mimetic neuro-musculo-skeletal model*. Biological Cybernetics **84**, 1 (2001).

[53] K. Endo, T. Maeno, and H. Kitano. *Co-evolution of morphology and walking pattern of biped humanoid robot using evolutionary computation -consideration of characteristic of the servomotors-*. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2678–2683 (2002).

[54] A. Fujii, A. Ishiguro, and P. Eggenberger. *Evolving a CPG controller for a biped robot with neuromodulation*. In *Proceedings of the Fifth International Conference on Climbing and Walking Robots*, pp. 17–24 (2002).

[55] T. Reil and P. Husbands. *Evolution of central pattern generators for bipedal walking in a real-time physics environment*. IEEE Transactions on Evolutionary Computation **6**(2), 159 (2002).

[56] C. Paul. *Bilateral decoupling in the neural control of biped locomotion*. In *Proceedings of the 2nd International Symposium on Adaptive Motion of Animals and Machines*, pp. CD–ROM (Kyoto, Japan, 2003).

[57] C. Paul, H. Yokoi, and K. Matsushita. *Design and control of humanoid robot locomotion with passive legs and upper body*. In *Proceedings of the International Symposium on Robotics*, pp. CD–ROM (Paris, France, 2003).

[58] A. J. Ijspeert, J. Nakanishi, and S. Schaal. *Learning rhythmic movements by demonstration using nonlinear oscillators*. In *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, pp. 958–963 (2002).

[59] A. J. Ijspeert, J. Nakanishi, and S. Schaal. *Movement imitation with nonlinear dynamical systems in humanoid robots*. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1398–1403 (2002).

[60] M. Lungarella and L. Berthouze. *On the interplay between morphological, neural, and environmental dynamics: A robotic case study*. Adaptive Behavior **10**(3-4), 223 (2002).

[61] S. Mochon and T. A. McMahon. *Ballistic walking*. Jounal of Biomechanics **13**, 49 (1980).

[62] S. Mochon and T. A. McMahon. *Ballistic walking: An improved model*. Mathematical Bioscience **52**, 241 (1981).

[63] T. A. McMahon. *Muscles, reflex, and locomotion*. (Princeton University Press, Princeton, NewJersey, 1984).

[64] R. Q. van der Linde. *Actively controlled ballistic walking*. In *Proceedings of the IASTED International Conference Robotics and Applications 2000*, pp. 1–8 (2000).

[65] R. Q. vand der Linde. *Passive bipedal walking with phasic muscle contraction*. Biological Cybernetics **81**, 227 (1999).

[66] R. Q. van der Linde. *Active leg compliance for passive walking*. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pp. 2339–2344 (1998).

[67] R. Q. van der Linde. *Design, analysis, and control of a low power joint for walking robots, by phasic activation of McKibeen muscles*. IEEE Transactions on Robotics and Automation **15**(4), 599 (1999).

[68] M. Wisse, A. L. Schwab, and R. Q. vd. Linde. *A 3D passive dynamic biped with yaw and roll compensation*. Robotica **19**, 275 (2001).

[69] M. Wisse and J. van Frankenhuyzen. *Design and construction of MIKE; a 2D autonomous biped based on passive dynamic walking*. In *Proceedings of 2nd International Symposium on Adaptive Motion of Animals and Machines*, pp. CD–ROM (WeP–I–1) (Kyoto, Japan, 2003).

[70] J. E. Pratt and G. A. Pratt. *Exploiting natural dynamics in the control of a 3D bipedal walking simulation*. In *Proceedings of International Conference on Climbing and Walking Robots (CLAWAR99)*, pp. CD–ROM (Portsmouth, UK, 1999).

[71] J. Pratt. *Exploiting Ingerent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots*. Ph.D. thesis, MIT (2000).

[72] K. Ono, R. Takahashi, and T. Shimada. *Self-excited walking of a biped mechanism*. The International Journal of Robotics Resarch **20**(12), 953 (2001).

[73] K. Ono, R. Takahashi, A. Imadu, and T. Shimada. *Self-excitation control for biped walking mechanism*. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1143–1148 (2000).

[74] K. Ono, T. Furuichi, and R. Takahashi. *Self-excited walking of a biped mechanism with feet*. The International Journal of Robotics Research **23**(1), 55 (2004).

[75] K. Tsuchiya, K. Tsujita, K. Manabu, and S. Aoi. *An emergent control of gait patterns of legged locomotion robots*. In *Proceedings of the Symposium on Intelligent Autonomous Vehicles*, pp. 271–276 (2001).

[76] K. Tsuchiya, S. Aoi, and K. Tsujita. *Locomotion control of a biped locomotion robot using nonlinear oscillators*. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1745–1750 (2003).

[77] A. M. Formal'sky. *Ballistic Locomotion of a Biped Design and Control of Two Biped Machines* (CISM:Springer-Verlag, Udine, Italy, 1997).

[78] A. Formal'sky, C. Chevallereau, and B. Perrin. *On ballistic walking locomotion of a quadruped*. The International Journal of Robotics Resarch **19**(8), 743 (2000).

[79] R. Featherstone. *The calculation of robot dynamics using articulated-body inertia*. The International Journal of Robotics Research **2**(1), 13 (1983).

[80] D. Winter. *Kinematic and kinetic patterns of human gait; variability and compensating effects*. Human Movement Science **3**, 51 (1984).

[81] A. Shumway-Cook and M. H. Woollacott. *Motor Control: theory and Practical Applications* (Williams & Wilkins, 1995).

[82] K. Nishiwaki, S. Kagami, J. Kuffner, M. Inaba, and H. Inoue. *Walking control system of a humanoid for tracking a moving object with estimate of the target motion*. In *Proceedings of IEEE International Conference on Humanoid Robots*, pp. CD–ROM (2003).

[83] S. Kagami, K. Nishiwaki, J. J. Kuffner, K. Okada, M. Inaba, and H. Inoue. *Vision-based 2.5d terrain modeling for humanoid locomotion*. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pp. 2141–2146 (Taipei, Taiwan, 2003).

[84] J. F. Seara, K. Strobl, and G. Schmidt. *Information management for gaze control in vision guided biped walking*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 31–36 (2002).

[85] J. F. Seara, K. H. Strobl, and G. Schmidt. *Path-dependent gaze control for obstacle avoidance in vision guided humanoid walking*. In *Proceedings of the IEEE International Conference on Robotics and Automation* (Taipei, Taiwan, 2003).

[86] K. F. MacDorman, K. Tatani, Y. Miyazaki, M. Koeda, and Y. Nakamura. *Protosymbol emergence based on embodiment: Robot experiments*. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1968–1974 (2001).

[87] T. Nakamura and M. Asada. *Motion sketch: Acquisition of visual motion guided behaviors*. In *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 127–132 (1995).

[88] K. Hosoda, T. Miyashita, and M. Asada. *Emergence of quadruped walk by a combination of reflexes*. In *Proceedings of the International Symposium on Adaptive Motion of Animals and Machines*, p. CDROM (2000).

[89] T. Miyashita, K. Hosoda, and M. Asada. *Motion repertory for a legged robot from a reflective walk*. vol. 19, pp. 855–862 (2001).

[90] A. J. Ijspeert and M. Arbib. *Visual tracking in simulated salamander locomotion*. In *Proceedings of the 6th International Conference on the Simulatation of Adaptive Behavior (SAB2000)* (2000).

[91] G. Taga. *A model of the neuro-musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance*. Biological Cybernetics **78**, 9 (1998).

[92] A. de Rugy, G. Taga, G. Montagne, M. J. Buekers, and M. Laurent. *Modeling human locomotor pointing: From optics to the neural basis of locomotion*. Biological Cybernetics **87**, 141 (2002).

[93] H. Kimura, S. Akiyama, and K. Sakurama. *Realization of dynamic walking and running of the quadruped using neural oscillator*. Autonomous Robots **7**(3), 247 (1999).

[94] H. Kitano and M. Asada. *The robocup humanoid challenge as the millennium challenge for advanced robotics*. Advanced Robotics **13**(8), 723 (2000).

[95] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction* (MIT Press, 1998).

[96] M. Laurent and J. A. Thomson. *The role of visual information in control of a constrained locomotor task*. Jounal of Motor Behavior **20**, 17 (1988).

[97] Y. Murase, Y. Yasukawa, and K. Sakai. *Design of a compact humanoid robot as a platform*. In *Proceedings of 19th Conference of Robotics Society of Japan*, pp. 789–790 (2001).

[98] S. N. M. Asada, S. Tawaratumida, and K. Hosoda. *Purposive behavior acquisition for a real robot by vision-based reinforcement learning*. Machine Learning **23**, 279 (1996).

Appendix A

Planning the reference trajectory around the pitch axis

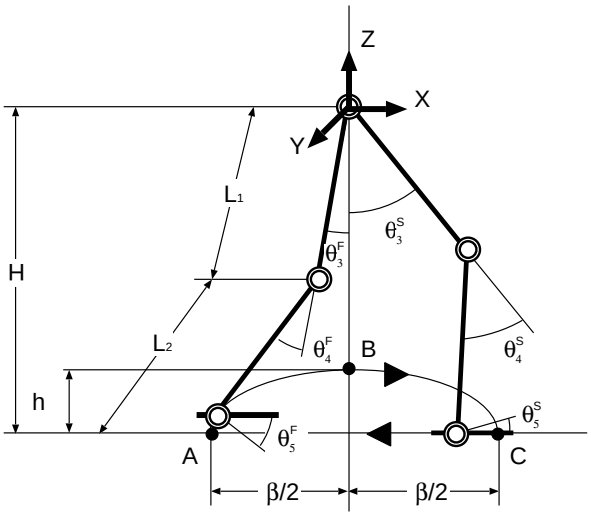


FIGURE A.1: Joint angles and the reference trajectory of the foot

The position of the foot determines the reference trajectories of joints 3, 4 and 5. Let x and z be the position of the foot in the plane XZ which is perpendicular to the pitch axis. The reference

trajectory of the foot is given by

$$x_F = -\frac{\beta}{2}\cos(\phi^F), \tag{A.1}$$

$$z_F = -H + h\sin(\phi^F), \tag{A.2}$$

$$x_S = -\frac{\beta}{2}\cos(\phi^S), \tag{A.3}$$

$$z_S = -H, \tag{A.4}$$

$$\tag{A.5}$$

where (x_F, z_F) and (x_S, z_S) are the positions of the foot in the swing and support phase, respectively, H is the length from the ground to the joint 3, β is the step length, and h is the maximum height of the foot from the ground (Fig. A.1). When the position of the foot is determined, the angle of each joint to be realized is calculated by the inverse kinematics as follows,

$$\theta_3 = \frac{\pi}{2} + atan2(z, x) - atan2(k, x^2 + z^2 + L_1^2 - L_2^2), \tag{A.6}$$

$$\theta_4 = atan2(k, x^2 + z^2 - L_1^2 - L_2^2), \tag{A.7}$$

$$\theta_5 = -(\theta_3 + \theta_4), \tag{A.8}$$

where k is given by the following equation,

$$k = \sqrt{(x^2 + z^2 + L_1^2 + L_2^2)^2 - 2\{(x^2 + z^2)^2 + L_1^4 + L_2^4\}}. \tag{A.9}$$

In this research, the value of each parameter is set as follows: $H = 185$ [mm], $h = 8$ [mm], $W = 13$ [deg], $L_1 = 100$ [mm] and $L_2 = 100$ [mm].