

é2.1429

UNIVERSITY OF STIRLING

**Fast Biped Walking with A  
Neuronal Controller and Physical  
Computation**

by

Tao Geng

A thesis submitted in partial fulfillment for the  
degree of Doctor of Philosophy

in the  
Faculty of Human Sciences  
Department of Psychology

April 2007

UNIVERSITY OF STIRLING

ABSTRACT

FACULTY OF HUMAN SCIENCES  
DEPARTMENT OF PSYCHOLOGY

Doctor of Philosophy

by Tao Geng

Biped walking remains a difficult problem and robot models can greatly facilitate our understanding of the underlying biomechanical principles as well as their neuronal control. The goal of this study is to specifically demonstrate that stable biped walking can be achieved by combining the physical properties of the walking robot with a small, reflex-based neuronal network, which is governed mainly by local sensor signals. This study shows that human-like gaits emerge without specific position or trajectory control and that the walker is able to compensate small disturbances through its own dynamical properties. The reflexive controller used here has the following characteristics, which are different from earlier approaches: (1) Control is mainly local. Hence, it uses only two signals (AEA=Anterior Extreme Angle and GC=Ground Contact) which operate at the inter-joint level. All other signals operate only at single joints. (2) Neither position control nor trajectory tracking control is used. Instead, the approximate nature of the local reflexes on each joint allows the robot mechanics itself (e.g., its passive dynamics) to contribute substantially to the overall gait trajectory computation. (3) The motor control scheme used in the local reflexes of our robot is more straightforward and has more biological plausibility than that of other robots, because the outputs of the motorneurons in our reflexive controller are directly driving the motors of the joints, rather than working as references for position or velocity control. As a consequence, the neural controller and the robot mechanics are closely coupled as a neuro-mechanical system and this study emphasises that dynamically stable biped walking gaits emerge from the coupling between neural computation and physical computation. This is demonstrated by different walking experiments using two real robot as well as by a Poincaré map analysis applied on a model of the robot in order to assess its stability. In addition, this neuronal control structure allows the use of a policy gradient reinforcement learning algorithm to tune the parameters of the neurons in real-time, during walking. This way the robot can reach a record-breaking walking speed of 3.5 leg-lengths per second after only a few minutes of online learning, which is even comparable to the fastest relative speed of human walking.

# Contents

<b>Acknowledgements</b>	<b>iv</b>
<b>Declaration</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Neural control in animal locomotion . . . . .	1
1.1.1 Central pattern generator . . . . .	1
1.1.2 Reflexes in locomotion control . . . . .	3
1.1.2.1 Cruse’s model of reflexive walking control in a stick insect	3
1.1.2.2 The function of various reflexes in mammals walking control	3
1.1.2.3 Stepping reflex in infants . . . . .	4
1.2 Physical computation in animal locomotion . . . . .	4
1.2.1 Preflex responds rapidly to disturbances . . . . .	5
1.2.2 Muscle properties simplify the locomotion controller . . . . .	5
1.2.3 Smooth movements in animals’ locomotion . . . . .	6
1.3 Biped robots . . . . .	6
1.3.1 Physical computation in passive biped robots . . . . .	6
1.3.2 Various control strategies in powered biped robots . . . . .	7
1.4 Objective . . . . .	9
1.5 Thesis Contributions . . . . .	10
1.6 Thesis Outline . . . . .	11
<b>2 Design of the reflexive controller</b>	<b>12</b>
2.1 Introduction . . . . .	12
2.2 Mechanical design of the robot . . . . .	13
2.3 The neural structure of reflexive controller . . . . .	15
2.3.1 Model neuron circuit of the top level . . . . .	17
2.3.1.1 Inter-neuron model . . . . .	18
2.3.1.2 Stretch receptors . . . . .	18
2.3.1.3 Ground contact sensor neurons . . . . .	19
2.3.2 Neural circuit of the bottom level . . . . .	20
2.4 Robot walking experiments . . . . .	22
2.4.1 Passive movements of the robot . . . . .	22
2.4.2 Walking at different speeds and a perturbed gait . . . . .	24
2.4.3 Walking up a shallow slope . . . . .	26

---

2.5	Stability analysis of the walking gaits . . . . .	27
2.5.1	Dynamic model of the robot . . . . .	27
2.5.2	Stability analysis with Poincaré maps . . . . .	29
2.6	Discussion . . . . .	32
2.6.1	Minimal set of phasic feedbacks . . . . .	32
2.6.2	Physical computation and approximation . . . . .	34
2.6.3	Is CPG a necessitate in biped walking control? . . . . .	34
2.7	Conclusions . . . . .	35
<b>3</b>	<b>Fast dynamic biped walking</b>	<b>36</b>
3.1	Introduction . . . . .	36
3.2	The robot . . . . .	38
3.3	The neural structure of RunBot's controller . . . . .	40
3.3.1	Tuning the neuron parameters . . . . .	43
3.4	Runbot walking experiments . . . . .	44
3.4.1	Changing speed on the fly . . . . .	45
3.4.2	Walking on irregular terrain . . . . .	45
3.5	Fast walking with online policy searching . . . . .	47
3.6	An improved design of RunBot to climb slopes . . . . .	51
3.7	Comparing the speed of robots and humans . . . . .	52
3.8	Froude number . . . . .	53
3.9	Conclusion . . . . .	54
<b>4</b>	<b>Biped walking with a neuromuscular-like controller</b>	<b>55</b>
4.1	Introduction . . . . .	55
4.2	Implementing a muscle model in the robot . . . . .	55
4.3	Robot walking . . . . .	59
4.4	Conclusion . . . . .	61
<b>5</b>	<b>Conclusions</b>	<b>63</b>
5.1	Achievements . . . . .	63
5.2	Potential extensions . . . . .	64
5.2.1	Extending the robot design to 3D . . . . .	64
5.2.2	Extending the learning capability . . . . .	64
5.2.3	Scaling up the robot design . . . . .	65
5.3	Future work . . . . .	65
<b>A</b>	<b>Dynamics equations of the robot</b>	<b>67</b>
<b>B</b>	<b>Simulation analysis</b>	<b>83</b>

## Acknowledgements

Above all, a great thanks to my parents and my wife. Without their support and encouragement, I would not have been able to pursue my interests in scientific research. Also, I must thank my 12-year-old son. Unbelievably, the seemingly naive questions my son often asked me have given me some important inspirations in the design of my robot.

I am very grateful to Prof.Florentin Wörgötter for the guidance, the precious freedom and trust he gave me to develop my ideas in robotics. In addition, I would like to thank Florentin for his great patience in correcting the huge number of language errors in my papers during the past three years.

Special thanks to Dr.Bernd Porr. Bernd taught me many knowledge and skills in Linux and design of amplifiers. Whenever I had problems in Linux or the USB DUX board used on the robots, I could always obtain quick help and advice from Bernd.

Also special thanks to Dr.Peter Hancock. Peter corrected the errors in my papers, and provided many valuable advice in improving my presentations,

Warmest thanks to the former and current members of the Computational Neuroscience Group for making the lab such a lively place: Sinan, Ausra, Matthias, Nicolas, Kevin, Tomas, and Minija.

Finally, many thanks to Bruce Sutherland, Stephen, Hazel O'Donnell, Kay Faichney, Hazel Telfer, Nicola Hunt, and other people in the Department of Psychology. Their warmhearted and timely help have made the three years of my stay in Stirling a very happy time.

## Declaration

Publications based upon the work contained in this thesis:

Geng, T., Porr, B. and Wörgötter, F. Fast biped walking with a sensor-driven neuronal controller and real-time online learning. *International Journal of Robotics Research*, Vol. 25, No. 3, March 2006, p 243-259.

Geng, T., Porr, B. and Wörgötter, F. A Reflexive Neural Network for Dynamic Biped Walking Control. *Neural Computation*, Vol. 18, No. 5, May, 2006, p 1156-1196.

Geng, T., Porr, B. and Wörgötter, F. Fast biped walking with a reflexive controller and real-time policy searching. *Advances in Neural Information Processing Systems (NIPS)* 18, Canada, 2005, p 86-93.

Geng, T., Porr, B. and Wörgötter, F. Fast biped walking with a reflexive neuronal controller and policy gradient reinforcement learning. *Proceedings of 3rd International Symposium on Adaptive Motion in Animals and Machines*, Illmenau, Germany, Sep 2005, p 124-129.

# Chapter 1

## Introduction

### 1.1 Neural control in animal locomotion

#### 1.1.1 Central pattern generator

Neurophysiological studies have revealed that, in most animals' locomotion (walking, swimming, flying, etc.), motor-neurons are driven by central networks referred to as Central Pattern Generators (CPGs). CPGs generate the rhythm and form the pattern of the locomotor bursts of the motor-neurons (Duysens et al., 2002)

CPGs were first discovered by Donald Wilson in 1961 (Purves, 2004). It was demonstrated in his experiments that the oscillatory output of the locust wing beat CPG was maintained in the complete absence of sensory input. Main features of Wilson's model of the CPG were: (1) two neurons reciprocally inhibit each other; (2) they both receive the same constant input; (3) one neuron is more sensitive than the other; (4) both neurons fatigue.

Wilson's CPG (see figure 1.1) produces an oscillatory pattern as follows (Purves, 2004):

1. the constant input comes on and excites the more sensitive neuron first;
2. that neuron fires and inhibits the second neuron;
3. after a short while the first neuron fatigues and releases the second neuron from inhibition;
4. the second neuron is now free to respond to the constant input; the second neuron fires and inhibits the first;



5. then the second neuron fatigues and releases the first from inhibition;
6. again the first neuron fires and inhibits the second, and the cycle repeats.

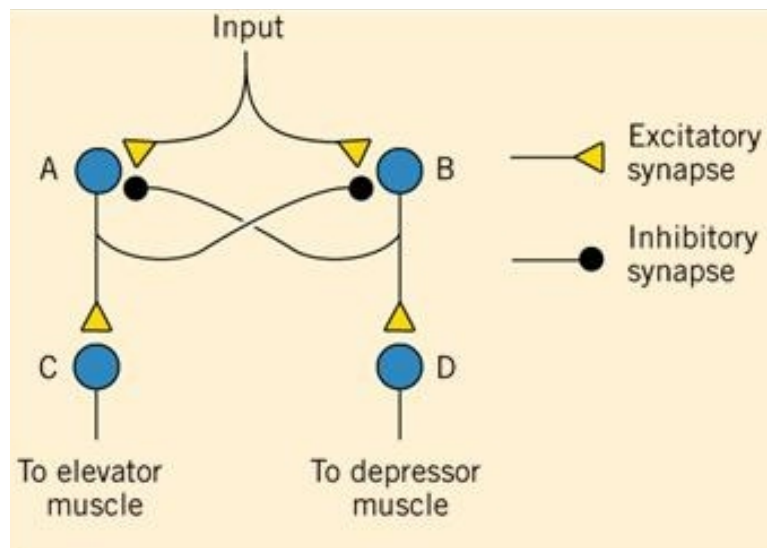


FIGURE 1.1: Wilson's CPG model reproduced from (Purves, 2004).

In the cat, the CPGs are located in the spinal cord. There is at least one such CPG for each limb. Compelling evidence for the existence of such spinal locomotor CPG is that the output pattern can persist even when the cats have a transected spinal cord and are motionless (Duyssens et al., 2002).

In humans, there is indirect evidence for the existence of CPGs. In the case of incomplete SCI (Spinal Cord Injury), the human lumbar cord isolated from brain influence can be trained to respond with rhythmic, locomotor-like electromyography (EMG) activity to peripheral afferents, which are activated by externally-induced stepping movements where the subject was suspended over a moving treadmill (Printer and Dimitrijevic, 1999; Vaughan, 2003).

While not necessary for rhythm generation, sensory feedback is essential for properly coordinating the actual movements of joints and limbs during locomotion. The main sensory feedback to the CPGs is provided by sensory receptors in joints and muscles (Ijspeert, 2001). For example, the hip joint angle of the cat can trigger a new step cycle as the body is propelled over its respective limb on the ground. Activation of muscle stretch receptors can also trigger a new step cycle through contraction of several limb muscles (Cohen and Boothe, 1999). Similarly, the bending of the tail-fin in dogfish or lamprey entrains the swimming cycle to have a length appropriate for the environmental conditions (Grillner and Wallen, 1982). In summary, all CPGs must be entrained by sensory signals to ensure the adaptivity of the CPG-controlled locomotion.

## 1.1.2 Reflexes in locomotion control

Unlike a CPG, a reflex is a local motor response to a local sensation. In the locomotion of human and animals, various reflexes act together in an integrative manner to control the limbs and also contribute to the regulation of the locomotive cycle (Zehr and Stein, 1999).

### 1.1.2.1 Cruse's model of reflexive walking control in a stick insect

There is little evidence for the existence of a strong central pattern generator for walking control in stick insects (Cruse and Warnecke, 1992; Cruse et al., 1998). Cruse developed a reflexive controller model to understand the locomotion control of a slowly-walking stick insect (*Carausius morosus*).

Cruse's model can be divided into two parts according to their functions; one for the control in the leg level, the other for the inter-leg coordination. At the leg level, there are six modules, each for one leg. Each module is composed of several agents or subnetworks (e.g., swing net, selector net). For example, the swing net is for the swing movement control, the stance net for the control in stance phases. The selector net is triggered by the GC (Ground Contact signal) and PEP (Posterior Extreme Position), and determines which movement (swing or stance) is to be performed in the leg. At the inter-leg level, some mechanisms found in the behavioral experiments of insects are used for the coordination between legs. The beginning of a swing phase and the end point of a stance phase are modulated by three influences arising from ipsilateral legs: (1) a rostrally-directed inhibition during the swing movement of the next caudal leg; (2) a rostrally-directed excitation when the next caudal leg begins active retraction; (3) a caudally-directed influence depending on the position of the next rostral leg. Influence (2) and (3) are also active between contralateral legs. At the end of the swing phase, the AEP (Anterior Extreme Position) is modulated by a single, caudally directed influence, which depends on the position of the next rostral leg (Cruse et al., 1998).

### 1.1.2.2 The function of various reflexes in mammals walking control

Although CPGs play a fundamental role in the rhythm generation in most animals, various reflexes work in a harmonic way for adaptivity and robustness of walking gaits.

Akazawa et al. (1982) studied stretch reflexes during locomotion in the mesencephalic cat preparation and observed that stretch and H-reflexes were deeply modulated throughout

the step cycle such that they were large during stance and small during swing phases. It was concluded that stretch reflexes assist in load compensation during gait, particularly during the extension or stance phase (Zehr and Stein, 1999). Dietz et al. (1992) suggested that the regulation of stance during human gait depends strongly on load receptors in the extensor muscles.

Forsberg (1979) evaluated the functional role of cutaneous reflexes by measuring both neural responses and kinematics. Via both electrical and mechanical stimulation of the dorsal surface of the paw in the cat's distal hind-limb during locomotion, a coordinated reflex forming a "stumbling corrective response" was found. This reflex consisted of a sequential neural activation of the hind-limb musculature to allow the perturbed swing limb to continue past the encountered obstacle and maintain stability of ongoing locomotion (Zehr and Stein, 1999).

### 1.1.2.3 Stepping reflex in infants

When held in a standing position on a firm flat surface, a newborn baby will make stepping movements, alternating flexion and extension of each leg, which looks like "walking". This is called "stepping reflex", elicited by the foot's touching of a flat surface. This reflex is controlled by the stepping circuit, which sits in the spinal cord and programs the alternate flexion and extension of each leg.

Normal human walking involves moving the lower limbs alternately, supporting body weight and propelling the body forward, being able to control balance of the whole body. The automatic stepping reflex in infants is essentially the first component, moving the limbs alternately, while supporting some of the body weight and generating some propulsion. However, there is considerable evidence that automatic stepping eventually develops into independent walking (Yang et al., 1998). Presumably, with maturation, there is increased control from the brain that allows the other components of walking control to develop.

## 1.2 Physical computation in animal locomotion

While neural systems modelled as CPGs or reflexive controllers explicitly or implicitly compute walking gaits, the special properties and the mechanics of the musculo-skeletal system also "compute" a large part of the walking movements (Lewis, 2001). This is called physical computation, namely exploiting the system's physics, rather than explicit

models, for global trajectory generation and control. Thus, in all animals, locomotion control is shared between neural computation and physical computation. The musculoskeletal system performs a large part of the gait computation by using its segment mass, length, inertia, elasticity, and damping.

### 1.2.1 Preflex responds rapidly to disturbances

One distinct example of physical computation in animal locomotion is the “preflex”, the nonlinear, passive visco-elastic properties of the musculoskeletal system itself (Brown and Loeb, 1999). Due to the physical nature of the preflex, the system can respond rapidly to disturbances (Cham et al., 2000).

Mechanical reactions to landing, caused by preflex and passive dynamics of the linked segments, may respond to and adjust for new surfaces more rapidly than reflexes. Analytic models reveal that preflexes can stabilize human posture, knee bends, and arm flexion after perturbations. Passive dynamics of the linked segments also help stabilise humans after perturbations. For example, passive dynamics play key roles when walking humans step over an obstacle or recover from tripping. In these cases, active knee flexion results in passive hip flexion due to the mechanical interaction of adjacent segments. Muscle preflexes play a key role in fast adjustments to surface changes especially during rapid locomotion (Moritz and Farley, 2004).

### 1.2.2 Muscle properties simplify the locomotion controller

In human and animal locomotion, the muscle plays more roles than an activator or a motor does in a walking robot. The intrinsic properties of active muscle tissue may be sufficient to produce smooth motions even in the absence of specifically programmed neural inputs (Andrew and Rymer, 1997). The force-length-velocity properties of muscles make them rapidly respond to disturbances (Gerritsen, 1998), facilitating stability during walking. These properties of muscles have tremendously simplified the control demands of the nervous system for walking.

Wagner and Blickhan (2003) investigated in which way minor perturbations to the stable patterns of walking or running can be compensated in a smooth way without disrupting the cycle, and found that pairs of antagonistic muscles are able to stabilise the movement without neuronal feedback. In animals’ locomotion control, in order to guarantee a self-stabilising ability of the muscle-skeletal system, the muscle properties such as force-length

relationship, force-velocity relationship and the muscle geometry have been tuned to the geometric properties of the linkage system (Morasso et al., 2005).

### 1.2.3 Smooth movements in animals' locomotion

The inertia inherent in the mass of the limbs and the intrinsic properties of active muscle tissue may, in themselves, be sufficient to produce smooth motion, even in the absence of specifically programmed neural inputs. These so-called mechanical filtering properties of muscles can filter out the high frequency components of the externally-imposed positional perturbations and the neural signal delivered to the muscle. In a simulation study (Andrew and Rymer, 1997), a model incorporating only simple muscle properties and constant excitation generates substantial trajectory smoothing that is quantitatively comparable to the predictions of the minimum-jerk optimisation model.

## 1.3 Biped robots

During the past 20 years, research on biped robots is increasingly gaining interest. An important motivation for some researchers to build biped robots is to understand the mechanics and control of human walking. Human/animal walking involves huge numbers of sensors, actuators (muscles, tendons), and many redundant degree-of-freedom as well as complex neural systems, which are difficult to measure or analyse. Instead, a biped robot can be regarded as a mechanical counterpart of human that has a tremendously reduced complexity but is still able to represent the essence of the biped locomotion problem. Such a reduced model can help us to obtain some essential insights in biomechanics of human walking. These insights could be particularly useful in the field of rehabilitation (e.g., designing proper prostheses to help disabled people to recover their walking capacity).

### 1.3.1 Physical computation in passive biped robots

Passive dynamic walkers can walk down a shallow slope, demonstrating a smooth and stable gait with out any sensing, control, or actuation. Their gait generation completely depends on physical computation involving the foot-impacting, inertial effects and gravity of the mechanical structure.

McGeer was the first to analyse the dynamics and stability of this kind of passive biped with Poincaré map method. Using a variant of McGeer's original method, some people

studied bifurcations and chaos of the gaits of various 2D and 3D passive biped models (Garcia, 1999). Ruina’s group extended McGeer’s 2D model to 3D passive biped model and have built several biped robots. One of them is shown in figure 1.2. Some researchers have proposed approaches to improve the performance of passive biped robots (e.g., being able to walk on flat ground) (see figure 1.2). Linde made a biped robot walk on level ground by pumping energy into a passive biped at each step (Linde, 1998). Tedrake applied reinforcement learning on a 3D half-passive biped to get dynamic stable gaits (Collins et al., 2005)(see figure 1.2).

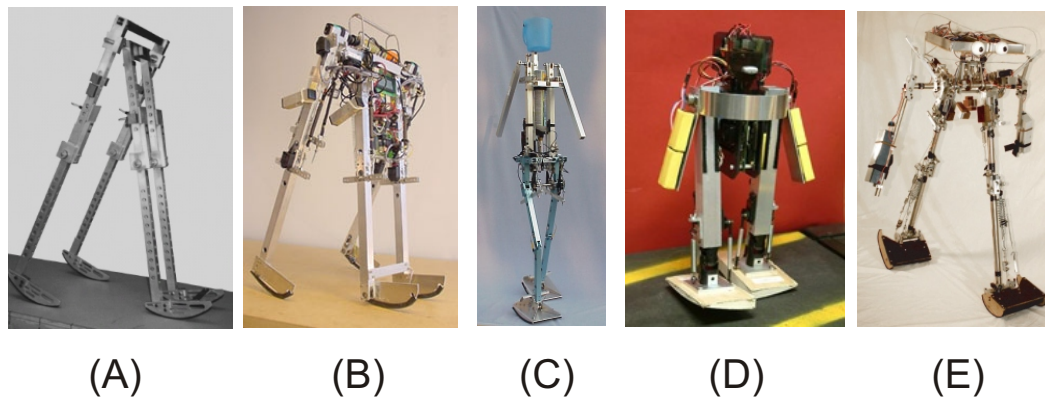


FIGURE 1.2: (A). A copy of McGeer’s planar passive biped robot walking down a slope (Wisse and van Frankenhuyzen, 2003). (B). “Mike”, similar to McGeer’s robot, but equipped with pneumatic actuators at its hip joints. Thus it can walk half-passively on level ground (Wisse and van Frankenhuyzen, 2003). (C). “Denis”, 3D robot with knees and an upper body built by Biorobotics Lab, Technical University of Delft. It uses artificial muscles to provide minimal actuation to walk on a flat floor. Passive springy ankles help to provide stability in the frontal plane and horizontal plane using a lean-yaw coupling mechanism. (D). A non-kneed 3D biped built by Tedrake (Collins et al., 2005). It is equipped with curved feet to facilitate ground clearance of the swing foot and to ensure the stability in frontal plane. (E). A 3D biped robot built by Ruina’s group (Collins et al., 2005). It has actuated hips and passive knee. Arms are used for balance in the frontal plane.

### 1.3.2 Various control strategies in powered biped robots

By contrast to passive bipeds, powered biped robots (especially 3D powered bipeds) usually use various model-based controllers that do not take into account the natural dynamics of the robot (see figure 1.3).

Powered biped robots of the early date (e.g., the SD-2 biped in figure 1.3) used an architecture involving conventional trajectory planning and tracking (Zheng and Shen, 1990). First, the desired joint angles are obtained off-line using the robot’s inverse dynamics

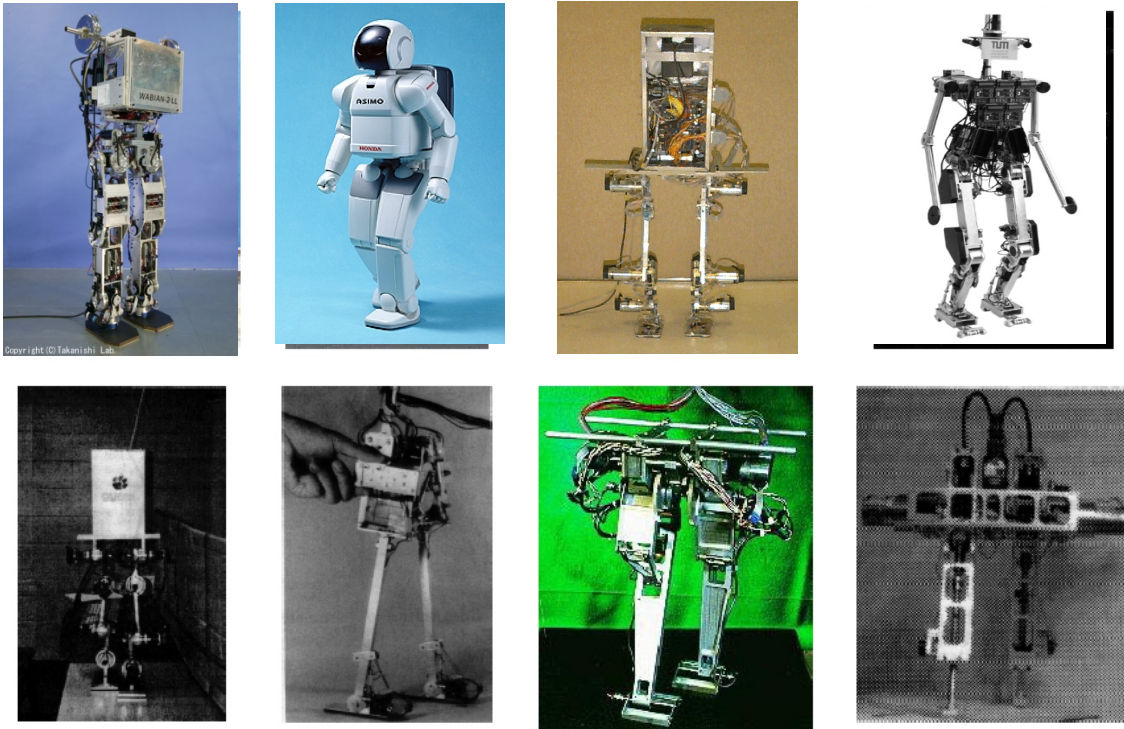


FIGURE 1.3: Powered biped robots. Upper: WABIAN-2/LL from Aseda, ASIMO from Honda, Toddler from UNH, Johnnie from the Technical University of Munich. Lower: SD-2 from Clemson and Ohio State, Biper from University of Tokyo, Meltran II from Tsukuba, and Timmy from Harvard.

model according to the required walking pattern. Then, a linear feedback controller drives the robot joints to follow the desired trajectories. In order to improve the robustness of the robot's gait, some robots used online trajectory planning and a simplified dynamics model which is usually an inverted pendulum model (Park and Kim, 1998). Nevertheless, the most significant progress in the area of biped robots has been made in the last decade. Since 2001, Honda has presented several versions of ASIMO (see figure 1.3), a humanoid robot of human-size. Using sophisticated sensor and control technologies, ASIMO can demonstrate smooth walking gaits on fat ground and stairs (Chestnutt et al., 2005). "Johnnie" is another humanoid robot that has a performance comparable to ASIMO (see figure 1.3). The control system of "Johnnie" is designed such that the orientation of the upper body is controlled throughout all phases of the gait pattern. Furthermore, a sophisticated measurement and control of the foot torques has been implemented (Loffler et al., 2003). In summary, to tackle the huge challenges and difficulties in dynamic biped walking control, sophisticated modern robot control technologies have been applied in biped walking robots, e.g., posture stabilization, inverted pendulum stabilization, contact torque control, foot vertical force control, and torque distribution control.

While most of the controllers of 3D powered biped robots described above ignore or even fight the natural dynamics of the robot, some planar biped robots have been designed with an intention to exploit the natural dynamics at least to some extent. The first to incorporate natural dynamics in powered biped robots was Raibert and Hodgins (1993). Their biped robot used pneumatic cylinders as a spring to exploit the passive dynamics in the vertical direction. “Flamingo”, a planar biped (Pratt, 2000), also exploited the natural dynamics by implementing an intuitive controller and series elastic actuators which were composed of a motor connected in series with a linear elastic element. Passive dynamics generated from spring or elastic legs are not the only form of natural dynamics demonstrated in biped robots. Some biped robots employed mechanical designs that allow the stance leg to pivot freely as an inverted pendulum (Yamaguchi et al., 1999).

## 1.4 Objective

The aim of this study is to understand the coupling of neural computation and physical computation in walking control using a biped robot. Two legs coupled to each other is the minimal mechanical structure from which realistic bipedal walking gaits can emerge. Moreover, the biped model has its special significance. Quadruped gaits usually use legs in pairs: trotting (diagonal legs as pairs), pacing (lateral pairs), and bounding (front pair and rear pair). Thus quadruped control can be reduced to the control of an equivalent virtual biped (Raibert and Hodgins, 1993). When insects are walking fast, they take the form of an alternating tripod gait wherein the first and third ipsi-lateral legs move in phase with the contra-lateral middle leg, all three in anti-phase to the opposite tripod. In this fashion, a tripod acts as a virtual single leg, and the tripod pair is coordinated in the manner of a virtual biped (Full and Tu, 1990; Klavins et al., 2002). Just as stated by Full and Tu 1990, “one human leg works like two dog legs, three cockroach legs, and four crab legs”.

On the other hand, dynamic biped walking is the most difficult to control among various forms of legged locomotion. While multi-legged animals can have at least two legs supporting their bodies at any moment in their locomotion, a biped has only one foot touching the ground during most time of a gait cycle. This poses huge difficulties for the control mechanism as the biped is always tending to fall or trip. One particular objective of this thesis is to show that, even in the most difficult form of walking mechanism, a minimal neural controller coupled with physical computation can generate fast and somewhat adaptive biped walking gaits.

We will show how a dynamically-stable biped walking gait emerges as a result of a com-



combination of neural- and physical computation. Several issues are addressed in this theses which we believe are of relevance for the understanding of biologically-motivated walking control. Specifically we will show that it is possible to realise fast dynamic biped walking with a very sparse set of input signals and with a controller that operates in an approximate and self-regulating way. Both aspects may be of importance in biological systems too, because they allow for a much more limited structure of the neural network and reduce the complexity of the required information processing. Furthermore, in our robot the controller is directly linked to the robot’s motors (its “muscles”) leading to a more realistic, reflexive sensor-motor coupling than implemented in related approaches. These mechanisms allowed us for the first time to arrive at a dynamically stable artificial biped combining physical computation with a pure reflexive controller.

↓-----

The experimental part of this study is complemented by a dynamical model and the assessment of its stability using the Poincaré map approach. Although the simulation analysis has provided insight for our design of the real robot, robot simulations have been recently criticized, raising the issue that complex systems, like a walking robot, cannot be fully simulated because of uncontrollable contingencies in the design and in the world in which it is embedded. This notion, known as the “embodiment problem” has been discussed to a large extent in the robotics literature in the last years (Porr and Wörgötter, 2005). In contrast to simulated robots in a computer, “embodiment refers to the physical existence of a robot. With his robot experiments, Brooks (1986) argued that embodiment is vital to the development of artificial intelligence in a robot. This issue reappears also in our case of biped walking robots. On the one hand, no simulation can exactly simulate the real world. Some transient stages of a walking gait, for example, the impact between the ground and the landing foot, are difficult to model, but are critical to the physical computation in the gait generation. The best way to simulate the function of physical computation is to use a physical model instead of a computational model. On the other hand, in human and animals’ walking, stable gaits emerge from the global entrainment between the neuro-musculo-skeletal system and the environment (Taga, 1995). Therefore, it makes sense to try to understand biped walking control using a real robot in the real world rather than using a simulated model only.

**R7****R7**

↑-----

↑-----

**R1**

## 1.5 Thesis Contributions

The contributions of this thesis are:

1. The biomechatronic design of two biped walking robots, which are used in experiments to show the critical role physical computation can play in stable dynamic walking control.

2. The design of a reflexive neuronal controller, which requires a minimal set of sensor signals at the inter-joint level.
3. The demonstration of record-breaking fast dynamic biped walking gaits in these biped robots.
4. The design of a control algorithm implementing a simple muscle model at the motors of the robot.
5. The implementation of reinforcement learning in fast biped walking control.

## 1.6 Thesis Outline

This thesis proceeds as follows:

Chapter 2 presents the design and experiments of a biped robot and the reflexive controller. Simulation results using Poincaré map are also given in this chapter to analyse the stability of the gaits.

Chapter 3 focuses on one question: how to realise fast biped walking comparable to the relative walking speed of human. It presents a simplified version of the reflexive controller and a redesign of the robot that facilitates fast biped walking.

Chapter 4 presents the implementation of a simple muscle model on the joints of the robot.

Chapter 5 briefly concludes the thesis.

# Chapter 2

## Design of the reflexive controller

### 2.1 Introduction

Reflexive controllers such as Cruse's model involve no central processing unit that demands information on the real-time state of every limb and computes the global trajectory explicitly. Instead, local reflexes of every limb require only very little information concerning the state of the other limbs. Coordinated locomotion emerges from the interaction between local reflexes and the ground. Thus, such a distributed structure can immensely decrease the computational burden of the locomotion controller. With these eminent advantages, Cruse's reflexive controller and its variants have been implemented on some multi-legged robots (Ferrell, 1995).

There existed biped robots exploiting some form of reflexive mechanisms. However, their reflexes usually work as an auxiliary function or as infrastructural units for other non-reflexive high-level or parallel controllers. For example, on a simulated 3D biped robot (Boone and Hodgins, 1997), specifically-designed reflexive mechanisms were used to respond to two types of ground surface contact errors of the robot, slipping and tripping, while the robot's hopping height, forward velocity, and body attitude were separately controlled by three decoupled conventional controllers. On a real biped robot (Funabashi et al., 2001), two pre-wired reflexes are implemented to compensate for two distinct types of disturbances representing an impulsive force and a continuous force, respectively. To date, no real biped robot has existed that depends exclusively on reflexive controllers for walking control. This may be because of the intrinsic instability specific to bipedwalking, which makes the dynamic stability of biped robots much more difficult to control than that of multi-legged robots. After all, a pure local reflexive controller itself involves no mechanisms to ensure the global stability of the biped.

In this chapter, we present the design of a novel reflexive neural controller that has been implemented on a planar biped robot. This chapter is organised as follows. First we describe the mechanical design of our biped robot. Next, we present our neural model of a reflexive network for walking control. Then we demonstrate the result of several biped walking experiments and apply Poincaré map analysis on the robot model. Finally, we compare our reflexive controller with other walking control mechanisms.

## 2.2 Mechanical design of the robot

While the controllers of biped walking robots generally require some kind of continuous position feedback for trajectory computation and stability control, some animals' fast locomotion is largely self-stabilised due to the passive, visco-elastic properties of their musculoskeletal system (Full and Tu, 1990). Not surprisingly, some robots (e.g., passive bipeds) can display a similar self-stabilisation property (Iida and Pfeifer, 2004).

Passive biped robots are usually equipped with circular feet, which can increase the basin of attraction of stable walking gaits, and can make the motion of the stance leg look smoother. Instead, powered biped robots typically use flat feet so that their ankles can effectively apply torque to propel the robot to move forward in the stance phase, and to facilitate its stability control. Although our robot is a powered biped, it has no actuated ankle joints, rendering its stability control even more difficult than that of other powered bipeds. Since we intended to exploit our robot's passive dynamics during some stages of its gait cycle, similarly to the passive biped, its foot bottom also follows a curved form with a radius equal to the leg-length.

As for the mechanical design of our robot, it is 23 cm high, foot to hip. It has four joints: left hip, right hip, left knee, and right knee. Each joint is driven by an RC servo motor. A hard mechanical stop is installed on the knee joints, thus preventing the knee joint from going into hyperextension, similar to the function of knee caps on animals' legs. The built-in PWM (Pulse Width Modulation) control circuits of the RC motors are disconnected while its built-in potentiometer is used to measure joint angles. Its output voltage is sent to a PC through a DA/AD board. Each foot is equipped with a modified Piezo transducer (DN 0714071 from Farnell) to sense ground contact events. We constrain the robot only in the saggittal plane by a boom. All three axes (pitch, roll, and yaw) of the boom can rotate freely (see figure 2.1 C), thus having no influence on the dynamics of the robot in the sagittal plane. Note that the robot is not supported by the boom in the saggittal plane. In fact, it is always prone to trip and fall.

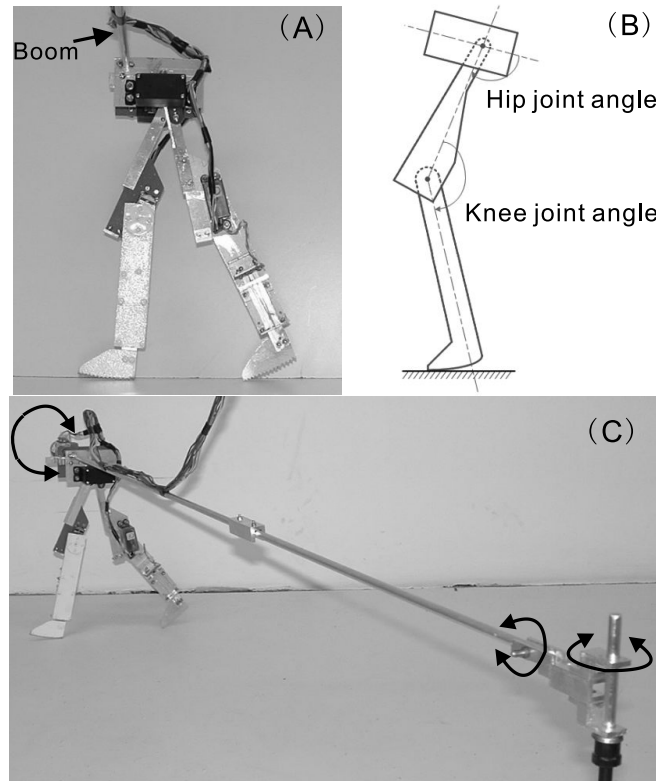


FIGURE 2.1: A) The robot and B) a schematic of the joint angles of one leg. C) The structure of the boom. All its three orthogonal axes (pitch, roll and yaw) rotate freely, thus having no influence on the robot dynamics in its sagittal plane.

The most important consideration in the mechanical design of our robot is the location of its centre of mass. Its links are made of aluminium alloy, which is light and strong enough. The motor of each hip joint is a HS-475HB from Hitec. It weighs 0.04 Kg and can output a torque up to 0.55 Kgcm. Due to the effect of the mechanical stop, the motor of the knee joint bears a smaller torque than the hip joint in stance phases, but must rotate quickly during swing phases for foot clearance. We use a PARK HPXF from Supertec on the knee joint, which is light (0.019 Kg) but fast with 21rad/s. Thus, about seventy percent of the robot's weight is concentrated on its trunk. The parts of the trunk are assembled in such a way that its centre of mass is located as far forward as possible (see figure 2.2). The effect of this design is illustrated in figure 2.2. As shown, one walking step includes two stages, the first from (A) to (B), the second from (B) to (C). During the first stage, the robot has to use its own momentum to rise up on the stance leg. When walking at a low speed, the robot may have not enough momentum to do this. So, the distance the centre of mass has to cover in this stage should be as short as possible, which can be fulfilled by locating the center of mass of the trunk far forward. In the second stage, the robot just falls forward naturally and catches itself on the next stance leg (see figure 2.2). Then the walking cycle is repeated. The figure also shows clearly the movement of the

curved foot of the stance leg. A stance phase begins with the heel touching the ground, and terminates with the toe leaving the ground.

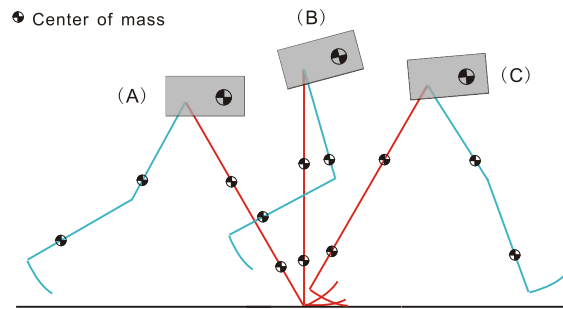


FIGURE 2.2: Illustration of a walking step of the robot.

Aside from the consideration described above, the central issue in the mechanical design is how to determine the exact size and mass distribution of the various parts of the robot. The final design we obtained is a result of several rounds of trial and improvement. At the first round, we built two legs supported by two wheels that can avoid its falling. With this structure, we tested the torques of the motors at different voltages and thus chose a proper leg-length for the robot. At the second round, the supporting wheels were removed. We test the robot's performance by changing the location of the mass centre of each links. However, it is impossible and much time-consuming to try all the various locations by repetitively changing the mechanical parts. So, we went to the third round to find the a good design by simulation analysis. This process of trial and improvement cost about six months.

R2

## 2.3 The neural structure of reflexive controller

As described in chapter 1, the reflexive controller model of Cruse et al. (1998) that has been used to understand the walking control of a stick insect can be roughly divided into two levels: the single leg level and the inter-leg level. Figure 2.3 shows how Cruse's model creates a single leg movement pattern. A protracting leg switches to retraction as soon as it attains the AEP (Anterior Extreme Position). A retracting leg switches to protraction when it attains the PEP (Posterior Extreme Position). On the inter-leg level, six different mechanisms have been described (Cruse et al., 1998), which coordinate leg movements via modifying the AEP and PEP of a receiving leg according to the state of a sending leg.

Although Cruse's model, as a reflexive controller, is for hexapod locomotion, where the

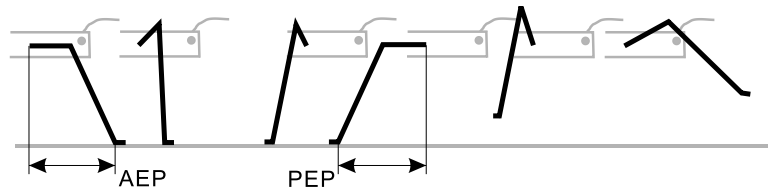


FIGURE 2.3: Single leg movement pattern of Cruse's reflexive controller model (Cruse et al., 1998).

problem of inter-leg coordination is much more complex than in biped walking, we can still compare its mechanism for the generation of single leg movement patterns with that of our reflexive controller. Cruse's model depends on PEP, AEP and GC (Ground Contact) signals to generate the movement pattern of the individual legs. Whereas our reflexive controller presented here uses only GC and AEA (Anterior Extreme Angle of hip joints) to trigger switching between stance and swing phases of each leg. Creation of the single leg movement pattern for our model is illustrated in figure 2.4.

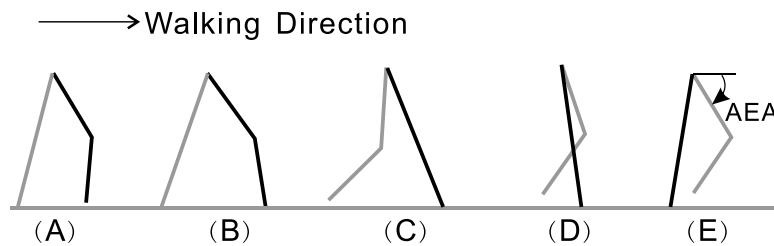


FIGURE 2.4: Illustration of single leg movement pattern generation.

Fig. 2.4 (A)-(E) represents a series of snapshots of the robot configuration while it is walking. At the time of figure 2.4 B, the left foot (black) has just touched the ground. This event triggers four local joint reflexes at the same time: flexor of left hip, extensor of left knee, extensor of right hip, and flexor of right knee. At the time of figure 2.4 E, the right hip joint attains its AEA, which triggers only the extensor reflex of the right knee. When the right foot (grey) contacts the ground, a new walking cycle will begin. Note that on the hip joints and knee joints, extensor means forward movement while flexor means backward movement.

The reflexive walking controller of our robot follows a hierarchical structure (see figure 2.5). The bottom level is the reflex circuit local to the joints, including motor neurons and angle sensor neurons involved in joint reflexes. The top level is a distributed neural network consisting of hip stretch receptors, ground contact sensor neurons, and inter-neurons for reflexes. Neurons are modelled as non-spiking neurons simulated on a Linux PC, and communicated to the robot via the DA/AD board. Though somewhat

AL, (AR)	Stretch receptor for anterior angle of left (right) hip
GL, (GR)	Sensor neuron for ground contact of left (right) foot
EI, (FI)	Extensor (Flexor) reflex inter-neuron
EM, (FM)	Extensor (Flexor) reflex motor-neuron
ES, (FS)	Extensor (Flexor) reflex sensor neuron

TABLE 2.1: *Some abbreviations used in this thesis.*

simplified, they still retain some of the prominent neuronal characteristics.

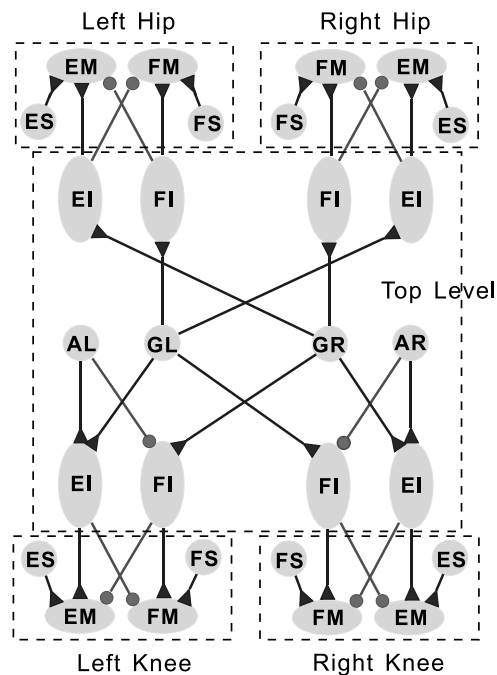


FIGURE 2.5: The neuron model of reflexive controller on our robot. Grey circles are Sensor-neurons or Receptors, Vertical ovals are Inter-neurons, horizontal ovals are Motor-neurons. Synapses: black circle means excitatory, black triangle means inhibitory. For meanings of EI, FI, EM, FM, etc. see Table 2.1.

### 2.3.1 Model neuron circuit of the top level

The joint coordination mechanism in the top level is implemented with the neuron circuit illustrated in figure 2.5. Each of the ground contact sensor neurons has excitatory connections to the inter-neurons of the ipsi-lateral hip flexor and knee extensor as well as to the contra-lateral hip extensor and knee flexor. The stretch receptor of each hip has excitatory connections to its ipsi-lateral inter-neuron of the knee extensor, and inhibitory connection to its ipsi-lateral inter-neuron of the knee flexor. Detailed models



of the inter-neuron, stretch receptor, and ground contact sensor neuron are described in following subsections.

### 2.3.1.1 Inter-neuron model

The inter-neuron model is adapted from one used in the neural controller of a hexapod simulating insect locomotion (Beer and Chiel, 1992). The state of each model neuron is governed by equations 2.1 and 2.2 (Gallagher et al., 1996):

$$\tau_i \frac{dy_i}{dt} = -y_i + \sum \omega_{i,j} u_j \quad (2.1)$$

$$u_j = (1 + e^{\Theta_j - y_j})^{-1} \quad (2.2)$$

where  $y_i$  represents the mean membrane potential of the neuron. Equation 2.2 is a sigmoidal function (see figure 2.6) that can be interpreted as the neuron's short-term average firing frequency,  $\Theta_j$  is a bias constant that controls the firing threshold,  $\tau_i$  is a time constant associated with the passive properties of the cell membrane (Gallagher et al., 1996), and  $\omega_{i,j}$  represents the connection strength from the  $j$ th neuron to the  $i$ th neuron.

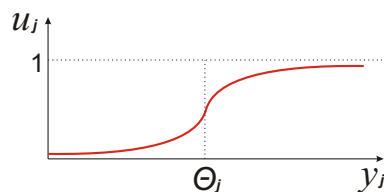


FIGURE 2.6: The output function of the inter-neuron model.

### 2.3.1.2 Stretch receptors

Stretch receptors play a crucial role in animal locomotion control. When the limb of an animal reaches an extreme position, its stretch receptor sends a signal to the controller, resetting the phase of the limbs. There is also evidence that phasic feedback from stretch receptors is essential for maintaining the frequency and duration of normal locomotive movements in some insects (Chiel and Beer, 1997).

While other biologically-inspired locomotive models and robots use two stretch receptors on each leg to signal the attaining of the leg's AEP and PEP respectively, our robot has

only one stretch receptor on each leg to signal the AEA of its hip joint. Furthermore, the function of the stretch receptor on our robot is only to trigger the extensor reflex on the knee joint of the same leg, rather than explicitly (in the case of CPG models) or implicitly (in the case of reflexive controllers) to reset the phase relations between different legs.

As a hip joint approaches the AEA, the output of the stretch receptors for the left (AL) and the right hip (AR) are increased as:

$$\rho_{AL} = \frac{1}{1 + e^{\alpha_{AL}(\Theta_{AL} - \phi)}} \quad (2.3)$$

$$\rho_{AR} = \frac{1}{1 + e^{\alpha_{AR}(\Theta_{AR} - \phi)}} \quad (2.4)$$

where  $\phi$  is the real time angular position of the hip joint,  $\Theta_{AL}$  and  $\Theta_{AR}$  are the hip anterior extreme angles whose values are crucial to the stability and speed of the gait and are tuned according to the result of simulation analysis that will be described later in this chapter, mistuned values of these parameters can cause instable gait,  $\alpha_{AL}$  and  $\alpha_{AR}$  are positive constants. This model is inspired by a sensor neuron model presented in Wadden and Ekeberg (1998) that is thought capable of emulating the response characteristics of populations of sensor neurons in animals.

### 2.3.1.3 Ground contact sensor neurons

Another kind of sensor neuron incorporated in the top level is the ground contact sensor neuron, which is active when the foot is in contact with the ground. Its output, similar to that of the stretch receptors, changes according to:

$$\rho_{GL} = \frac{1}{1 + e^{\alpha_{GL}(\Theta_{GL} - V_L + V_R)}} \quad (2.5)$$

$$\rho_{GR} = \frac{1}{1 + e^{\alpha_{GR}(\Theta_{GR} - V_R + V_L)}} \quad (2.6)$$

where  $V_L$  and  $V_R$  are the output voltage signals from piezo-sensors of the left foot and right foot respectively,  $\Theta_{GL}$  and  $\Theta_{GR}$  work as thresholds, and  $\alpha_{GL}$  and  $\alpha_{GR}$  are positive constants.

While AEP and PEP signals account for switching between stance phase and swing phase in other walking control structures, ground contact signals play a crucial role in phase transition control of our reflexive controller. This emphasised role of the ground contact signal has some biological plausibility. When held in a standing position on a firm flat surface, a newborn baby will make stepping movements, alternating flexion and extension

of each leg, which looks like “walking”. This is called “stepping reflex”, elicited by the foot’s touching of a flat surface. There is considerable evidence that the stepping reflex, though different from actual walking, eventually develops into independent walking (Yang et al., 1998).

Concerning its non-linear dynamics, the biped model is hybrid in nature, containing continuous (in swing phase and stance phase) and discrete (at the ground contact event) elements. Hurmuzlu (1993) applied discrete mapping techniques to study the stability of bipedal locomotion. It was found that the timing of ground contact events has a crucial effect on the stability of biped walking.

Our preference for using a ground contact signal instead of PEP or AEP signals also has other reasons. In PEP/AEP models, the movement pattern of a leg will break down as soon as the AEP or PEP can not be reached, which may happen as a consequence of an unexpected disturbance from the environment or due to intrinsic failure. This can be catastrophic for a biped, though tolerable for a hexapod due to its high degree of redundancy.

### 2.3.2 Neural circuit of the bottom level

In animals, a reflex is a local motor response to a local sensation. It is triggered in response to a suprathreshold stimulus. The quickest reflex in animals is the “monosynaptic reflex”, in which the sensor neuron directly contacts the motor-neuron. The bottom-level reflex system of our robot consists of reflexes local to each joint (see figure 2.5). The neuron module for one reflex is composed of one angle sensor neuron and the motor neuron it contacts (see figure 2.5). Each joint is equipped with two reflexes: extensor reflex and flexor reflex. Both are modelled as a monosynaptic reflex; that is, whenever its threshold is exceeded, the angle sensor neuron directly excites the corresponding motor neuron. This direct connection between angle sensor neuron and motor neuron is inspired by a reflex described in cockroach locomotion (Beer et al., 1997). In addition, the motor neurons of the local reflexes also receive an excitatory synapse and an inhibitory synapse from the inter-neurons of the top level, by which the top level can modulate the bottom level reflexes.

Each joint has two angle sensor neurons, one for the extensor reflex, and the other for the flexor reflex (see figure 2.5). Their models are similar to that of the stretch receptors described above. The extensor angle sensor neuron changes its output according to:

$$\rho_{ES} = \frac{1}{1 + e^{\alpha_{ES}(\phi - \Theta_{ES})}} \quad (2.7)$$

where  $\phi$  is the real time angular position obtained from the potentiometer of the joint (see figure 2.1 B).  $\Theta_{ES}$  is the threshold of the extensor reflex (see figure 2.1 B) and  $\alpha_{ES}$  a positive constant.

Likewise, the output of the flexor sensor neuron is modelled as:

$$\rho_{FS} = \frac{1}{1 + e^{\alpha_{FS}(\Theta_{FS} - \phi)}} \quad (2.8)$$

with  $\Theta_{FS}$  and  $\alpha_{FS}$  similar to above.

It should be particularly noted that the thresholds of the sensor neurons in the reflex modules do not work as desired positions for joint control, because our reflexive controller does not involve any exact position control algorithms that would ensure that the joint positions converge to a desired value. In fact, as will be shown in the next section, the joints often pass these thresholds in swing and stance phase, and begin their passive movement thereafter.

The sensor neurons involved in the local reflex module of each joint can only affect the movements of the joint they belong to, having no direct or indirect connection to other joints. This is different for the phasic feedback signal, AEA, which works at the top level (i.e., the inter-joint level), sensing the position of the hip joints and contacting the motor neurons of the knee joints.

The model of the motor neuron is the same as that of the inter-neurons presented in 2.3.1. Note that, on this robot, the output value of the motor neurons, after multiplication by a gain coefficient, is sent to the servo amplifier to drive the joint motor. In this way, the neural dynamics are directly coupled with the motor dynamics, and furthermore, with the biped walking dynamics. Thus, the robot and its neural controller constitute a closely coupled neuro-mechanical system.

The voltage of joint motor is determined by

$$V = M(G_E U_E + G_F U_F), \quad (2.9)$$

where  $V$  is the input voltage of the motor,  $M$  represents the magnitude of the servo amplifier,  $G_E$  and  $G_F$  are output gains of the motor neurons of the extensor- and flexor reflex respectively,  $U_E$  and  $U_F$  are the outputs of the motor neurons.

	$\Theta_{EI}$	$\Theta_{FI}$	$\Theta_{EM}$	$\Theta_{FM}$	$\alpha_{ES}$	$\alpha_{FS}$
Hip Joints	5	5	5	5	4	1
Knee Joints	5	5	5	5	4	4

TABLE 2.2: Parameters of neurons for hip and knee joints. For meaning of the subscripts, see table 2.1.

$\Theta_{GL}$ (v)	$\Theta_{GR}$ (v)	$\Theta_{AL}$ ( $^{\circ}$ )	$\Theta_{AR}$ ( $^{\circ}$ )	$\alpha_{GL}$	$\alpha_{GR}$	$\alpha_{AL}$	$\alpha_{AR}$
2	2	$= \Theta_{ES}$	$= \Theta_{ES}$	4	4	4	4

TABLE 2.3: Parameters of stretch receptors and ground contact sensor neurons.

## 2.4 Robot walking experiments

The model neuron parameters chosen jointly for all experiments are listed in Tables 2.2 and 2.3. Only the thresholds of the sensor neurons and the output gain of the motor neurons are changed in different experiments. The time constants  $\tau_i$  of all neurons take the same value of 5 ms. The weights of all the inhibitory connections are set to  $-10$ . The weights of all excitatory connections are 10, except those between inter-neurons and motor neurons, which are 0.1.

We encourage readers to watch the video clips of the robot walking experiments at:

Walking fast on a flat floor,

<http://www.cn.stir.ac.uk/~tgeng/robot/walkingfast.mpg>

Walking with a medium speed,

<http://www.cn.stir.ac.uk/~tgeng/robot/walkingmedium.mpg>

Walking slowly

<http://www.cn.stir.ac.uk/~tgeng/robot/walkingslow.mpg>

Climbing a shallow slope,

<http://www.cn.stir.ac.uk/~tgeng/robot/climbingslope.mpg>

These videos can be viewed with Windows Media Player ([www.microsoft.com](http://www.microsoft.com)).

### 2.4.1 Passive movements of the robot

In a walking experiment with specific parameters as given in table 2.6 the passive part of the movement of the robot is shown most clearly. (The sign of  $g_{EM}$  and  $g_{FM}$  depend on the hardware configurations of the motors on the left and right leg).

	$\Theta_{ES}$ ( $^{\circ}$ )	$\Theta_{FS}$ ( $^{\circ}$ )	$g_{EM}$	$g_{FM}$
Hip Joints	115	70	$\pm 2$	$\pm 2$
Knee Joints	180	100	$\pm 1.8$	$\pm 1.8$

TABLE 2.4: Specific parameters for walking experiments.

Figure 2.7 shows the motor voltage and the angular movement of one of its hip joints while the robot is walking. During roughly more than half of every gait cycle, the hip joint is moving passively.

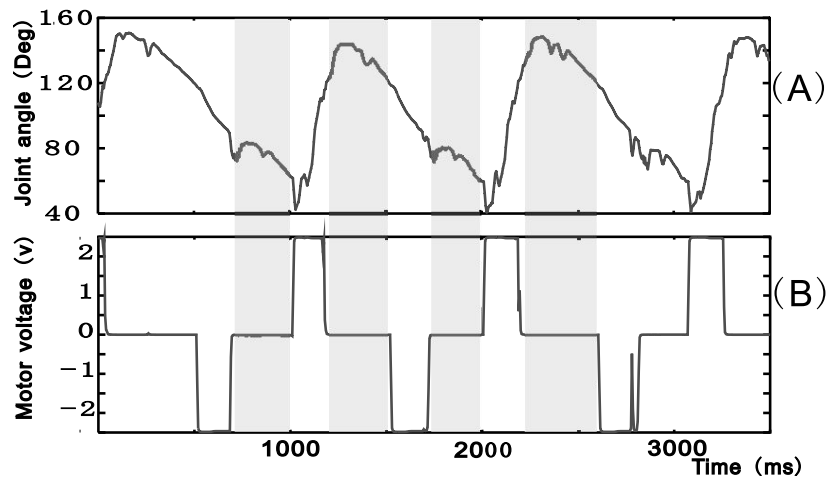


FIGURE 2.7: Real time data of one hip joint. (A) Hip joint angle. (B) Motor voltage measured directly at the motor neurons of the hip joint. During some periods (the grey areas), the motor voltage remains zero, and the hip joint moves passively.

As shown in figure 2.8, during some period of every gait cycle (e.g., grey area in figure 2.8), the motor voltages of the motor neurons on all four joints remain zero, so all joints move passively until the swing leg touches the ground (see figure 2.9). During this time, which is roughly one third of a gait cycle (see figure 2.8 and figure 2.9), the movement of the whole robot is exclusively under the control of “physical computation” following its passive dynamics; no feedback based active control acts on it. This demonstrates very clearly how neurons and mechanical properties work together to generate the whole gait trajectory. This is also analogous to what happens in animal locomotion. Muscle control of animals usually exploits the natural dynamics of their limbs. For instance, during the swing phase of the human walking gait, the leg muscles first experience a power spike to begin leg swing and then remain limp throughout the rest of the swing phase, similar to what is shown in figure 2.9. Note that, in figure 2.9 and the corresponding stick diagrams of walking gait, we omitted the detailed movement of the curved foot, in order to show clearly the leg movements. The point on which the stance leg stands is the orthographic

projection of the mid-point of the foot and not its exact ground-contact point.

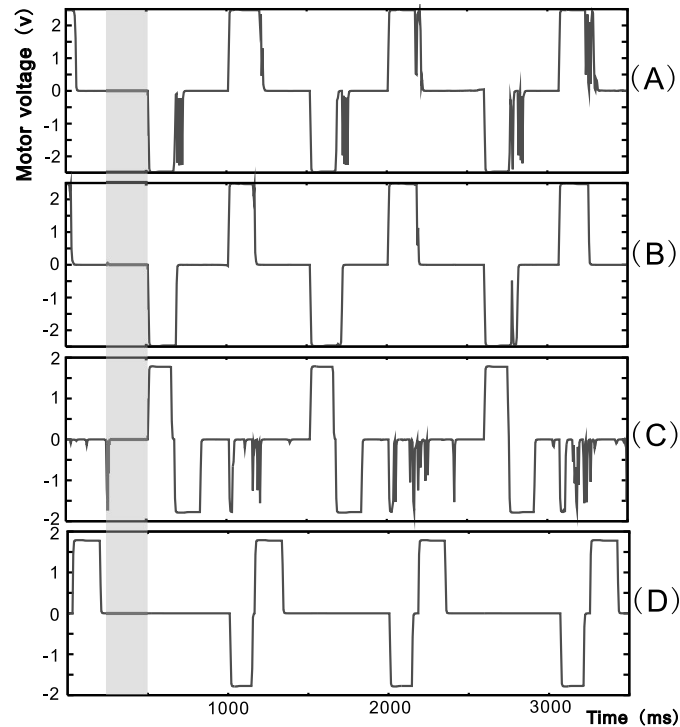


FIGURE 2.8: Motor voltages of the four joints measured directly at the motor neurons, while the robot is walking: (A) left hip; (B) right hip; (C) left knee; (D) right knee. Note that during one period of every gait cycle (grey area), all four motor voltages remain zero, and all four joints (i.e., the whole robot) move passively (see figure 2.9).

## 2.4.2 Walking at different speeds and a perturbed gait

The walking speed of the robot can be changed easily by adjusting only the thresholds of the reflex sensor neurons and the output gain of the motor neurons (see table 2.5). Figure 2.10 A and B show two phase plots of the hip and knee joint positions, which were recorded while the robot was walking with different speeds on a flat floor.

Figure 2.10 C shows a perturbed walking gait. The bulk of the trajectory represents the normal orbit of the walking gait, while the few outlying trajectories are caused by external disturbances induced by small obstacles such as thin books (less than four percent of robot size) obstructing the robot path. After a disturbance, the trajectory returns to its normal orbit quickly, demonstrating that the walking gaits are stable and to some degree robust against external disturbances. Here robustness is defined as rapid convergence to a steady-state behavior despite unexpected perturbations (Lewis, 2001).

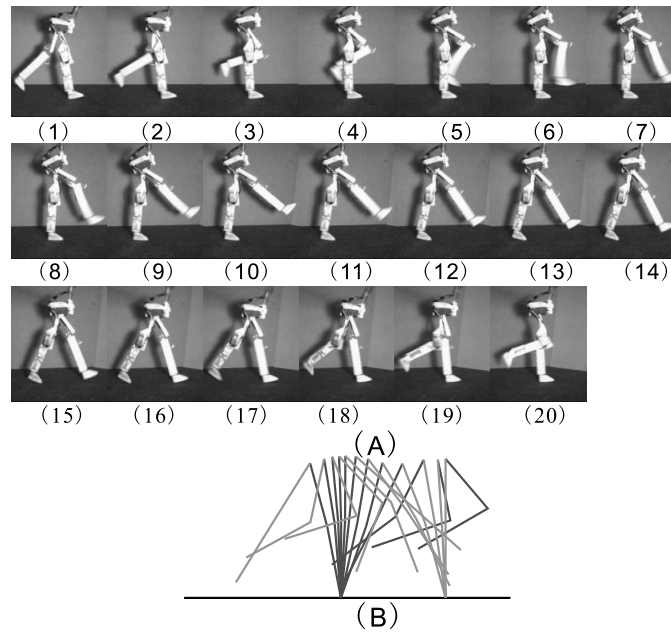


FIGURE 2.9: A). A series of sequential frames of a walking gait cycle. The interval between every two adjacent frames is 33 ms. Note that, during the time between frame (10) and frame (15), which is nearly one third of the time length of a gait cycle (corresponding to the grey area in figure 2.8), the robot is moving passively. At the time of frame (15), the swing leg touches the floor and a new gait cycle begins. (B). Stick diagram of the gait drawn from the frames in (A). The interval between any two consecutive snapshots is 67 ms.

		$\Theta_{ES}$ ( $^{\circ}$ )	$\Theta_{FS}$ ( $^{\circ}$ )	$g_{EM}$	$g_{FM}$
Low speed walking see Fig. 2.10 A	Hip Joints	120	70	$\pm 1.4$	$\pm 1.3$
	Knee Joints	180	100	$\pm 1.5$	$\pm 1.5$
High speed walking see Fig. 2.10 B	Hip Joints	110	85	$\pm 2.5$	$\pm 2.5$
	Knee Joints	180	100	$\pm 1.8$	$\pm 1.8$
Perturbed walking gait see Fig. 2.10 C	Hip Joints	115	90	$\pm 2.5$	$\pm 2.5$
	Knee Joints	180	100	$\pm 1.5$	$\pm 1.5$

TABLE 2.5: The different values of neuron parameters chosen to generate different speeds (see figure 2.10).

With neuron parameters changed in the cases of fast walking and slow walking, walking dynamics are implicitly drawn into a different gait cycle. Figure 2.10 D shows an experiment in which the neuron parameters are changed abruptly online while the robot is walking at a slow speed (33 cm/s, the big orbit). After a short transient stage (the outlying trajectories), the gait cycle of the robot is automatically transferred into another stable, high-speed orbit (the small one, 57 cm/s). In other words, when the neuron parameters are changed, physical computation closely coupled with neural computation can autonomously shift the system into another global trajectory that is also dynamically



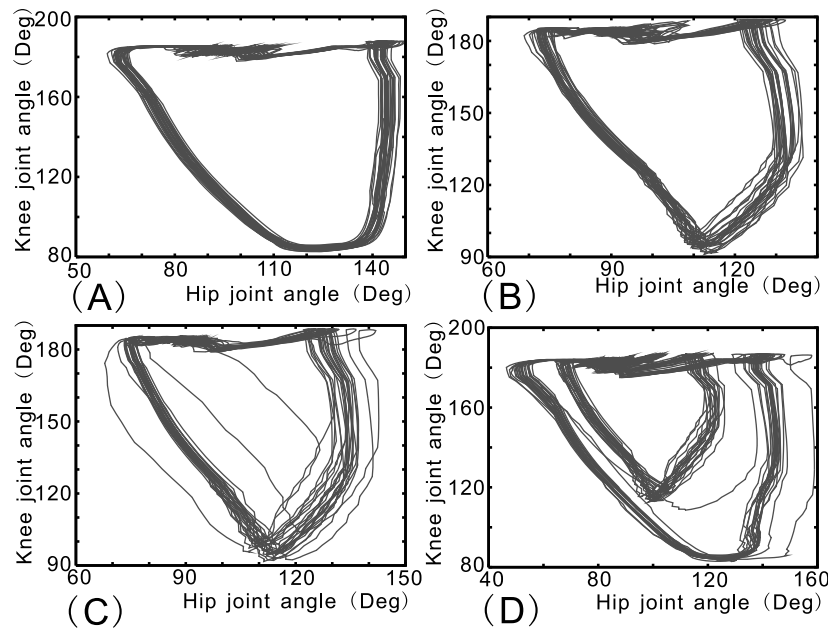


FIGURE 2.10: Phase diagrams of hip joint position and knee joint position of one leg. Robot speed: (A) 28 cm/s; (B) 63 cm/s. (C) A perturbed walking gait. For values of the neuron parameters chosen in these experiments, see table 2.6. Note that the hip joint angle in these figures is an absolute value, not the angle relative to the robot body as shown in Fig. 2.1 B. (D) The walking speed is changed online.

stable. This experiment shows that our biped robot, as a neuro-mechanical system, is stable in a relatively large domain of its neuron parameters.

With other real-time biped walking controllers based on biologically inspired mechanisms (e.g., CPG) or conventional trajectory preplanning and tracking control, it is still a puzzling problem how to change walking speed on the fly without undermining dynamical stability at the same time. However, this experiment shows that the walking speed of our robot can be drastically changed (nearly doubled) on the fly while the stability is still retained due to physical computation.

### 2.4.3 Walking up a shallow slope

Figure 2.11 is a stick diagram of the robot when it is walking up a shallow slope of about 4 degrees. Steeper slopes could not be mastered. In figure 2.11, we can see that, when the robot is climbing the slope, its step length is becoming smaller, and the movement of its stance leg is becoming slower (its stick snapshots are becoming denser). Note that these adjustments of its gait take place autonomously due to the robot's physical properties (physical computation), not relying on any pre-planned trajectory or precise

control mechanism. This experiment demonstrates that such a closely coupled neuro-mechanical system can to some degree autonomously adapt to an unstructured terrain.

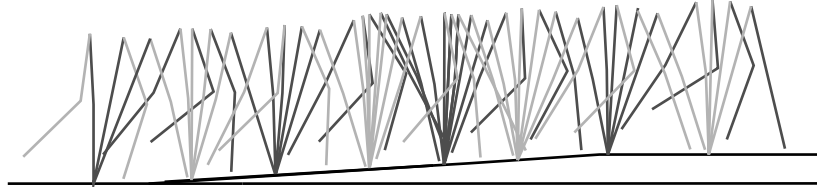


FIGURE 2.11: The robot is climbing a shallow slope. The interval between any two consecutive snapshots is 67 ms.

## 2.5 Stability analysis of the walking gaits

### 2.5.1 Dynamic model of the robot

The dynamics of our robot are modelled as shown in figure 2.12. With the Lagrange method, the equations that govern the motion of the robot are:

$$D(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau \quad (2.10)$$

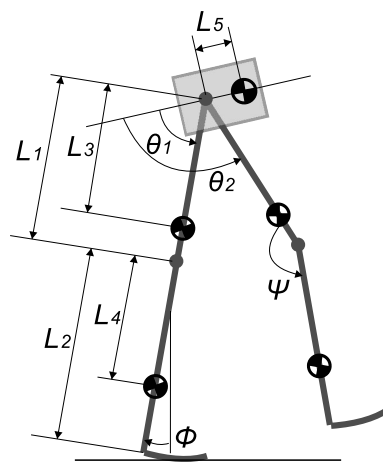


FIGURE 2.12: Model of the dynamics of our robot. Sizes and masses are the same as those of the real robot.

where  $q = [\phi, \theta_1, \theta_2, \psi]^T$  is a vector describing the configuration of the robot (for definition of  $\phi, \theta_1, \theta_2, \psi$ , see figure 2.12),  $D(q)$  is the  $4 \times 4$  inertia matrix,  $C(q, \dot{q})$  is the  $4 \times 1$  vector of centripetal and coriolis forces,  $G(q)$  is the  $4 \times 1$  vector representing gravity forces,  $\tau = [0, \tau_1, \tau_2, \tau_3]^T$ ,  $\tau_1, \tau_2, \tau_3$  are the torques applied on the stance hip (the hip joint of the stance leg in figure 2.12), the swing hip, and the swing knee joints, respectively. Details of this equation can be found in Appendix A.

The dynamics of the dc motor (including gears) of each joint can be described with the following equations (here, the hip of the stance leg is taken as an example. The models of other joints are likewise):

$$L_a \frac{di_a}{dt} + R_a i_a + nk_3 \dot{\theta}_1 = V_1 \quad (2.11)$$

$$\tau_1 + I_1 \ddot{\theta}_1 + k_f \dot{\theta}_1 = nk_2 i_a \quad (2.12)$$

where,  $V_1$  is the applied armature voltage of the stance hip motor, which is obtained from the output of the motor neurons according to equation 2.9,  $i_a$  is the armature current,  $L_a$  the armature inductance,  $R_a$  the armature resistance,  $k_3$  is the emf constant,  $k_2$  is the motor torque constant,  $I_1$  is the combined moment of inertial of the stance-hip motor and gear train referred to the gear output shaft,  $k_f$  is the vicious-friction coefficient of the combination of the motor and the gear and  $n$  is the gear ratio.

Considering that the electrical time-constant of the motor is much smaller than the mechanical time-constant of the robot, we neglect the dynamics of the electrical circuits of the motor, which leads to  $\frac{di_a}{dt} = 0$ . Thus equation 2.11 is reduced to,

$$i_a = \frac{1}{R_a} (V_1 - nk_3 \dot{\theta}_1) \quad (2.13)$$

Combining equations 2.10, 2.12 and 2.13, we can get the dynamics model of the robot with the applied motor voltages as its control input.

The heel strike at the end of swing phase and the knee strike at the end of knee extensor reflex are assumed to be inelastic impacts, which is in accordance with observations on our real robot and existing passive biped robots. This assumption implies the conservation of angular momentum of the robot just before and after the strikes, with which the value of  $\dot{q}$  just after the strikes can be computed using its value just before the strikes. Because the transient double support phase is very short in our robot walking (usually less than 40 ms), it is neglected in our simulation as often done in the analysis of other passive bipeds (Garcia, 1999).

## 2.5.2 Stability analysis with Poincaré maps

↓-----

In dynamical systems, a Poincaré map, named after Henri Poincaré, is the intersection of a periodic orbit in the state space of a dynamical system with a certain lower dimensional subspace (called the Poincaré section) transversal to the flow of the system. The method of Poincaré maps is a particularly useful tool for stability analysis of passive bipeds (Garcia, 1999). That is because:

**R6**

1. The movement of the biped robot is cyclic and nonlinear.
2. Due to the impact of the landing foot on the ground, the biped robot is a hybrid dynamic system, which makes it difficult to apply other analytical methods in its stability analysis.

Because our reflexive controller exploits natural dynamics for the robot's motion generation, and no trajectory planning or tracking control are used, thus the Poincaré map approach can also be applied to the dynamics model of our robot together with the reflexive network as its controller.

↑-----

We choose the Poincaré section to be right after the heel strike of the swing leg. Each cyclic walking gait is a limit cycle in the state space, corresponding to a fixed point on the Poincaré section. Fixed points can be found by solving the roots of the mapping equation:

$$P(x^n) - x^n = 0 \quad (2.14)$$

where  $x^n = [q, \dot{q}]^T = [\phi, \theta_1, \theta_2, \psi, \dot{\phi}, \dot{\theta}_1, \dot{\theta}_2, \dot{\psi}]^T$  is a state vector on the Poincaré section at the beginning of the  $n_{th}$  gait cycle.  $P(x^n)$  is a map function mapping  $x^n$  to  $x^{n+1}$ , which is built by combining the reflexive controller and the robot dynamics model described above.

Near a fixed point,  $x^*$ , the map function  $P(x^*)$  can be linearised as:

$$P(x^* + \hat{x}) \approx P(x^*) + J\hat{x} \quad (2.15)$$

where  $J$  is the  $8 \times 8$  Jacobian matrix of partial derivatives of  $P$

$$J = \frac{\partial P}{\partial x} \quad (2.16)$$

TABLE 2.6: Fixed parameters of the knee joints.

	$\Theta_{ES,k}$ ( $^\circ$ )	$\Theta_{FS,k}$ ( $^\circ$ )	$G_{M,k}$
Knee Joints	180	110	$0.9G_{M,h}$

With any fixed point,  $J$  can be obtained by numerically evaluating  $P$  eight times in a small neighborhood of the fixed point. According to equation 2.15, small perturbations  $\hat{x}^i$  to the limit cycle  $x^*$  at the start of  $i$ th step will grow or decay from the  $i$ th step to the  $i + 1$ th step approximately according to  $\hat{x}^{i+1} \approx J\hat{x}^i$ . So, if all eigenvalues of  $J$  lie within the unit circle, any small perturbation will decay to 0 and the perturbed walking gait will return to its limit cycle, which means the limit cycle is asymptotically stable (Garcia, 1999).

The movements of the knee joints are needed mainly for timely ground clearance without much influence on the stability of the walking gait. Therefore, in the simulation analysis and real experiment below we set the knees' neuron parameters to fixed values (see Table 2.6) that can ensure fast movements of the knee joints, preventing any possible scuff of the swing leg.

For simplicity, we also fix the threshold of the flexor sensor neurons of the hips ( $\Theta_{FS,h}$ ) to  $85^\circ$  in simulation and real experiments below. This will not damage the generality of the results, because similar results can be obtained provided that  $\Theta_{FS,h}$  is in the interval  $70 - 90^\circ$ . For values outside this range the robot will either fall or produce gaits which are very unnatural. Thus, now we only need to tune two parameters of the hip joints: the threshold of the extensor sensor neurons ( $\Theta_{ES,h}$ ) and the gain of the motor neurons of hip joints ( $G_{M,h}$ ), which work together to determine the gait properties.  $G_{M,h}$  determines the amplitude of the applied voltage of the motors on the hip joint. Since these two parameters have such clear physical interpretations, their tuning is straightforward.

With each set of the controller parameter  $\Theta_{ES,h}$  and  $G_{M,h}$ , we use a Newton-Raphson method solving equation éPoincaré to find the fixed point (Garcia, 1999). Then we compute the Jacobian matrix  $J$  of the fixed point using the approach described above, and evaluate the stability of the fixed point according to its eigenvalues. The result of this Poincaré map analysis is shown in figure 2.13. We have found that asymptotically stable fixed points exist in a considerably large range of the controller parameters  $\Theta_{ES,h}$  and  $G_{M,h}$  (see figure 2.13). For comparison, figure 2.13 also shows the stable range of these two parameters obtained in real robot experiments. In the real robot, because no definite stability criterion, like using eigenvalues, is applicable, we regard a walking gait as stable if the robot does not fall.

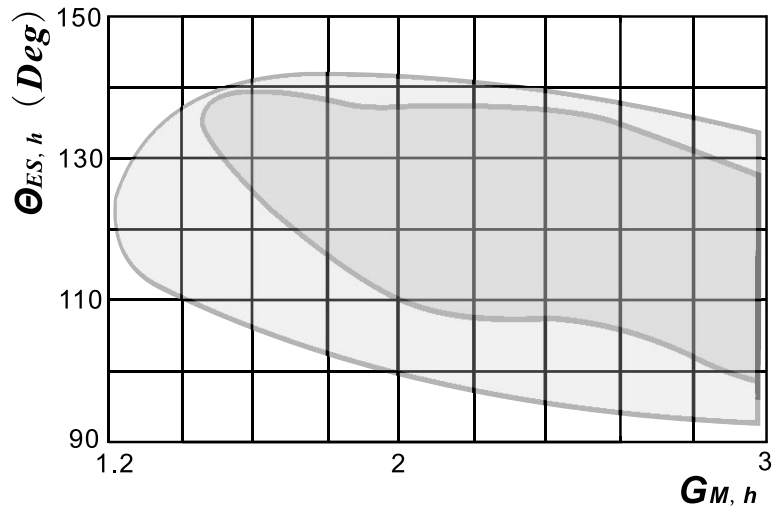


FIGURE 2.13: Stable domain of the controller parameter,  $\Theta_{ES,h}$  and  $G_{M,h}$ . The big area enclosed by the outer curve represents the range obtained with simulations in which fixed points are stable. The shaded area is the range of the two parameters, in which stable gaits will appear in experiments performed with the real robot. The maximum permitted value of  $G_{M,h}$  is 2.95 (higher values will destroy the motor of the hip joint). The two closed curves are a manual, continuous interpolation of the discrete boundaries obtained in simulations and real experiments, respectively.

The best way to visualise the properties of a limit cycle is using the phase plane, which can be easily obtained in the simulations, but is not available in our real robot due to the lack of absolute position and speed sensors. Figure 2.14 shows two phase plane plots of the absolute angular position of one hip joint,  $\phi$  (see figure 2.12) and its derivative,  $\dot{\phi}$ . After being perturbed, the walking gait returns to its limit cycle quickly in only a few steps, which is in accordance with the experiment results of the real robot presented in the last section.

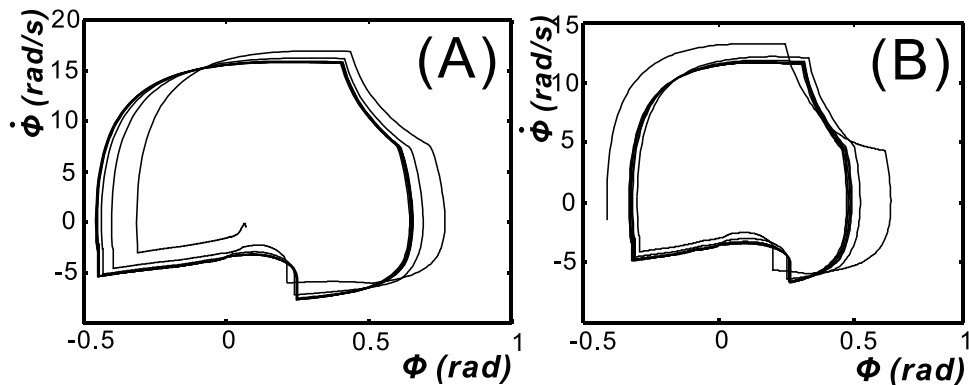


FIGURE 2.14: Two limit cycles in the phase plan of  $\phi$  and  $\dot{\phi}$ . (A) is corresponding to a fixed point found with this set of controller parameters,  $\Theta_{ES,h} = 125^\circ$ ,  $G_{M,h} = 2.8$ , (B) corresponding to  $\Theta_{ES,h} = 110^\circ$ ,  $G_{M,h} = 2.5$ .

Because some details of the robot dynamics such as uncertainties of the ground contact, nonlinear frictions in the joints and the inevitable noise and lag of the sensors are difficult, if not impossible, to model precisely, the results of simulation and real experiments are not exactly equivalent (see figure 2.13). However, stability analysis and experiments with our real robot have in general shown that our biped robot under control of the reflexive network will demonstrate stable walking gaits in a wide range of the critical controller parameters and that it will return to its normal orbit quickly after a disturbance.

## 2.6 Discussion

### 2.6.1 Minimal set of phasic feedbacks

The aim of locomotion control structures (modelled either with CPG or with reflexive controllers) is to control the phase relations between limbs or joints, attaining a stable phase locking that leads to a stable gait. Therefore, the locomotion controller needs phasic feedback from the legs or joints. In the case of reflexive controllers like Cruse's model (Cruse et al., 1998), the phasic feedback signals sent to the controller are AEP and PEP signals, which can provide sufficient information on phase relations at least between adjacent legs. It is according to this information that the reflexive controller adjusts the PEP value of the leg, thus effectively changing the period of the leg, synchronizing it in, or out of phase with its adjacent legs (Klavins et al., 2002).

On the other hand, in the case of a CPG model, which can generate rhythmic movement patterns even without sensory feedback, it must nonetheless be entrained to phasic feedback from the legs in order to achieve realistic locomotion gaits. In some animals, evidence exists that every limb involved in cyclic locomotion has its own CPG (Delcomyn, 1980), and phasic feedback from muscles is indispensable to keep its CPGs in phase with the real time movement of the limbs. Not surprisingly, CPG mechanisms used on various locomotive robots also require phasic feedback. Lewis et al. (2003) implemented a CPG oscillator circuit to control a simple biped. AEP and PEP signals from its hip joints define the feedback to the CPG, resetting its oscillator circuit. Removal of the AEP or PEP signals caused rapid deterioration of this biped's gait. On another quadruped robot (Fukuoka et al., 2003), instead of discrete AEP and PEP signals, continuous position signals of the hip joints provide feedback to the neural oscillators of the CPG. The neural oscillator parameters were tuned in such a way that the minimum and maximum of the hip positions would reset the flexor and extensor oscillator respectively. Apparently, this scheme functions identically with AEP/PEP feedback.

In summary, because AEP and PEP provide sufficient information about phase relations between legs, walking control structures usually depend on them (or their equivalents) as phasic feedback from the legs. However, the top level of the reflexive controller on our robot requires only AEA signals as phasic feedback. Furthermore, this AEA signal is only for triggering the flexor reflex on the knee joint of the robot, rather than triggering stance phases as in other robots. In this sense, the role (and number) of the phasic feedback signals is much reduced in our reflexive controller.

In spite of the fact that the AEA signal is by itself not sufficient to control the phase relations between legs, stable walking gaits have appeared in our robot walking experiments (see section 4). This is because reflexive controller and physical computation cooperate to accomplish the task of phasic walking gait control. This shows that physical computation can help to simplify the controller structure.

As described above, CPGs have been successfully applied on some quadruped, hexapod and other multi-legged robots. However, in biped walking control based on CPG models, most of the current studies are performed with computer simulations. To our knowledge, no one has successfully realised real-time dynamic biped walking using a CPG model as a single controller, because the CPG model itself can not ensure stability of the biped gait. A considerably well-known biped robot controlled by a CPG chip has been developed by Lewis et al. (2003). Its walking/running gaits look very nice, though on a treadmill instead of on a floor. But this biped robot has a fatally weak point in that its hips are fixed on a boom (not rotating freely around the boom axes as in our robot). So it is actually supported by the boom. The boom is greatly facilitating its control, avoiding the most difficult problem of dynamic stability control that is specific to biped robots. Thus, this robot is indeed not a dynamic biped in its real sense. Instead, it is rather more equivalent to one pair of legs of a multi-legged robot.

Using computer simulations, Taga (1995) found that stable biped gaits can be generated by combining CPGs and human biomechanics. In animals, a CPG is a neural structure which is much more complex than the local reflex in anatomy and function. There is evidence that, in mammal and human locomotion, CPGs work on top of reflexes and take their effects via modulating them. In evolution, simple monosynaptic reflexes must have appeared much earlier than the much more complex CPG structures. Not only with simulation analysis, but also with our real system experiments, the current study has shown that local neuronal reflexes connected by a simple network are sufficient as a controller for dynamic biped walking, the most difficult form of legged locomotion in view of dynamic stability.



## 2.6.2 Physical computation and approximation

In contrast to exact representations and world models, physical computation often implies approximation. Approximation in control mechanism gives more room and possibility for physical computation. While conventional robots rely on precise trajectory planning and tracking control, biologically-inspired robots rarely use preplanned or explicitly computed trajectories. Instead, they compute their movements approximately by exploiting physical properties of their self and the world, thus avoiding the accurate calibration and modelling required by conventional robotics. But, in order to achieve real time walking gaits in a real world, even these biological inspired robots often have to depend on some kind of position or velocity control on their joints. For example, on a hexapod, simulating the distributed locomotion control of insects (Beer et al., 1997), outputs of motor neurons were integrated to produce a trajectory of joint positions that was tracked using proportional feedback position control. On a quadruped, built by Kimura's group, that implemented CPGs (neural oscillators) and local reflexes, all joints are controlled to move to their desired angles (Fukuoka et al., 2003). Even on a half passive biped, controlled by a CPG chip, position control worked on its hip joints, although passive dynamics of its knee joints was exploited for physical computation (Lewis, 2001).

The principle of approximation embodied in the reflexive controller of our robot, however, goes even one step further, in the sense that there is no position or velocity control implemented on our robot. The neural structure of our reflexive controller does not depend on, or ensure the tracking of, any desired position. Indeed, it is this approximate nature of our reflexive controller that allows the physical properties of the robot itself, especially the passive dynamics of the robot (see figure 2.9), to contribute implicitly to generation of overall gait trajectories, and ensures its stability and robustness to some extent. Just as argued by Raibert and Hodgins (1993), "Many researchers in neural motor control think of the nervous system as a source of commands that are issued to the body as direct orders. We believe that the mechanical system has a mind of its own, governed by the physical structure and the laws of physics. Rather than issuing commands, the nervous system can only make suggestions which are reconciled with the physics of the system and the task."

## 2.6.3 Is CPG a necessitate in biped walking control?

This work has shown that a series of mono-synaptic reflexes can generate stable and robust biped walking gait, not needing more complex neural structures like CPG.

## 2.7 Conclusions

In this chapter, we presented our design and some walking experiments performed by a novel neuro-mechanical structure for reflexive walking control. We demonstrated with a closely coupled neuro-mechanical system how physical computation can be exploited to generate a dynamically-stable biped walking gait. In the experiments of walking at different speeds and climbing a shallow slope, it was also shown that the coupled dynamics of this neuro-mechanical system are sufficient to induce an autonomous, albeit limited, adaptation of the gait.

# Chapter 3

## Fast dynamic biped walking

### 3.1 Introduction

Building and controlling fast biped robots demands a deeper understanding of biped walking than for slow robots (Pratt, 2000). While slow robots may walk statically, fast biped walking has to be dynamically balanced and more robust as less time is available to recover from disturbances (Pratt, 2000). Although many biped robots have been developed using various technologies in the past 20 years (see figure 1.3), their walking speeds are still not comparable to that of their counterpart in nature, humans. While the fastest biped robot today can cover 1.5 leg-lengths per second, the Olympic record of human walking is equivalent to more than 4 leg-lengths/s. Now the question is, why biped robots are so slow.

Most of the successful biped robots have commonly used the ZMP (Zero Moment Point) as the criterion for stability control and motion generation (Hirai, 1997; Yamaguchi et al., 1999; Inoue and Tachi, 2000; Nishiwaki et al., 2000; Kuroki et al., 2001; Miyakoshi and Cheng, 2002). The ZMP is the point on the ground where the total moment generated by gravity and inertia equals zero (Vukobratovic et al., 1990). This measure has two deficiencies in the case of high-speed walking. First, the ZMP must always reside in the convex hull of the stance foot, and the stability margin is measured by the minimal distance between the ZMP and the edge of the foot. To ensure an appropriate stability margin, the foot has to be flat and large, which will reduce the robot's performance and pose great difficulty during fast walking. This difficulty can be seen clearly when humans try to walk with skis or swimming fins. Second, the ZMP criterion does not permit rotation of the stance foot at the heel or the toe, which, however, can amount to up to eighty percent of a normal human walking gait (Hardt and von Stryk, 2002), and is

important and inevitable in fast biped walking (see figure 3.1). Just try whether you can walk fast without rolling your feet on the ground.

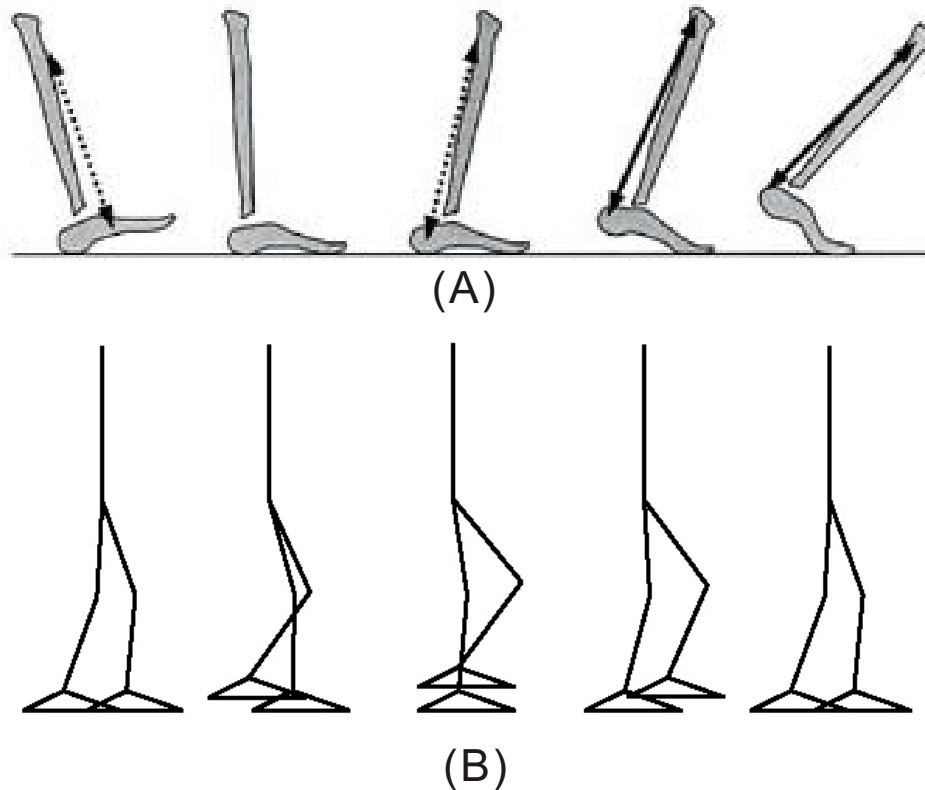


FIGURE 3.1: (A). The rolling movement of the stance foot around the heel and the toe in human walking. (B). ZMP-based controller does not permit foot-rolling in biped robots (Azevedo et al., 2004).

Of course, not all powered biped robots employed ZMP in their controller. Some biped robots employ various types of model-based control of an inverted pendulum model of the upper body (Kajita and Kobayashi, 1987; Miyazaki and Arimoto, 1987; Sano and Furusho, 1990). Chevallereau et al. (2003) designed a trajectory tracking controller based on the zero dynamics of a planar biped robot with unactuated ankles, by which asymptotically stable walking gaits were realised. However, these robots are also very slow because their trajectory planning and tracking control usually more or less fight rather than exploit the natural dynamics of the robots.

On the other hand, sometimes dynamic biped walking can be achieved simply without explicitly considering any stability criterion. Specific trajectories and precise trajectory tracking are not indispensable for biped walking. For example, passive biped robots can walk stably down a shallow slope with no sensing or control. Usually equipped with point feet or curved feet, only one point of the foot touches the ground at any time, which would

be unstable when applying the ZMP criterion. However, compared with powered bipeds, passive biped robots have obvious drawbacks, e.g., their need for walking down a slope and their inability to control the speed (Pratt, 2000). Some researchers have proposed approaches to equip a passive biped with actuators to improve its performance. Nevertheless, no one has yet built a passive biped robot that can walk at a speed comparable to humans', though humans also exploit passive movements in some stages of their walking gaits.

Unlike the robots using controllers based on the ZMP criterion, humans' stable and robust gaits can emerge from the global entrainment between the neuro-musculo-skeletal system and the environment (Taga, 1995). In this study, we will realise fast planar biped walking with a simple neuro-mechanical system, which involves a simplified version of the reflexive neuronal controller and a redesign of the robot. The basic components required by ZMP control (e.g., dynamics models, position and trajectory control) will not be employed in our neural controller. Instead, It will be shown in our experiments that fast and stable biped walking can emerge from the interaction between such a neuro-mechanical system and the ground without needing intensive feedback control as in the case ZMP-based control.

This chapter is organised as follows. First we describe the mechanical design of our new biped robot named "RunBot". Next, we present the structure of the newly designed neuronal controller. Then we demonstrate the result of several biped walking experiments.

## 3.2 The robot

RunBot is a mechanical redesign of our previous robot (Geng et al., 2006) with a simplified controller and specific properties to allow for fast walking. RunBot (see figure 3.2) is 23 cm high, foot to hip joint axis. Its thigh is 11 cm long, and shank (including the foot) 12 cm. It also uses the same model of motors as the previous robot. The mass of its trunk is 130 g. The mass of the two legs is 60 g. Similar to the previous robot, RunBot is constrained in the sagittal plane by a boom of one meter length. The robot is attached to the boom via a freely-rotating joint while the boom is attached to the central column with a universal joint (see figure 3.2). Thus, its movements are constrained on the surface of a sphere. However, considering that the length of the boom is more than 4 times RunBot's height, we think that the influence of the boom on its dynamics in the sagittal plane is very small. The boom is still allowing it to freely trip or fall.

Similar to the previous design, RunBot also has unactuated feet, which can be very light,

↓-----

**R14**

↑-----

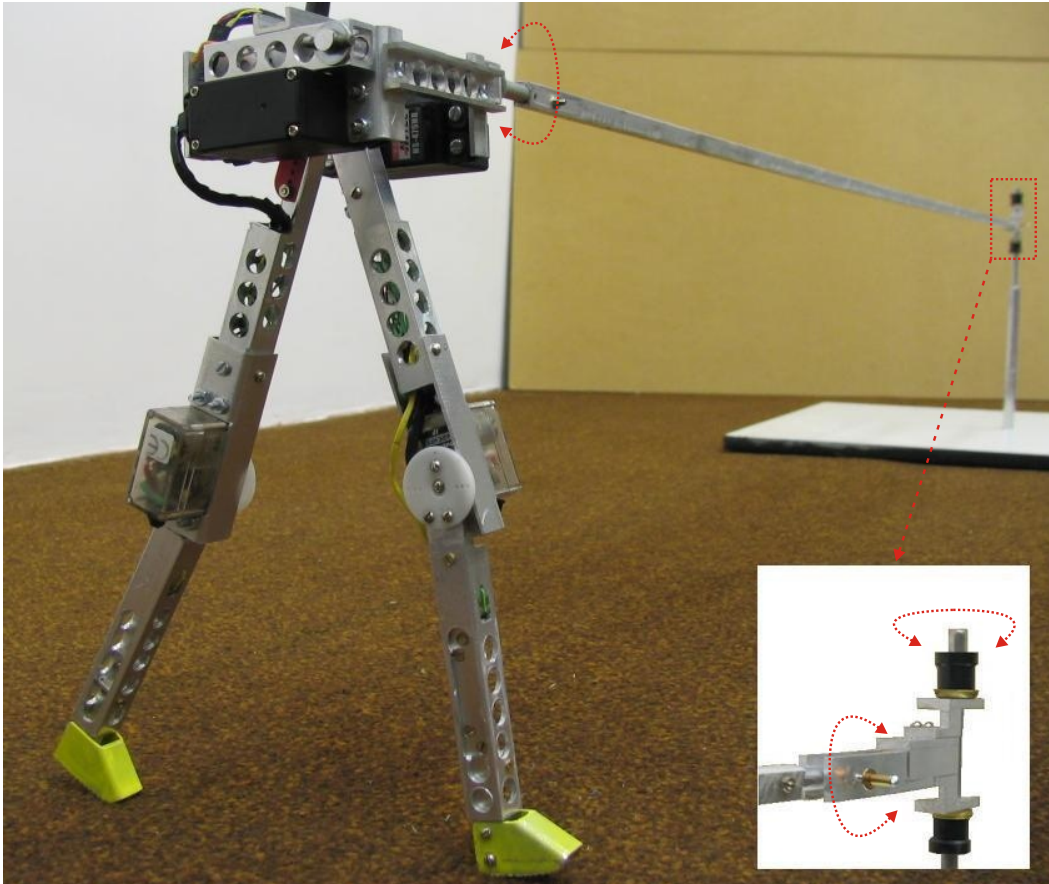


FIGURE 3.2: The robot, RunBot, and its boom structure. The three orthogonal axes of the boom indicated with curved arrows rotate freely.

being more efficient for fast walking. Since we intended to exploit its natural dynamics during some stages of its gait cycle, similar to passive bipeds and the previous design, its foot bottom is also curved with a radius equal to half the leg-length (with a too large radius, the tip of the foot may strike the ground during its swing phase). During the stance phase of such a curved foot, always only one point touches the ground, thus allowing the robot to roll passively around the contact point, which is similar to the rolling action of human feet. Therefore, with curved feet the difficulties caused by flat feet in fast walking can be avoided. However, how long should such a foot be? In theory, larger curved feet bring more stability for passive biped walking. In practice, however, large feet make foot clearance of the swing leg difficult, and tremendously limit the walking speed of the robot. In order to achieve a fast speed, RunBot is equipped with small feet (4.5 cm long) whose relative length, the ratio between the foot-length and the leg-length, is 0.20, less than that of humans (about 0.30) and that of other biped robots (powered or passive).

In fast human walking and running, the mass centre of the upper body is located a little

forward of the legs. As in the previous design, RunBot's mass centre is also located forward. To evaluate the effect of the location of the mass center on the walking speed, I have performed an additional simulation analysis (see Appendix B). According to the simulation results described in Appendix B, the mass centre of RunBot should be located 3 cm ahead of the hip joints.

In summary, our mechanical design of RunBot has the following special features that distinguish it from other powered biped robots and facilitate high-speed walking and exploitation of natural dynamics.

1. Small curved feet allowing for rolling action.
2. Unactuated, hence, light ankles.
3. Light-weight structure.
4. Light and fast motors.
5. Proper mass distribution of the limbs.
6. Properly positioned mass centre of the trunk.

### 3.3 The neural structure of RunBot's controller

↓-----

The controller of RunBot is a simplified version of the previous design in figure 2.5. The main difference between this design and the previous one is that the eight inter-neurons **R15(1)** have been removed. The model and synapses of the angle sensor neurons have also been changed, which will be described below. Despite of these changes, the controller can still realise the same logic function. The new design still follows a hierarchical structure (see figure 3.3). The bottom level represents the neuron modules local to the joints, including motor neurons and angle sensor neurons. The top level is a distributed neural network consisting of hip stretch receptors and ground contact sensor neurons, which modulate the motor neurons of the bottom level. ↑-----

The directions of the extensor (flexor) movements and the thresholds of the sensor neurons are illustrated in figure 3.4. At the bottom level, the function of the thresholds of the sensor neurons ( $\Theta_{ES,h}$ ,  $\Theta_{FS,h}$ ,  $\Theta_{ES,k}$ ,  $\Theta_{FS,k}$ , see figure 3.3 and figure 3.4) in each neuron module is to limit the extensor and flexor movements of the joint. At the top level, the functions of the AEA signal and the ground contact signal are shown in figure 3.5.

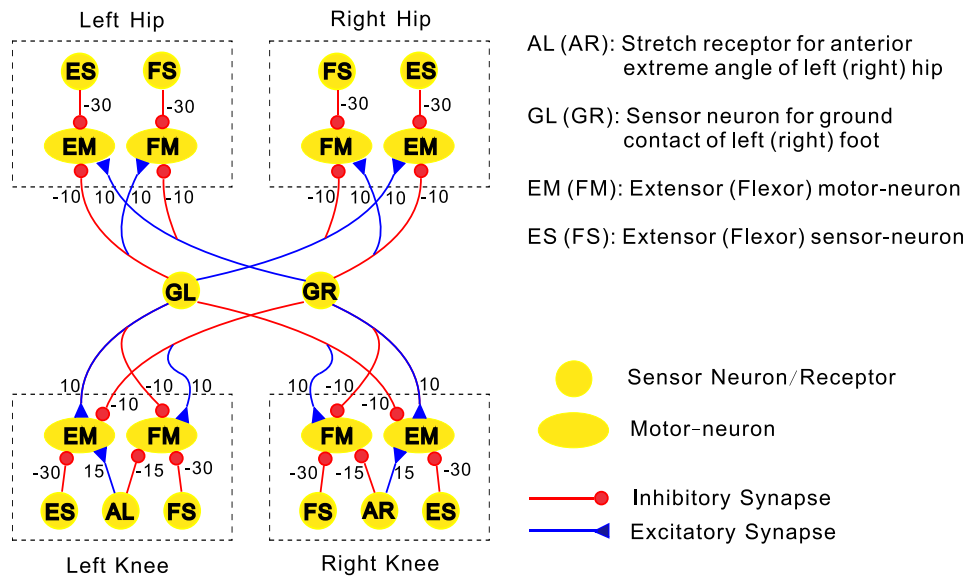


FIGURE 3.3: The neuron model of the controller on RunBot. The small numbers give the values of the connection weights.

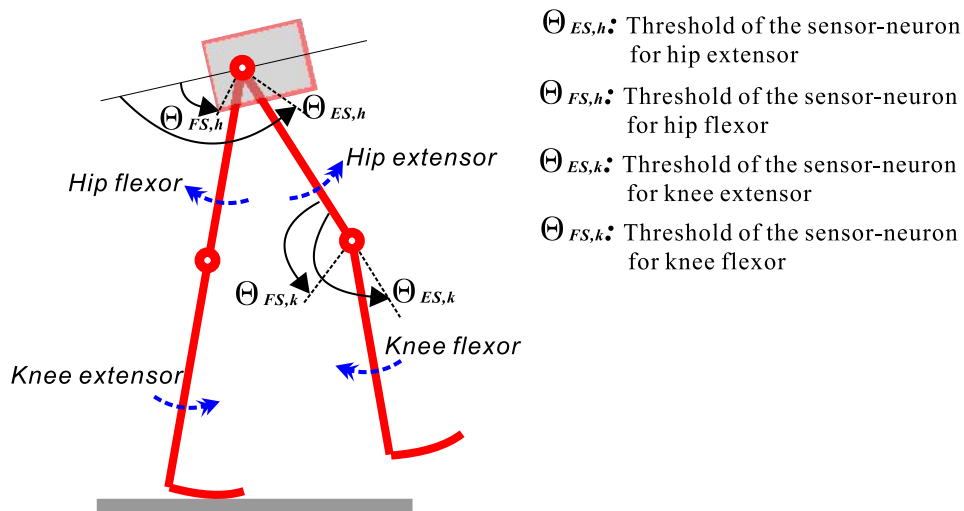


FIGURE 3.4: Control parameters for the joint angles.

The joint coordination mechanism in the top level is implemented with the neuron circuit illustrated in figure 3.3. The ground contact sensor neuron of each leg has excitatory connections to the motor neurons of the hip flexor and knee extensor of the same leg as well as to the hip extensor and knee flexor of the other leg. The stretch receptor of each hip has excitatory (inhibitory) connections to motor neuron of the knee extensor (flexor) in the same leg. Detailed models of the stretch receptor, and ground contact sensor neuron are the same as those described in the last chapter.



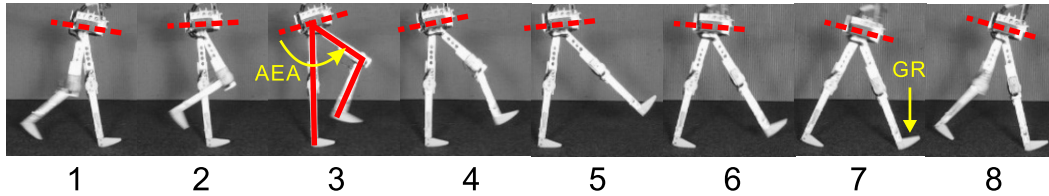


FIGURE 3.5: Series of frames of one walking step. At the time of frame (3), the stretch receptor (AEA signal) of the swing leg is activated, which triggers the extensor of the knee joint in this leg. At the time of frame (7), the swing leg begins to touch the ground. This ground contact signal triggers the hip extensor and knee flexor of the stance leg, as well as the hip flexor and knee extensor of the swing leg. Thus the swing leg and the stance leg swap their roles thereafter.

The neuron module for each joint is composed of two angle sensor neurons and the motor neurons they contact (see figure 3.3). Whenever its threshold is exceeded, the angle sensor neuron directly inhibits the corresponding motor neuron (see figure 3.3).

The model of angle sensor neurons is a little different from that of the previous design. The extensor angle sensor neuron changes its output according to:

$$\rho_{ES} = \frac{1}{1 + e^{\alpha_{ES}(\Theta_{ES} - \phi)}} \quad (3.1)$$

where  $\phi$  is the real time angular position obtained from the potentiometer of the joint (see figure 3.4),  $\Theta_{ES}$  is the threshold of the extensor motor neuron (see figure 3.4) and  $\alpha_{ES}$  a positive constant.

Likewise, the output of the flexor sensor neuron is modelled as:

$$\rho_{FS} = \frac{1}{1 + e^{\alpha_{FS}(\phi - \Theta_{FS})}} \quad (3.2)$$

with  $\Theta_{FS}$  and  $\alpha_{FS}$  similar as above.

The definition and direction of the joint angles is illustrated in figure 3.4. The direction of extensor on both hip and knee joints is forward while that of flexors is backward.

The motor neuron model is the same as that of the previous design. For convenience, we reproduce it below. The state and output of each extensor motor neuron is governed by equations 3.3 and 3.4 (Gallagher et al., 1996). Those of the flexor motor neurons are similar.

$$\tau \frac{dy}{dt} = -y + \sum \omega_X \rho_X \quad (3.3)$$

$$u_{EM} = (1 + e^{\Theta_M - y})^{-1} \quad (3.4)$$

where  $y$  represents the mean membrane potential of the neuron. Equation 3.4 is a sigmoidal function that can be interpreted as the neuron's short-term average firing frequency,  $\Theta_M$  is a bias constant that controls the firing threshold,  $\tau$  is a time constant associated with the passive properties of the cell membrane (Gallagher et al., 1996),  $\omega_X$  represents the connection strength from the sensor neurons and stretch receptors to the motor neuron (figure 3.3),  $\rho_X$  represents the output of the sensor neurons and stretch receptors that contact this motor neuron (e.g.,  $\rho_{ES}$ ,  $\rho_{AL}$ ,  $\rho_{GL}$ , etc.)

Note that, in RunBot, the output value of the motor neurons, after multiplication by a gain coefficient, is sent to the servo amplifier to drive the joint motors.

The voltage of the motor in each joint is determined by:

$$V = MG_M(S_E u_E + S_F U_F), \quad (3.5)$$

where  $M$  represents the magnitude of the servo amplifier, which is 3 on RunBot,  $G$  stands for the output gain of the motor neurons in the joint,  $S_E$  and  $S_F$  are signs for the motor voltage of flexor and extensor in the joint, being  $+1$  or  $-1$ , depending on the polarity of the motors,  $U_E$  and  $U_F$  are the outputs of the motor neurons (see figure 3.3).

### 3.3.1 Tuning the neuron parameters

Most of the values for the neuron parameters are chosen by trial and error. In this subsection, we address the tuning of the various neuron parameters except two parameters at the hip joints,  $\Theta_{ES,h}$  (see figure 3.4) and  $G_{M,h}$  (the gain of the motor neurons in hip joints), which will be tuned in the experiments below.

The positive constants of the sensor neurons and the stretch receptors ( $\alpha_{ES}$ ,  $\alpha_{FS}$ ,  $\alpha_{AL}$ ,  $\alpha_{AR}$ ,  $\alpha_{GL}$ ,  $\alpha_{GR}$ ) affect their response speed. We set these constants to 2, ensuring a quick response of these neurons.

The threshold of the sensor neurons for the extensor (flexor) in the neuron module roughly limits the movement range of the joint. The thresholds of these sensor neurons in the neuron modules of the knee joints are chosen as:  $\Theta_{FS,k} = 110^\circ$ ,  $\Theta_{ES,k} = 175^\circ$  (see figure 3.4), which is in accordance with the observation of humans' normal gaits. The movements of the knee joints are needed mainly for timely ground clearance without big contributions to the walking speed. After some trials, we set the gain of the motor neurons in knee joints to be  $G_{M,k} = 0.9G_{M,h}$ . The threshold of the stretch receptors is simply chosen to be the same as that of the sensor neurons for the hip extensor,  $\Theta_{AL(AR)} = \Theta_{ES,h}$ .

The threshold of the ground contact sensor neurons is chosen to be 2 volt according to test results on the piezo sensors. In a certain range, the output voltage of the piezo sensor is roughly proportional to the pressure acted on the foot bottom when it is touching the ground. The time constant of the motor neurons,  $\tau$  (see equation 3.4), is chosen as 10 ms, which is in the normal range of data in biology.

To simplify the problem, we also fix the threshold of the flexor sensor neurons of the hips ( $\Theta_{FS,h}$ ) to  $85^\circ$ . There are three kinds of synapses in the neuronal controller (see figure 3.3). Here we use following symbols to represent the absolute value of the weights of these synapses:  $W_{GM}$  is weights of the synapses between the ground contact sensor neurons and the motor neurons,  $W_{AM}$  is weights of the synapses between the stretch receptors and the motor neurons,  $W_{SM}$  is weights of the synapses between the angle sensor neurons and the motor neurons in the neuron modules of the joints.

The threshold of the motor neurons,  $\Theta_M$  (see equation 3.4), can be any positive value as long as following conditions are satisfied:

$$\begin{aligned} W_{GM} &\geq \Theta_M + 4 \\ W_{AM} - W_{GM} &\geq \Theta_M + 4 \\ W_{SM} - W_{AM} - W_{GM} &\geq \Theta_M + 4 \end{aligned}$$

The function of these rules is to make sure that, among all the neurons which contact the motor neurons, the angle-sensor neurons in the neuron modules of each joint have the first priority while the stretch receptors have second priority and the ground contact sensor neurons have the lowest priority. So, we simply choose them as:  $\Theta_M = 1$ ,  $W_{GM} = 10$ ,  $W_{AM} = 15$ ,  $W_{SM} = 30$  (see figure 3.3).

Obviously, the function of this neuronal controller can also be realised with a simple mode-switching controller. We prefer using model neurons for two reasons: first, the passive properties of the cell membrane (see equation 3.4) can naturally make the output of the neuronal controller smoother, thus reducing the jerk in the joint movement. Second, our long-term aim is to investigate the effect of neuronal plasticity on the walking behavior with a biped robot. Neuronal plasticity will be embodied by a high-level neural structure, which then can be seamlessly connected with this neuronal controller. ↓-----  
**R9(2)**  
↑-----

### 3.4 Runbot walking experiments

In the experiments described below, we now only need to tune the two parameters of the hip joints: the threshold of the extensor sensor neurons ( $\Theta_{ES,h}$ ) and the gain of the

motor neurons ( $G_{M,h}$ ). They work together to determine the walking speed and the gait properties of RunBot. In experiments on walking on a flat floor, we have found that stable gaits can appear in a considerably large range of the parameters  $\Theta_{ES,h}$  and  $G_{M,h}$  (see figure 3.6).

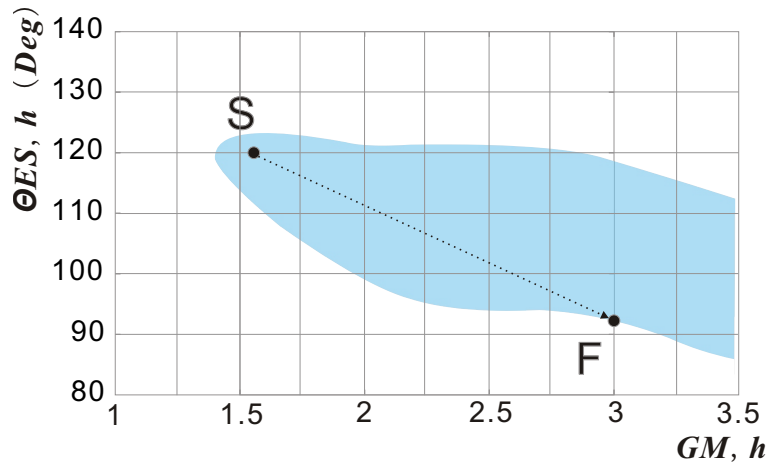


FIGURE 3.6: The shaded areas are the range of the two parameters, in which stable gaits appear. The maximum permitted value of  $G_{M,h}$  is 3.45 (higher values will destroy the motor of the hip joint). For the meanings of points S and F, see the text.

### 3.4.1 Changing speed on the fly

RunBot's walking speed can be changed on the fly without problems by tuning  $\Theta_{ES,h}$  and  $G_{M,h}$  as long as they still remain in the stable area shown in figure 3.6. Figure 3.7 shows the gait when the parameters are changed greatly and abruptly from point S to F (see figure 3.6) at time  $t$  (indicated with a line in figure 3.7). The walking speed is immediately changed from slow (0.38m/s) to fast (0.70m/s). By exploiting the natural dynamics, the neuronal controller is robust to such drastic parameter variations as shown in figure 3.6. The video clip of this experiment can be seen at, <http://www.cn.stir.ac.uk/~tgeng/runbot/speedchange.mpg>

### 3.4.2 Walking on irregular terrain

With parameters in the central area in figure 3.6, the walking gait shows more robustness. As shown in figure 3.8, RunBot can walk over a low obstacle with a height of 0.9 cm. Figure 3.9 shows a stick diagram of RunBot's gait walking down a shallow slope of 5 degree. Note that RunBot can neither detect the disturbance nor adjust any parameters

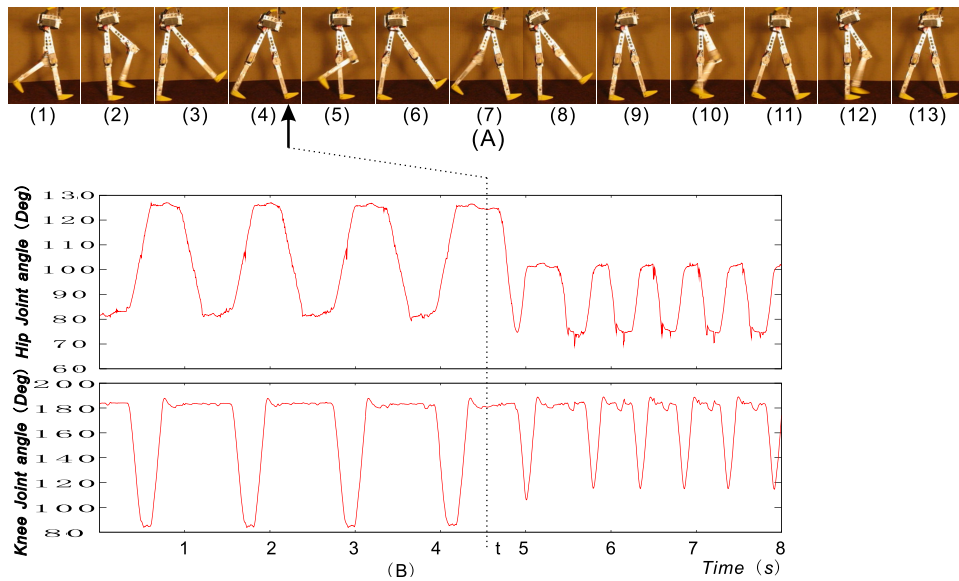


FIGURE 3.7: (A) Series of sequential frames of the walking gait. The neuron parameter is changed at the time of frame (4). The interval between two adjacent frames is 133 ms. (B) Real-time data of the angular position (in trunk coordinates as illustrated in figure 3.4) of hip joint and knee joint of one leg (indicated with an arrow in frame (4) of (A)) while the walking speed is changed at time  $t$ .

of its controller to address it. Nonetheless, after the disturbance, the walking gait returns soon to its normal orbit, demonstrating that the walking gait is to some degree robust against disturbances.

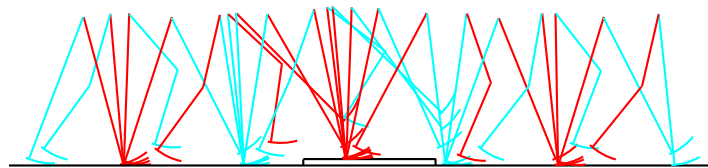


FIGURE 3.8: Stick diagram of RunBot walking over a low obstacle (9mm high, higher ones cannot be tackled). The interval between any two consecutive snapshots is 100 ms.

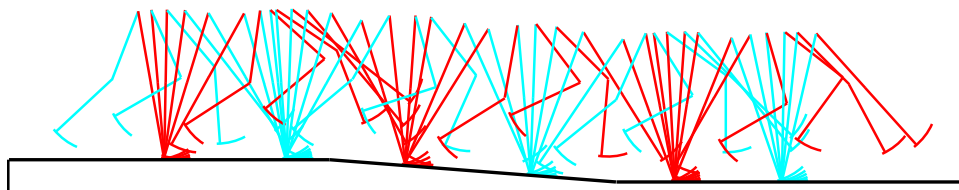


FIGURE 3.9: Stick diagram of RunBot walking down a shallow slope of 5 degree. The interval between any two consecutive snapshots is 67 ms.

### 3.5 Fast walking with online policy searching

Because there is no position or trajectory tracking control in RunBot, it is impossible to control its walking speed directly. Moreover, the neuronal controller does not employ any form of dynamics model of the robot, ruling out the possibility to analytically describe the relationship between neuronal parameters and the walking speed. On the other hand, as the robot's walking is stable in a quite large range of its controller parameters, it is not a good idea to evaluate its speed in the entire space of the controller parameters. However, knowing that RunBot's walking gait is determined almost exclusively by two parameters,  $\Theta_{ES,h}$  and  $G_{M,h}$  (see figure 3.10), we can formulate RunBot's fast walking control as a policy gradient reinforcement learning problem by considering each point in the parameter space (figure 3.10) as an open-loop policy that can be executed by RunBot in real-time. While walking continuously, RunBot can evaluate the policies selected by this online learning algorithm and finally find the optimal policy that generates the fastest walking speed.

↓-----

**R18(1)**

↑-----

Our approach is similar to that of Kohl and Stone (2004), except for the algorithms for adaptive step size and for local optimum avoiding, which are designed particularly for the biped walking in RunBot. Learning starts from an initial parameter vector  $\pi_0 = (\theta_1, \theta_2)$  (here  $\theta_1$  and  $\theta_2$  represent  $G_{M,h}$  and  $\Theta_{ES,h}$ , respectively) and proceeds to evaluate the following five policies at or near  $\pi$ :

$$\begin{aligned} R_1 &= (\theta_1, \theta_2) \\ R_2 &= (\theta_1, \theta_2 - \epsilon_2) \\ R_3 &= (\theta_1 - \epsilon_1, \theta_2) \\ R_4 &= (\theta_1, \theta_2 + \epsilon_2) \\ R_5 &= (\theta_1 + \epsilon_1, \theta_2) \end{aligned}$$

where each  $\epsilon_j$  is a fixed value that is small relative to  $\theta_j$ . Each policy represents a point in the parameter space (see figure 3.10), which corresponds to a specific gait of the robot. Thus,  $R_2, R_3, R_4, R_5$  are four points around  $R_1$  in the parameter space. The evaluation of each policy generates a score,  $S_{R_i}$ , that is a measure of the speed of the gait described by that policy ( $R_i$ ). We use these scores to construct an adjustment vector  $A$  (Kohl and Stone, 2004):

↓-----

**R16**

↑-----

$$A_1 = 0 \text{ if } S_{R_1} > S_{R_3} \text{ and } S_{R_1} > S_{R_5}$$

$$A_1 = S_{R_5} - S_{R_3} \text{ otherwise}$$

Similarly,

$A_2 = 0$  if  $S_{R1} > S_{R2}$  and  $S_{R1} > S_{R4}$

$A_2 = S_{R4} - S_{R2}$  otherwise

If  $A = 0$ , this means a possible local optimum is encountered. In this case, we replace  $A$  with a stochastically generated vector. Although this is a very simple strategy, our experiments show that it can effectively prevent the real-time learning from getting trapped locally.

Then  $A$  is normalised and multiplied by an adaptive step-size:

$$\eta = \eta_0(v_{max} - s_{max})/v_{max} \quad (3.6)$$

where  $v_{max}$  stands for the maximum speed RunBot has ever attained during the time before,  $s_{max}$  is the maximum value of  $S_{Ri}$  of this current iteration,  $\eta_0$  is a constant ( $\eta_0 = 3$ ). If  $\eta < \eta_{min}$  (or  $\eta > \eta_{max}$ ), it is set to be  $\eta_{min}$  (or  $\eta_{max}$ ),  $\eta_{min}$  and  $\eta_{max}$  are predefined lower and upper limits for  $\eta$  ( $\eta_{min} = 0.5$ ,  $\eta_{max} = 5$ ).

We use a sensor at the central axis of the boom to measure the angular speed of the boom when RunBot is walking, from which the walking speed can be calculated. To get an accurate speed, each policy is executed for  $N_{cyc}$  gait cycles (one gait cycle includes two steps). Because the speed of the first gait cycle of each policy is still influenced by the last policy, it is neglected and the average speed of these  $N_{cyc} - 1$  cycles is regarded as the speed of the gait corresponding to this policy. At the beginning of the learning process,  $N_{cyc}$  is set to be 2. Then  $N_{cyc}$  is recalculated at the end of each iteration according to the following rule:

$$N_{cyc} = (int)((v_{max} - v_{min})/3)$$

where  $v_{min}$  stands for the minimum speed RunBot has ever attained during previous gait cycles.

Finally,  $A$  is added to  $\pi_0$ , obtaining a new parameter vector,  $\pi_1$ , and the next iteration begins. Results are shown in figure 3.10A and figure 3.11. RunBot starts walking with parameters at point S in figure 3.10A corresponding to a speed of 41 cm/s (see figure 3.11C). After 240 seconds of continuous walking with the learning algorithm and no human intervention, RunBot attains a walking speed of about 80 cm/s (see figure 3.11C), which is equivalent to 3.5 leg-lengths per second. Figure 3.12 shows video frames of walking gaits at a fast and a medium speed, respectively, in which we can clearly see the change of gaits during the process of the learning.

In the second experiment, RunBot starts walking with different parameters corresponding

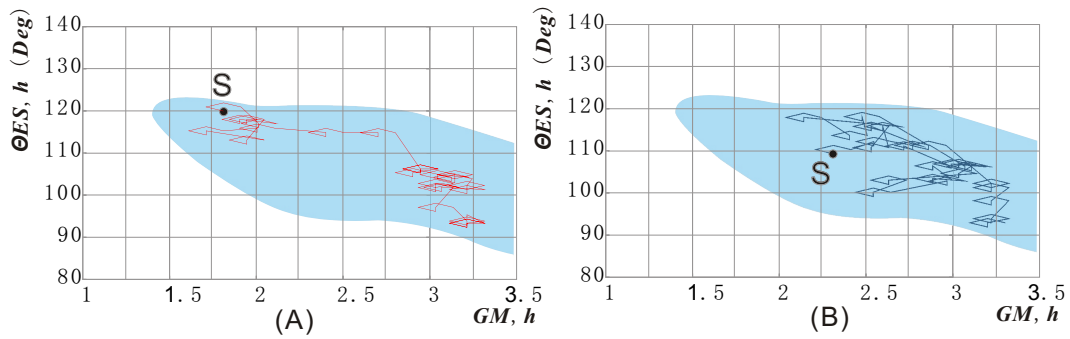


FIGURE 3.10: Changing of the controller parameters,  $G_{M,h}$  and  $\Theta_{ES,h}$ , during two experiments of online learning. See text for more information.

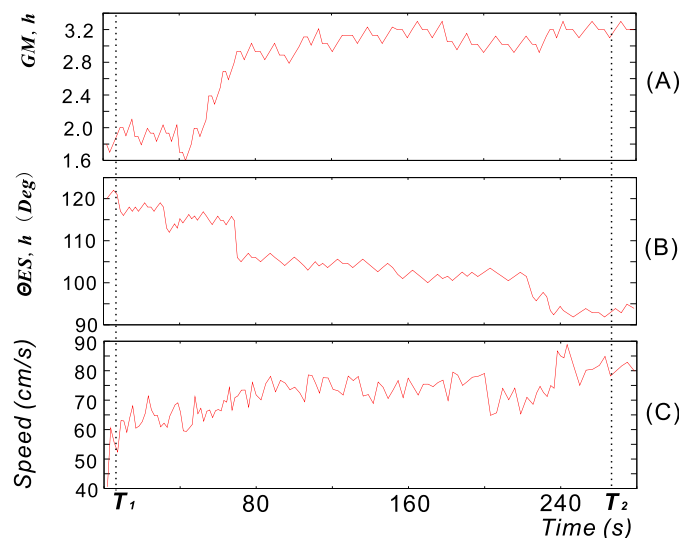


FIGURE 3.11: Real-time data of one experiment. Changes of the controller parameters ((A) and (B)) and the walking speed (C) during the entire process of learning.

to point S in figure 3.10B. The data of this experiment are shown in figure 3.13. In 280 seconds, the robot also attains a speed of around 80 cm/s (see figure 3.13).

In the two experiments on online learning reported above, learning started from policies located in the upper or middle part of the stable area (see figure 3.10). In this case, the subsequent policies usually do not exceed the boundaries of the stable area. But, in some cases, if learning starts from a policy near the lower boundary of the stable area, subsequent policies can indeed sometimes leave the stable area. To prevent this, we use the following strategy: At the beginning of a iteration (e.g., the  $i_{th}$  iteration), if any of the five policies that will be evaluated at or near  $\pi_i$  is located outside the stable area, the vector  $\pi_i$  is replaced with another vector in the stable area,  $\hat{\pi}_i$ , which is nearest to  $\pi_i$  on the coordinate of  $\theta_1$  ( $\theta_2$ ), and has a distance of  $\epsilon_1$  ( $\epsilon_2$ ) to the boundary of the stable area



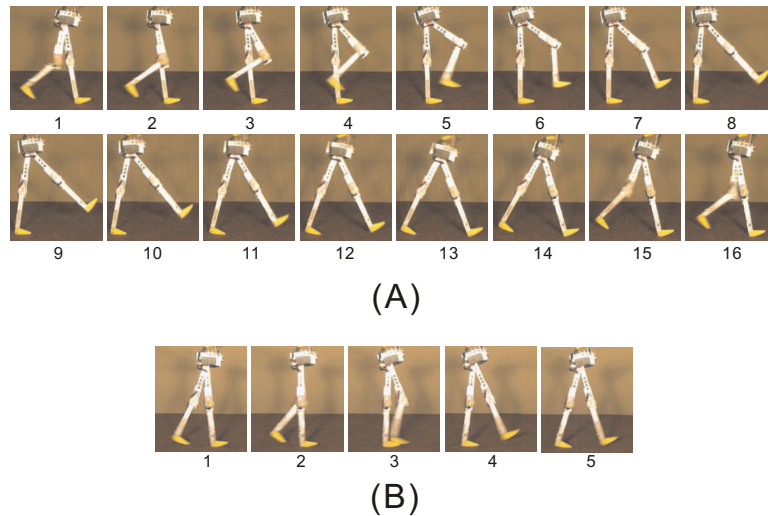


FIGURE 3.12: Series of sequential frames of two walking gaits. The interval between two adjacent frames is 33 ms. (A). Gait of a medium speed (53 cm/s), the parameter values of which are indicated as  $T_1$  in figure 3.11. Note that, during the time between frame 8 and frame 13, which is nearly one third of the duration of a step, the whole robot is moving unactuatedly. At the time of frame 13, the swing leg touches the floor and a next step begins. (B). Gait of a fast speed (80cm/s), the parameter values of which are indicated as  $T_2$  in figure 3.11.

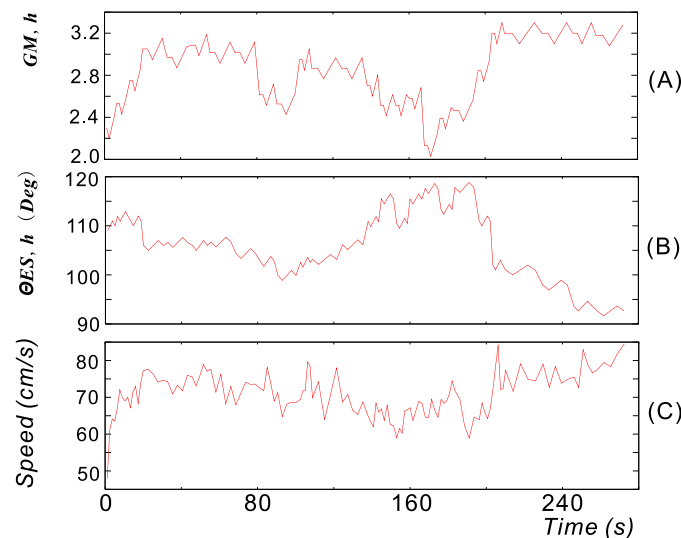


FIGURE 3.13: Real-time data of another experiment. Changes of the controller parameters ((A) and (B)) and the walking speed (C) during the entire process of learning.

(see figure 3.14).

In the experiments, self-stabilising properties as a result of increasing speed, such as those suggested by Seyfarth and Blickhan (2002) and Poulakakis and Buehler (2003) in

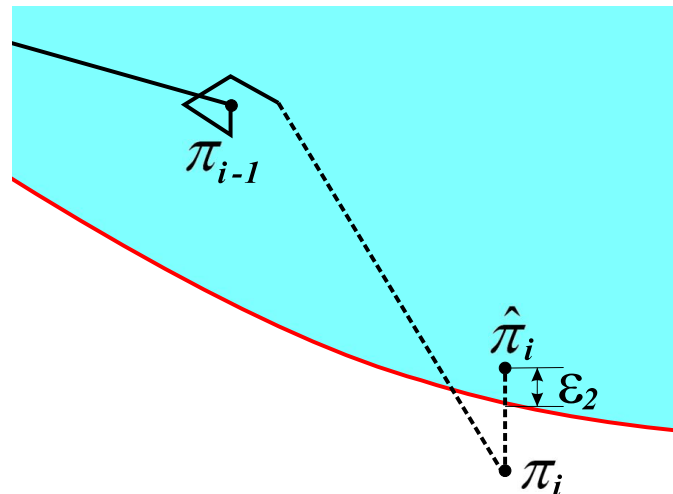


FIGURE 3.14: If the parameter vector  $\pi_i$  is not appropriate, it will be “pushed” back into the stable area. See text for more information.

monopod and quadruped walking, only seem to happen to a limited degree when starting the learning from a policy near the upper boundary or middle of the stable area. It is usually a puzzling problem how to measure the stability of the walking robots (like RunBot) that do not use any kind of dynamic model. The eigenvalues of the linearised Poincaré map are often used for stability analysis of walking robots (Garcia, 1999). In simulations, the eigenvalues of the linearized Poincaré map can be calculated by minutely perturbing the robot from the fixed point in each dimension. In real robots, however, the lack of sufficient and accurate sensor signals make this kind of idealized analysis very difficult (if not impossible). To build the Poincaré map of RunBot’s gait, we need both the position and the speed data of the four actuated joints and the unactuated stance ankle joint. But, only the position data of the four actuated joints is available. Even on these four joints, due to the noise and inaccuracy of the potentiometers, measuring tiny perturbations is almost impossible.

### 3.6 An improved design of RunBot to climb slopes

The biggest slope that can be overcome by RunBot and the previous design is at around 4 degree, which is quite small. This limitation of their performance is due to the lack of an upper body and actuated ankle joints. When climbing up a slope, humans usually lean their upper body forward to maintain the balance of the gait. In our recent work, we have added an upper body in RunBot’s design (see figure 3.15). The upper body is controlled by a reflex module triggered by the accelerometer sensor that can sense the

lean movement of the robot. With this new design, it can overcome a slope of 7.5 degree (see figure 3.15).

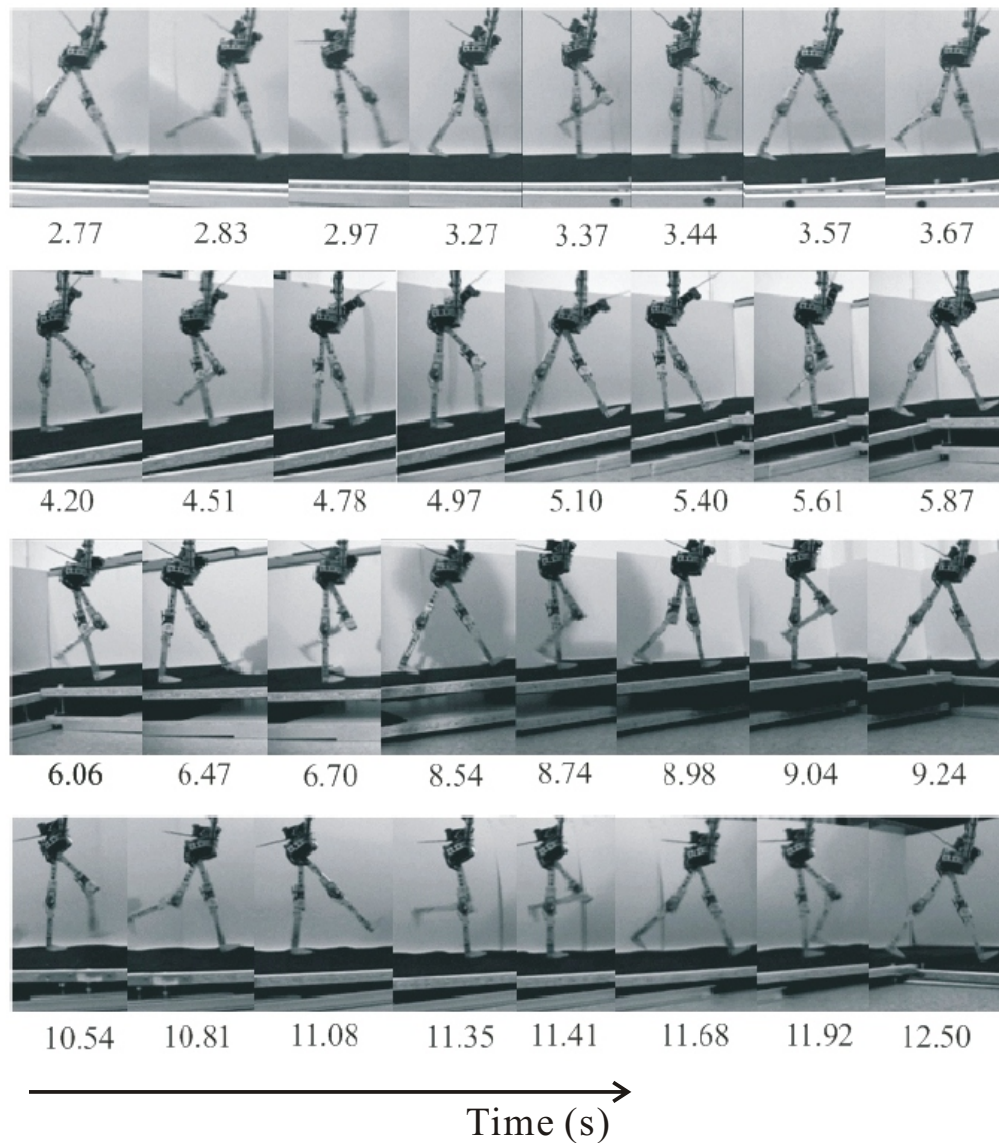


FIGURE 3.15: With a upper body, the robot can climb up a slope of 7.5 degree.

R12

↑-----

### 3.7 Comparing the speed of robots and humans

To compare the walking speed of various biped robots whose sizes are quite different from each other, we use the relative speed, speed divided by the leg-length. Maximum relative speeds of RunBot and some typical planar biped robots (passive or powered) are

shown in figure 3.16. We know of no other biped robot attaining such a fast relative speed. The world record for human walking is equivalent to about 4.0–4.5 leg-lengths per second. So, RunBot’s highest walking speed is comparable to that of humans. To get a feeling of how fast it can walk, we strongly encourage readers to watch a video clip at, <http://www.cn.stir.ac.uk/~tgeng/runbot/learning.mpg>



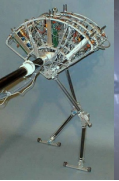



						
Leg length (m)	0.8	0.75	0.9	0.8	0.23	0.85-1.0
Max. speed (m/s)	0.6	0.40	1.25	1.20	0.80	4.21
<b>Relative Speed</b>	<b>0.7</b>	<b>0.5</b>	<b>1.4</b>	<b>1.5</b>	<b>3.5</b>	<b>4-5</b>
<b>Froude Number</b>	<b>0.05</b>	<b>0.02</b>	<b>0.17</b>	<b>0.18</b>	<b>0.28</b>	<b>1.8-2.1</b>
	(a)	(b)	(c)	(d)	(e)	(f)

FIGURE 3.16: Relative leg-length and Maximum relative speed of various planar biped robots. (a) A copy of McGeer planar passive biped robot walking down a slope (Wisse and van Frankenhuyzen, 2003). (b) “Mike”, similar to McGeer’s robot, but equipped with pneumatic actuators at its hip joints. Thus it can walk half-passively on level ground (Wisse and van Frankenhuyzen, 2003). (c) “Spring Flamingo”, a powered planar biped robot with actuated ankle joints (Pratt, 2000). (d) “Rabbit”, a powered biped with 4 degree-of-freedom and point feet. (e) RunBot. (f). Olympic record of human’s walking speed.

### 3.8 Froude number

Biped robots can help us to understand the biomechanics of human walking if their gaits are dynamically similar. The Froude number,  $Fr$ , has been used to describe the dynamical similarity of legged locomotion over a wide range of animal sizes and speeds on earth (Alexander and Jayes, 1983).

$$Fr = v^2/gl \quad (3.7)$$

where  $v$  is the walking speed,  $g$  gravity acceleration and  $l$  leg-length. The Froude number is only a prerequisite for dynamic similarity between the locomotion of different animals. Aside from having the same Froude number, dynamically similar animals must also meet the following conditions (Donelan and Kram, 1997):

1. They must move their legs with the same phase.

2. Their corresponding feet must have equal duty factors.
3. They must have equal relative stride length.
4. Their corresponding feet must exert forces that are equal multiples of body weight at corresponding points in the stride.
5. They must have mechanical power outputs proportional to the product of body weight and forward speed.

The Froude number of some typical biped robots are listed in figure 3.16, most of which are far below the normal value of the adult human's value of 0.20 (Vaughan and O'Malley, 2005), indicating that they are indeed not dynamically similar to adult humans, though some of them have been designed to mimic human walking. However, 0.20 is in the attainable range of RunBot's Froude number, implying that its walking gait, when at an appropriate speed (0.67m/s), could with some confidence be regarded as dynamically similar to that of an adult human. But, to precisely measure the similarity between RunBot and human gait using the conditions described above, we need many additional speed and force information, which are currently unavailable in the robot. We will deal with this question in our future work.

**R17**

↑-----

### 3.9 Conclusion

In this study, we have shown that, with a properly designed mechanical structure and a simple neuronal controller, our biped robot can attain a fast relative walking speed of 3.5 leg-lengths per second, which is not only faster than any other biped walking robot we know, but also comparable to human's fastest walking speed.

↓-----

To deal with the difficulty in RunBot's speed control caused by the lack of dynamics model in the neural controller, we have successfully applied an online reinforcement learning on top of the neural controller, and thus make RunBot attain its fast speed in an autonomous way by itself.

↑-----

**R18(2)**

# Chapter 4

## Biped walking with a neuromuscular-like controller

### 4.1 Introduction

While biped robots usually depend on position control or trajectory tracking control at least in some stages of their walking gaits, humans and animals use a totally different “technology”, neuromuscular control, by which stable walking gaits emerge from the global entrainment between the neuro-musculo-skeletal system and the environment (Taga, 1995). Moreover, as introduced in the first chapter, muscle properties can simplify the control structure and smooth the movements.

A growing number of simulation studies have focused on dynamic models of the neuro-musculo-skeletal systems (Taga, 1995), which combine CPGs and musculo-skeletal models of different degrees of complexity and fidelity. However, complex systems, like a dynamic biped under neuromuscular control, cannot be fully simulated because of uncontrollable contingencies in its biomechanics and in the world in which it is embedded. In this chapter, we will implement a simple muscle model in RunBot’s controller, investigating what improvements the muscle model can bring for RunBot’s walking gaits.

### 4.2 Implementing a muscle model in the robot

RunBot’s controller and neuronal parameters remain unchanged from its previous version **R15(2)** (figure 3.3) except:

↓-----

1. In order to make the movements of the thighs symmetric around the trunk like human's walking gaits, we add this constraint,

$$\Theta_{ES,h} + \Theta_{FS,h} = 180^\circ \quad (4.1)$$

where  $\Theta_{ES,h}(\Theta_{FS,h})$  is the threshold of the extensor (flexor) sensor neurons of the hip joints.

2. Now the outputs of the motor neurons directly drive a muscle model instead of the motors of the joints (see figure 4.1). The muscle model is realised with a control algorithm implemented on the motors of the joints, which will be described below. **R15(2)**

↑-----

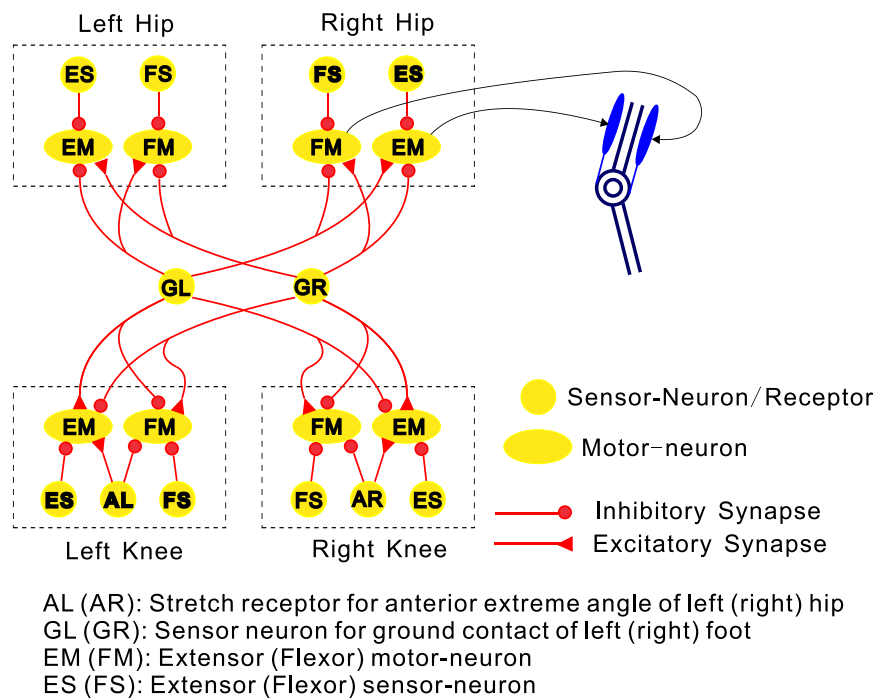


FIGURE 4.1: The neuron model of reflexive controller on RunBot. Only the muscle pair of one joint is illustrated. Other joints are omitted.

We use a linear viscous elastic muscle model composed of a spring in parallel with a viscous damper, directly controlled by the motor neuron output (Wadden and Ekeberg, 1998). Each joint has an antagonistic muscle pair of flexor and extensor, which are activated by extensor and flexor motor neuron, respectively (see Fig 4.1). The torque exerted by the muscle pair on one joint,  $T_1$ , can be described as (Wadden and Ekeberg, 1998),

$$T_1 = K(\phi_1 - L) - \delta\dot{\phi}_1 \quad (4.2)$$

$$K = \beta(M_E + M_F + \gamma) \quad (4.3)$$

$$L = \alpha(M_E - M_F)/K + L_0 \quad (4.4)$$

where  $K = 6$  is the effective stiffness of the muscle pair,  $L = 90^\circ$  the resting angle of the muscle pair, and  $L_0 = 10^\circ$  an offset,  $\alpha = 4$  and  $\beta = 1.5$  are gains,  $\gamma = 0.1$  is a base level of stiffness,  $\delta = 11$  the damping coefficient. For details of this muscle model, see Wadden and Ekeberg (1998). Before we can implement this muscle model in RunBot, we have to do some analysis on the dynamics of the robot and its joint motors.

The dynamics of RunBot are modelled as shown in Fig 4.2, which are similar to those used in the simulation analysis of previous chapters. For convenience, we reproduce them below.

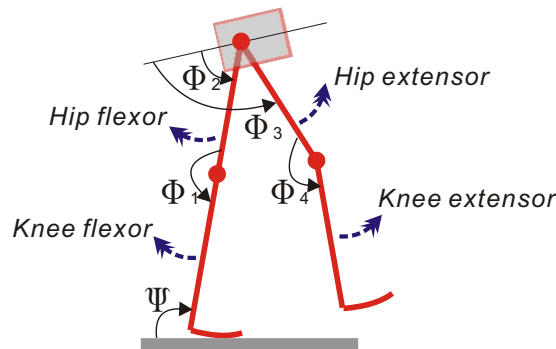


FIGURE 4.2: Dynamics model of RunBot and the definitions of its joint angles.

With the Lagrange method, the equations that govern the motion of the robot are:

$$D(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau \quad (4.5)$$

where  $q = [\phi_1, \phi_2, \phi_3, \phi_4, \psi]^T$  is a vector describing the configuration of the robot (for definition of  $\psi, \phi_1, \phi_2, \phi_3, \phi_4$ , see Fig 4.2).  $D(q)$  is the  $5 \times 5$  inertia matrix,  $C(q, \dot{q})$  is the  $5 \times 1$  vector of centripetal and coriolis forces,  $G(q)$  is the  $5 \times 1$  vector representing gravity forces,  $\tau = [\tau_1, \tau_2, \tau_3, \tau_4, 0]^T$ ,  $\tau_1, \tau_2, \tau_3, \tau_4$  are the torques applied on the joints.

The dynamics of the DC motor (including gears) of each joint can be described with the following equations. The joint indicated with  $\phi_1$  in Fig 4.2 is taken as an example. The



models of other joints are likewise:

$$L_a \frac{di_a}{dt} + R_a i_a + nk_3 \dot{\phi}_1 = V_1 \quad (4.6)$$

$$\tau_1 + I_1 \ddot{\phi}_1 + k_f \dot{\phi}_1 = nk_2 i_a \quad (4.7)$$

where,  $V_1$  is the applied armature voltage of the motor,  $i_a$  is the armature current,  $L_a$  the armature inductance,  $R_a$  the armature resistance,  $k_3$  is the emf constant,  $k_2$  is the motor torque constant,  $I_1$  is the combined moment of inertial of the joint motor and gear train referred to the gear output shaft,  $k_f$  is the vicious-friction coefficient of the combination of the motor and the gear,  $n$  is the gear ratio.

Considering that the electrical time-constant of the motor is much smaller than the mechanical time-constant of the robot, we neglect the dynamics of the electrical circuits of the motor, which leads to  $\frac{di_a}{dt} = 0$ . Thus equation 4.6 is reduced to,

$$i_a = \frac{1}{R_a} (V_1 - nk_3 \dot{\phi}_1) \quad (4.8)$$

We suppose

$$\hat{\tau} = \tau + I\ddot{q} \quad (4.9)$$

$$\text{where } \hat{\tau} = \begin{bmatrix} \hat{\tau}_1 \\ \hat{\tau}_2 \\ \hat{\tau}_3 \\ \hat{\tau}_4 \\ 0 \end{bmatrix}, I = \begin{bmatrix} I_1 & 0 & 0 & 0 & 0 \\ 0 & I_2 & 0 & 0 & 0 \\ 0 & 0 & I_3 & 0 & 0 \\ 0 & 0 & 0 & I_4 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$I_i$  is the combined moment of inertial of the joint motors and gear trains referred to the gear output shaft.

Combining equations 4.5 and 4.9, we can describe the dynamics of the robot in a new form with  $\hat{\tau}$  as the torque input.

$$(D(q) + I)\ddot{q} + C(q, \dot{q}) + G(q) = \hat{\tau} \quad (4.10)$$

Combining equations 4.7, 4.8, and 4.9,

$$\hat{\tau}_1 = \frac{nk_2}{R_a} V_1 - \left( \frac{n^2 k_2 k_3}{R_a} + k_f \right) \dot{\phi}_1 \quad (4.11)$$

We apply following control algorithm on the input voltage of this joint motor,

$$V_1 = \frac{R_a K}{nk_2}(\phi_1 - L) - \left(\frac{R_a(\delta - k_f)}{nk_2} - nk_3\right)\dot{\phi}_1 \quad (4.12)$$

Combining equations 4.11 and 4.12,

$$\hat{\tau}_1 = K(\phi_1 - L) - \delta\dot{\phi}_1 \quad (4.13)$$

Now from equations 4.2, 4.3, 4.4, 4.10, and 4.13, we can see that the muscle model has been implemented on this joint. Likewise, we can obtain the control algorithm for applying the muscle model on other joints.

### 4.3 Robot walking

With the muscle model implemented at its motors, RunBot's walking speed can be changed on the fly by tuning  $\Theta_{ES,h}$  (of course,  $\Theta_{FS,h}$  is also changed accordingly with equation 4.1). Fig 4.3 shows the relationship between  $\Theta_{ES,h}$  and the walking speed.

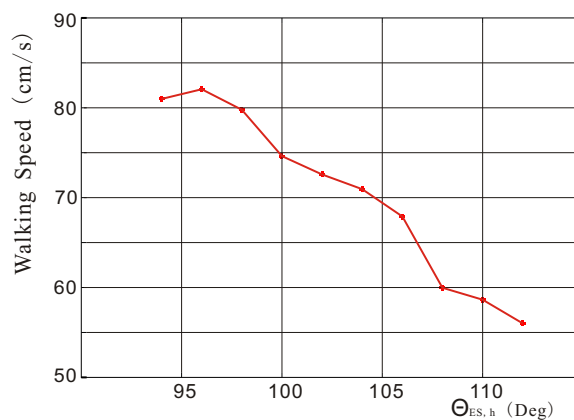


FIGURE 4.3: The change of walking speed while  $\Theta_{ES,h}$  is tuned manually. If  $\Theta_{ES,h} < 94^\circ$ , the robot tends to fall forward in a few steps. If  $\Theta_{ES,h} > 112^\circ$ , the robot ends to fall backward in a few steps.

Figure 4.4A shows the gait when  $\Theta_{ES,h}$  is changed greatly and abruptly from  $110^\circ$  to  $95^\circ$  at a time  $t$  (indicated with a line in figure 4.4B). The walking speed is immediately changed from 57 cm/s to 82 cm/s. Although there is no specifically designed controller in charge of the sensing and control of the transient stage of speed-changing, the natural dynamics of the robot itself and the muscle model properties

ensure stability during the change. The video clip of this experiment can be seen at, <http://www.cn.stir.ac.uk/~tgeng/smc/speedchange.mpg>

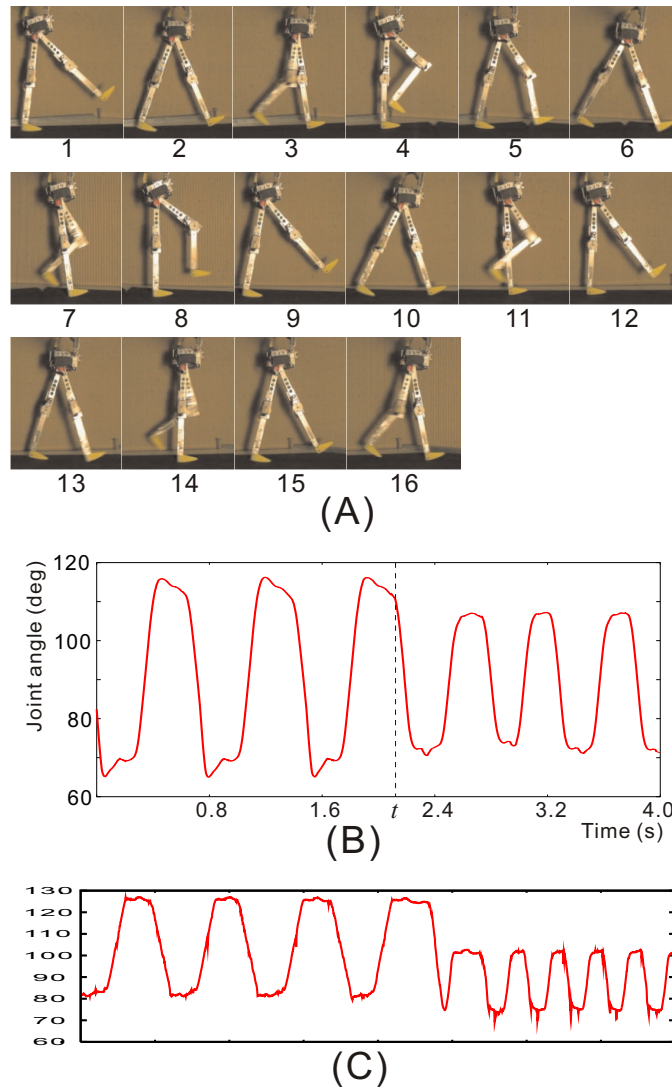


FIGURE 4.4: (A) Series of sequential frames of the walking gait,  $\Theta_{ES,h}$  is changed from  $110^\circ$  to  $95^\circ$  at the time of frame 10. The interval between two adjacent frames is 100 ms. (B) Real-time data of the angular position of one hip joint while the  $\Theta_{ES,h}$  is changed at time  $t$ . (C) The position of RunBot's left hips joint without using the muscle model (copied from Fig 3.7).

As shown in figure 4.5, with a small value of  $\Theta_{ES,h}$  ( $95^\circ$ ), RunBot can walk up a shallow slope. Figure 4.5 shows a series of frames of RunBot's gait walking up a shallow slope of 0.05 rad (any bigger slope cannot be overcome). Note, RunBot can neither detect the slope nor adjust any parameters of its controller to address it. For the video footage of this experiment, see <http://www.cn.stir.ac.uk/~tgeng/smc/slope.mpg>

Comparing the trajectory curves in figure 4.4(B) and 4.4(C), we can see the movement

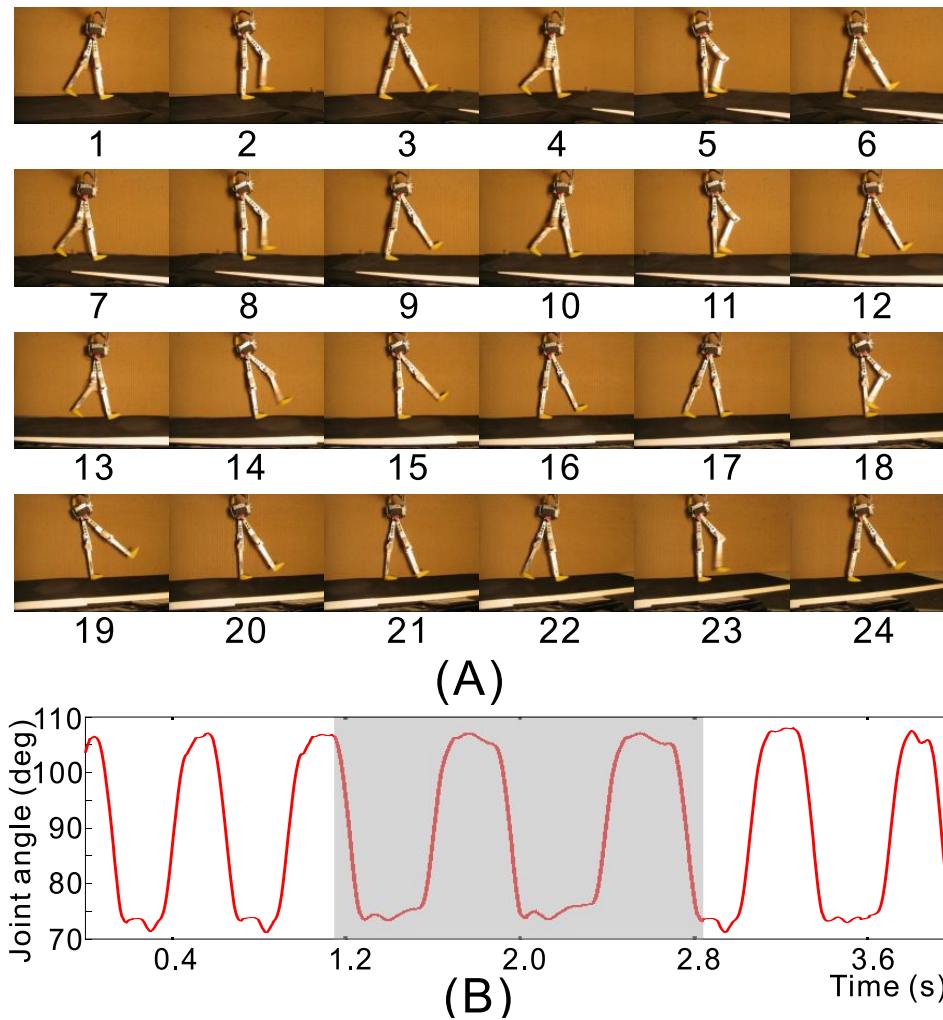


FIGURE 4.5: (A). Runbot is walking up a shallow slope of  $0.05$  rad. The time interval is  $100$  ms. Frames 1-9 are the normal gait on the flat floor. Frames 10-16 are the gait on the slope. The frames after 16 shows that the gaits return to normal on a flat surface. (B). Real-time data of the angular position of one hip joint while Runbot is walking up the slope (shaded area).

with the muscle model is a little smoother than before. But, the improvements in other aspects (e.g., the adaptivity on irregular terrain) are not evident. This can be attributed to the high gear-ratio of the RC motors used on the robot, which has caused difficulties in emulating the spring-damper property of the muscle model.

## 4.4 Conclusion

Using real-time experiments, this study has shown that fast dynamic biped walking can be achieved via neuromuscular-like reflexive controller without needing any trajectory control

a mechanisms. The natural dynamics of the robot and the viscous elastic muscle model implemented on its joints have contributed to the motion generation and smoothness of the walking gaits, thus simplifying the controller structure.

# Chapter 5

## Conclusions

### 5.1 Achievements

In this study, dynamic biped walking was realised with a simple neuronal controller using local reflexes and a minimal set of sensor signals. By means of real-time experiments, this thesis revealed the potential functions that physical computation can provide in biped locomotion control when coupled with a minimal neuronal controller, e.g., simplifying the computation for gait generation, smoothing the movements, and producing adaptivity in the gaits. In addition, a record-breaking fast walking speed has been achieved with a new design of our robot.

One common thread running through this thesis is the notion of physical computation, whereby a controller takes advantage of the natural dynamics in biped walking. In order to facilitate physical computation in our biped robots, we developed new structures in two aspects. First, in the controller design, we employed a minimal set of sensor signals triggering local reflexes. Unlike the sophisticated controller of other powered biped robots, the reflexive controller only has influence at some critical time of the walking gait cycle, leaving most of the gait generation tasks to the natural dynamics of the robot. Second, in the mechanical structure, we used some special designs (e.g., properly located centre of mass, small curved feet) that can enhance the stability and function of the natural dynamics. Although the design of the robot is very simple, it has immensely outperformed other biped robots in some aspects, e.g., the fast walking speed and the ability to change speed drastically.

One aim of this study was to understand human walking with a reduced mechanical structure rather than pure computer simulations. The question is, how much does the

robot walking resemble human walking? In the videos, the walking gait of the RunBot looks very natural, quite like human walking. Moreover, RunBot's gait can have the same Froude number as humans' normal walking gait. This means, RunBot is more likely to resemble human walking than other biped robots.

Another important issue addressed in this thesis is fast biped walking. While other robots using sophisticated modern control technologies have failed in their attempt to achieve high speed biped walking, RunBot's walking has demonstrated a record-breaking fast speed that is even comparable to human fast walking. This is another attestation of the argument made by Full and Tu (1990), that is, in fast locomotion, mechanical properties (physical computation) are more important than the adjustment of the neural system, and play a dominant role in gait generation.

## 5.2 Potential extensions

### 5.2.1 Extending the robot design to 3D

We used planar biped robots in our study because, in straight forward walking, the main characteristics and control problems of the gaits can be observed in the sagittal plane. The simplified planar biped walking in the sagittal plane can represent the essence of the biped walking. However, if we want to understand more realistic biped locomotion in the real world, we will have to use a 3D model or robot. The reflexive controller designed for RunBot can be used on a 3D biped robot in its sagittal plane as long as there is another controller in the frontal plane ensuring the lateral balance of the robot. A simple design is to use springy passive ankles in the frontal plane. The movements of the robot in the sagittal plane and frontal plane can both be regarded as something like an oscillation. If the oscillation in the frontal plane and the oscillation in the sagittal plane match each other well, the 3D gait will be stable. We can reach such a match between the two planes by adjusting the passive property in the frontal plane and the neuronal parameters in the sagittal plane.

### 5.2.2 Extending the learning capability

One good property of RunBot's reflexive controller is that it allows changing the walking gait by tuning just two neuronal parameters. This makes it easy to add a high level implementing autonomy or learning functions on top of the reflexive neuronal controller.

It is known that neurons can change their synaptic strength according to the temporal relation between their inputs and outputs. If the presynaptic signal arrives before the post-synaptic neuron fires, such a synapse gets strengthened, but it will get weakened if the order is reversed. Hence this form of plasticity depends on the timing of correlated spikes (STDP, spike timing-dependent plasticity). Other people in our group have used this property of STDP on top of the reflexive controller to stabilise the speed of the robot at a pre-set, desired value employing a plasticity rule which simulates spike timing-dependent plasticity.

### 5.2.3 Scaling up the robot design

Before building an aeroplane, engineers use a small model of the design to test its aerodynamics in a wind tunnel because the small model is dynamically similar to the aeroplane that will be built. Likewise, RunBot's design can be scaled up (at least in theory) to build a human-size fast biped robot as long as the dynamical similarity between them is retained. However, the current mechanical properties (e.g., mass distribution at legs and the location of the mass centre of the trunk) could be not optimal in a scaled-up design. To deal with this problem, we can use optimisation in simulation analysis to find a suitable design before we build the robot. The criterion for the optimisation will be the attainable fast speed and the stable range of the controller parameters.

↓-----

**R19(2)**

↑-----

## 5.3 Future work

In this work, we have shown that the natural dynamics integrated with a minimal neural controller can generate tremendously fast walking speed in a biped robot. In the future work, we will extend RunBot's neuro-mechanical design to realise fast biped running.

↓-----

**R19(1)**

Various sophisticated control technologies have been used in biped running robots, e.g., posture stabilization, inverted pendulum stabilization, contact torque control, foot vertical force control, torque distribution control. However, today's dynamic biped running robots are still well outperformed by their counterpart, humans, especially in speed and adaptivity. RunBot's success in fast walking is attributed to its properly designed mechanical structure that facilitates self-stabilising fast walking and the minimal neuronal controller that leaves most of the gait generation task to the natural dynamics. Similar to what we have demonstrated in RunBot's walking, the goal of our future work is to achieve fast biped running using a minimal neuronal controller coupled with natural



dynamics, which will be very different from many other biped robots based on various dynamic models and intensive feedback controls. Thus, the questions raised are: Is there any natural dynamics in biped running that can be exploited to tremendously simplify the control structure (just as in the case of RunBot's walking control)? What is the minimal controller that can generate stable and changeable running gaits?

We can find some cues to these questions in biomechanics literature. Biological experiments have shown that, in animals' high speed running, sensory signals are simply returned to the central nervous system too late to be effective. Mechanical properties of the musculoskeletal system play a dominant role (Dickinson and Farley, 2000). The spring-mass model has been proved to be an accurate tool for modelling animals' fast running, which can exhibit self-stabilizing running without any feedback control (Full and Tu, 1990). In fast running, the system has less time to recover from disturbances. The nonlinear, passive visco-elastic properties of the musculoskeletal system itself can respond rapidly to disturbances (Brown and Loeb, 1999). All these findings lead to a conclusion that the natural dynamics of the musculoskeletal system accounts for a large part of the gait generation in animals' fast running. What we need to do is to build a mechanical structure that can emulate at least some of the basic properties of a musculoskeletal system exhibited in Animals' running.

While the minimal neuronal controller used for RunBot's walking is completely reflexive, we think the running controller has to be anticipative. For example, the foot-landing event is critical to the stability of biped running. For this reason, human adjusts the stiffness of his whole body and retracts his swing leg just before the foot-landing in his running (Seyfarth et al., 2003). In its essence, cyclic walking/running gaits are a result of coordinated oscillations of the involved joints. To predict the dynamics of an oscillating system, the minimally required is its frequency and phase. These anticipative information are represented in the Central Pattern Generator (CPG) that has been found existing in most of the animals' neural systems for locomotion control (Duysens and Van de Crommert, 1998). Therefore, the minimal neuronal controller for biped running should involve CPG, which can be modelled with neuron-oscillators.

**R19(1)**

↑-----

# Appendix A

## Dynamics equations of the robot

In the following we list the terms of the equation used in the simulation to build the Poincaré map function. For definitions of  $l_1, l_2, l_3, l_4, l_5, \phi, \theta_1, \theta_2, \psi$ , see figure 2.12,  $r$  is the radius of the curved foot,  $m_t$  is the mass of the trunk,  $m_h$  the mass of the thigh,  $m_k$  the mass of the shank with foot,  $g$  is the gravity acceleration.

$$l_1 = 0.11m$$

$$l_2 = 0.12m$$

$$l_3 = 0.08m$$

$$l_4 = 0.09m$$

$$l_5 = 0.022m$$

$$r = 0.12m$$

$$m_t = 0.12Kg$$

$$m_k = 0.03Kg$$

$$g = 9.8m/s^2$$

$$\begin{aligned} D_{11} = & -4m_k \cos(\phi) r^2 - 2m_k l_4 l_2 + 2m_k r l_4 + 2m_k (l_1^2 + l_2^2) \\ & + 2m_t l_1 l_2 - 2l_1 r - 2m_t r l_2 \\ & + 4m_k r \cos(\phi) l_2 - 2m_k r \cos(\phi) l_4 + 2m_k l_4^2 + 2m_t r \cos(\phi) l_2 \\ & + 2m_t l_2 l_5 \cos(\theta_1) - 2m_t r l_5 \cos(\theta_1) \\ & + 2m_t r \cos(\phi) l_1 + 2m_t r l_5 \cos(\theta_1 - \phi) + 2m_t l_1 l_5 \cos(\theta_1) \\ & - 2m_t \cos(\phi) r^2 + m_t l_5^2 \\ & - 4m_h \cos(\phi) r^2 - 2m_h l_1 l_3 - 2m_h l_2 l_3 \end{aligned}$$

$$\begin{aligned}
& + 2m_h r l_3 + 2m_k l_1 l_2 - 2m_k l_1 r - 4m_k r l_2 + 4m_k r^2 \\
& + 2m_k l_2^2 + 4m_h l_1 l_2 - 4m_h l_1 r - 4m_h r l_2 \\
& + 2m_h l_1^2 + 4m_h r^2 + 2m_h l_2^2 \\
& - 2m_h r l_3 \cos(-\theta_2 + \phi) \\
& - 2m_h l_1 l_3 \cos(-\theta_1 + \theta_2) - 2m_h l_2 l_3 \cos(-\theta_1 + \theta_2) \\
& + 2m_h r l_3 \cos(-\theta_1 + \theta_2) - 2m_k r l_4 \cos(-\theta_2 + \psi) \\
& + 2m_k r l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \\
& - 2m_k r l_1 \cos(-\theta_1 + \theta_2 + \phi) \\
& - 2m_k l_1 l_4 \cos(\psi) + 2m_k l_4 l_2 \cos(-\theta_1 + \theta_2 + \psi) \\
& + 2m_k l_1 l_4 \cos(-\theta_1 + \theta_2 + \psi) \\
& + 2m_k r l_1 \cos(-\theta_1 + \theta_2) + 4m_h r \cos(\phi) l_2 \\
& - 2m_h r \cos(\phi) l_3 + 4m_h r \cos(\phi) l_1 \\
& - 2m_k l_1^2 \cos(-\theta_1 + \theta_2) \\
& + 2m_h l_3^2 + m_t l_1^2 + 2m_t r^2
\end{aligned}$$

$$\begin{aligned}
D_{12} = & -m_t r l_5 \cos(\theta_1 - \phi) + m_t r l_5 \cos(\theta_1) \\
& - m_t (l_1 + l_2) l_5 \cos(\theta_1) \\
& - m_t m_h r l_3 \cos(-\theta_1 + \theta_2 + \phi) \\
& - m_h r l_3 \cos(-\theta_1 + \theta_2) + m_h l_2 l_3 \cos(-\theta_1 + \theta_2) \\
& + m_h l_1 l_3 \cos(-\theta_1 + \theta_2) - m_h l_3^2 \\
& + m_k r l_1 \cos(-\theta_1 + \theta_2 + \phi) - m_k r l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \\
& + m_k l_2 l_1 \cos(-\theta_1 + \theta_2) - m_k r l_1 \cos(-\theta_1 + \theta_2) \\
& + 2m_k l_1 l_4 \cos(\psi) \\
& + m_k l_1^2 \cos(-\theta_1 + \theta_2) \\
& + m_k l_1 l_4 \cos(-\theta_1 + \theta_2 + \psi)
\end{aligned}$$

$$\begin{aligned}
& - m_k l_4 l_2 \cos(-\theta_1 + \theta_2 + \psi) \\
& + m_k r l_4 \cos(-\theta_1 + \theta_2 + \psi) \\
& - m_k l_1^2 - m_k l_4^2
\end{aligned}$$

$$\begin{aligned}
D_{13} = & -m_h r l_3 \cos(-\theta_1 + \theta_2 + \phi) \\
& + m_h r l_3 \cos(-\theta_1 + \theta_2) - m_h l_2 l_3 \cos(-\theta_1 + \theta_2) \\
& - m_h l_1 l_3 \cos(-\theta_1 + \theta_2) + m_h l_3^2 + m_k l_1^2 - m_k r l_1 \cos(-\theta_1 + \theta_2) \\
& + m_k r l_4 \cos(-\theta_1 + \theta_2) + m_k l_2 l_1 \cos(-\theta_1 + \theta_2) \\
& + m_k r l_1 \cos(-\theta_1 + \theta_2) \\
& - 2m_k l_1 l_4 \cos(\psi) - m_k l_1^2 \cos(-\theta_1 + \theta_2) \\
& + m_k l_1 l_4 \cos(-\theta_1 + \theta_2 + \psi) \\
& + m_k l_4 l_2 \cos(-\theta_1 + \theta_2 + \psi) \\
& - m_k r l_4 \cos(-\theta_1 + \theta_2 + \psi) + m_k l_1^2 + m_k l_4^2
\end{aligned}$$

$$\begin{aligned}
D_{14} = & m_k r l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \\
& - m_k r l_4 \cos(-\theta_1 + \theta_2 + \psi) \\
& + m_k l_4 (l_1 + l_2) \cos(-\theta_1 + \theta_2 + \psi) \\
& - m_k l_1 l_4 \cos(\psi) + m_k l_4^2
\end{aligned}$$

$$D_{21} = D_{12}$$

$$D_{22} = m_t l_5^2 + m_h l_3^2 + m_k l_1^2 - 2m_k l_1 l_4 \cos(\psi) + m_k l_4^2$$

$$D_{23} = -m_h l_3^2 - m_k l_1^2 + 2m_k l_1 l_4 \cos(\psi) - m_k l_4^2$$

$$D_{24} = m_k l_1 l_4 \cos(\psi) - m_k l_4^2$$

$$D_{31} = D_{13}$$

$$D_{32} = D_{23}$$

$$D_{33} = m_h l_3^2 + m_k l_1^2 - 2l_1 l_4 \cos(\psi) + m_k l_4^2$$

$$D_{34} = -m_k l_1 l_4 \cos(\psi) + m_k l_4^2$$

$$D_{41} = D_{14}$$

$$D_{42} = D_{24}$$

$$D_{43} = D_{34}$$

$$D_{44} = m_k l_4^2$$

$$\begin{aligned} C_1 = & 2m_k \sin(\phi) \dot{\phi}^2 r^2 - 4m_k \sin(\phi) \dot{\phi}^2 l_2 r \\ & - 2m_t \sin(\phi) \dot{\phi}^2 (l_1 + l_2) r \\ & + 2m_t l_5 \sin(\theta_1 - \phi) \dot{\phi}^2 r \\ & - 2m_t l_5 \cos(\theta_1 - \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_1 \\ & + m_t l_5 \sin(\theta_1 - \phi) \dot{\theta}_1^2 r \\ & + 2m_t \sin(\phi) \dot{\phi}^2 r^2 + 4m_h \sin(\phi) \dot{\phi}^2 r^2 \\ & - 3m_t l_5 \sin(\theta_1 - \phi) \dot{\theta}_1 \dot{\phi} r \\ & + 2m_t l_5 \sin(\theta_1 - \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) r \\ & + 2m_h l_3 \sin(\phi) \dot{\phi}^2 r \\ & + m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1^2 r \\ & + 2m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} r \\ & + 2m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\theta}_2 \sin(\phi) l_2 \\ & + 2m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\theta}_2 \sin(\phi) l_1 \\ & - m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1^2 \sin(\phi) l_2 \\ & + 3m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} r \\ & - m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1^2 \cos(\phi) r \\ & + 2m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\theta}_2 \cos(\phi) r \end{aligned}$$

$$\begin{aligned}
& - 2m_k l_1^2 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\theta}_2 \cos(\phi) \\
& - 2m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\theta}_2 \cos(\phi) l_2 \\
& - 2m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) r \\
& + 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\psi} \sin(\phi) l_2 \\
& - 2m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_2 \\
& - 2m_k l_1^2 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) \\
& + m_k l_1^2 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1^2 \cos(\phi) \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi}^2 \sin(\phi) l_1 \\
& - 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\psi} \sin(\phi) r \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2^2 \sin(\phi) l_1 \\
& + 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\psi} \sin(\phi) l_1 \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2^2 \sin(\phi) l_2 \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi}^2 l_1 \sin(-\theta_1 + \theta_2 + \phi) \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2^2 \sin(\phi) r \\
& - 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\theta}_2 \sin(\phi) l_1 \\
& + 2l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\theta}_2 \sin(\phi) r \\
& + 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\psi} \sin(\phi) r \\
& + 2m_k l_1^2 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\theta}_2 \sin(\phi) \\
& - m_k l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1^2 \sin(\phi) l_2 \\
& - m_k l_1^2 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1^2 \sin(\phi) \\
& + m_k l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1^2 \sin(\phi) r \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1^2 \sin(\phi) l_2 \\
& + 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \sin(\phi) \dot{\phi} l_2 \\
& - 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\psi} \sin(\phi) l_1 \\
& - 2l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\theta}_2 \sin(\phi) r
\end{aligned}$$

$$\begin{aligned}
& - 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \\
& - 2m_k l_4 \cos(-\theta_2 + \psi + \phi) \dot{\psi} l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\phi} \\
& - 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \sin(\phi) \dot{\phi} r \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1^2 \sin(\phi) r \\
& - 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\psi} \sin(\phi) l_2 \\
& - 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\theta}_2 \sin(\phi) l_2 \\
& - m_k l_1^2 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2^2 \sin(\phi) \\
& + m_k l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2^2 \sin(\phi) r \\
& - m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2^2 \cos(\phi) r \\
& - 2m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) r \\
& - m_k l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2^2 \sin(\phi) l_2 \\
& + m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2^2 \cos(\phi) l_2 \\
& + m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2^2 \cos(\phi) r \\
& + m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} r \\
& + 2m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) l_2 \\
& + m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2^2 r \\
& - 2m_k l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} r \\
& + 2m_k l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_2 \\
& + 2m_k l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\theta}_2 \sin(\phi) l_2 \\
& - m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2^2 \sin(\phi) l_2 \\
& - m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2^2 \sin(\phi) l_1 \\
& - m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1^2 \cos(\phi) r \\
& - 2m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\theta}_2 \sin(\phi) r \\
& + m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2^2 \sin(\phi) r \\
& - 2m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_1
\end{aligned}$$

$$\begin{aligned}
& - m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1^2 \sin(\phi) l_1 \\
& + 2m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\theta}_2 \cos(\phi) r \\
& + 2m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) r \\
& - 2m_h l_3 \sin(-\theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_2 \\
& + m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1^2 \cos(\phi) l_2 \\
& + m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1^2 \cos(\phi) l_1 \\
& - 2m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\theta}_2 \cos(\phi) l_2 \\
& + 2m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_2 \\
& + 2m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_1 \\
& + 2m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) l_1 \\
& - 2m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} l_1 \\
& - 4m_h \sin(\phi) \dot{\phi}^2 l_2 r \\
& - 4m_h \sin(\phi) \dot{\phi}^2 l_1 r \\
& + 2m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\phi}^2 r \\
& - 2m_t l_5 \cos(\theta_1 - \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_2 \\
& + m_t l_5 \cos(\theta_1 - \phi) \dot{\theta}_1^2 \sin(\phi) l_2 \\
& + m_t l_5 \cos(\theta_1 - \phi) \dot{\theta}_1^2 \sin(\phi) l_1 \\
& - m_t l_5 \cos(\theta_1 - \phi) \dot{\theta}_1^2 \sin(\phi) r \\
& + m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1^2 \sin(\phi) r \\
& - m_t l_5 \sin(\theta_1 - \phi) \dot{\theta}_1^2 \cos(\phi) r \\
& - 2m_t l_5 \sin(\theta_1 - \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_2 \\
& + 2m_t l_5 \cos(\theta_1 - \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} r \\
& + m_t l_5 \sin(\theta_1 - \phi) \dot{\theta}_1^2 \cos(\phi) l_2 \\
& + m_t l_5 \sin(\theta_1 - \phi) \dot{\theta}_1^2 \cos(\phi) l_1 \\
& - 2m_t l_5 \sin(\theta_1 - \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_1
\end{aligned}$$



$$\begin{aligned}
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\theta}_2 r \\
& - 3m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \dot{\phi} r \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi}^2 \cos(\phi) l_2 \\
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\psi} r \\
& - 3m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\phi} r \\
& + 3m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\phi} r \\
& - 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \dot{\phi} \cos(\phi) l_2 \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2^2 r \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1^2 r \\
& - 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_2 \\
& - 2m_k \sin(\phi) \dot{\phi}^2 l_1 r \\
& + m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1^2 r \\
& + 2m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\phi}^2 r \\
& + 3m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} r \\
& - 2m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\theta}_2 r \\
& - 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) l_2 \\
& - 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) l_1 \\
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \dot{\phi} \cos(\phi) r \\
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) r \\
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\psi} \cos(\phi) l_1 \\
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\psi} l_1 \cos(-\theta_1 + \theta_2 + \phi) \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1^2 \cos(\phi) l_2 \\
& - 3m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} r \\
& + m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2^2 r \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1^2 \cos(\phi) l_1
\end{aligned}$$

$$\begin{aligned}
& - 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\psi} \cos(\phi) l_2 \\
& - 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\psi} \cos(\phi) l_1 \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2^2 \cos(\phi) r \\
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\psi} \cos(\phi) r \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2^2 \cos(\phi) l_2 \\
& + 4m_k \sin(\phi) \dot{\phi}^2 r^2 \\
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_1 \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2^2 \cos(\phi) l_1 \\
& + 2m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) l_2 \\
& + 2m_k l_1^2 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) \\
& - 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\phi}^2 r \\
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\theta}_2 \cos(\phi) l_2 \\
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\psi} \cos(\phi) l_2 \\
& - 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\theta}_2 \cos(\phi) r \\
& - 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\psi} l_1 \cos(-\theta_1 + \theta_2 + \phi) \\
& - 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) r \\
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_2 \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1^2 \cos(\phi) r \\
& + m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2^2 \cos(\phi) l_2 \\
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\theta}_2 \cos(\phi) l_1 \\
& - 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\psi} \cos(\phi) r \\
& + m_k l_1^2 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2^2 \cos(\phi) \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi}^2 r \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi}^2 \sin(\phi) r \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi}^2 \sin(\phi) l_2
\end{aligned}$$

$$\begin{aligned}
& + 2m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) r \\
& + m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1^2 \cos(\phi) l_2 \\
& - 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_1 \\
& + 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \sin(\phi) \dot{\phi} l_1 \\
& + 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \\
& - 2m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} l_2 \\
& - 2m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin \sin(\phi) \dot{\phi} r \\
& + 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} r \\
& + 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} l_2 \\
& + 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} l_1 \\
& - 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} r \\
& + 2m_k l_1^2 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi}^2 \cos(\phi) r \\
& - 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \dot{\phi} \cos(\phi) l_1 \\
& + 2m_k l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} r \\
& - 2m_k l_1^2 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} \\
& - 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\psi} r \\
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \dot{\phi} l_1 \cos(-\theta_1 + \theta_2 + \phi)
\end{aligned}$$

$$\begin{aligned}
C_2 = & -m_t l_5 \sin(\theta_1 - \phi) \dot{\phi}^2 r \\
& + m_t l_5 \cos(\theta_1 - \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_1 \\
& + m_t l_5 \sin(\theta_1 - \phi) \dot{\theta}_1 \dot{\phi} r \\
& - m_t l_5 \sin(\theta_1 - \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) r \\
& + m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) r \\
& + m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_2
\end{aligned}$$

$$\begin{aligned}
& + m_k l_1^2 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi}^2 l_1 \sin(-\theta_1 + \theta_2 + \phi) \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \sin(\phi) \dot{\phi} l_2 \\
& + 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \\
& + 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\phi} \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \sin(\phi) \dot{\phi} r \\
& + m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) r \\
& - m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) l_2 \\
& - m_k l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_2 \\
& + m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_1 \\
& - m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) r \\
& + m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_2 \\
& - m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_2 \\
& - m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_1 \\
& - m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) l_1 \\
& - m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\phi}^2 r \\
& + m_t l_5 \cos(\theta_1 - \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_2 \\
& + m_t l_5 \sin(\theta_1 - \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_2 \\
& - m_t l_5 \cos(\theta_1 - \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} r \\
& + m_t l_5 \sin(\theta_1 - \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_1 \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \dot{\phi} r \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\phi} r \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\phi} r \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \dot{\phi} \cos(\phi) l_2 \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_2
\end{aligned}$$

$$\begin{aligned}
& - m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\phi}^2 r \\
& - m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} r \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi}^2 l_1 \cos(-\theta_1 + \theta_2 + \phi) \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \dot{\phi} \cos(\phi) r \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) r \\
& - 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\psi} l_1 \cos(-\theta_1 + \theta_2 + \phi) \\
& + m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} r \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_1 \\
& - m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) l_2 \\
& - m_k l_1^2 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\phi}^2 r \\
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\psi} l_1 \cos(-\theta_1 + \theta_2 + \phi) \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\phi} \cos(\phi) r \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_2 \\
& - m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) r \\
& + m_k l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} l_2 \\
& + m_k l_4 \cos(-\theta_2 + \psi + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_1 \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \sin(\phi) \dot{\phi} l_1 \\
& - 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \\
& + m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} l_2 \\
& + m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} r \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} r \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\phi} l_1 \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} r \\
& - m_k l_1^2 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi}
\end{aligned}$$

$$\begin{aligned}
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \dot{\phi} \cos(\phi) l_1 \\
& - m_k l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} r \\
& + m_k l_1^2 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} \\
& - 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \dot{\phi} l_1 \cos(-\theta_1 + \theta_2 + \phi)
\end{aligned}$$

$$\begin{aligned}
C_3 = & m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} r \\
& + m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} r \\
& - m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\phi} \cos(\phi) r \\
& - m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_2 \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi}^2 l_1 \sin(-\theta_1 + \theta_2 + \phi) \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \sin(\phi) \dot{\phi} r \\
& + m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) l_2 \\
& - m_k l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} r \\
& + m_k l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_2 \\
& - m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_1 \\
& + m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) r \\
& - m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_2 \\
& + m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_2 \\
& + m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_1 \\
& - m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} l_1 \\
& + m_h l_3 \sin(-\theta_1 + \theta_2 + \phi) \dot{\phi}^2 r \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \dot{\phi} r \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\phi} r \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\phi} r \\
& - m_k l_4 \sin(-\theta_2 + \psi + \phi) \dot{\psi} \dot{\phi} \cos(\phi) l_2
\end{aligned}$$

$$\begin{aligned}
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_2 \\
& + m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\phi}^2 r \\
& + m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} r \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) l_2 \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) l_1 \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \dot{\phi} \cos(\phi) r \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) r \\
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \dot{\psi} l_1 \cos(-\theta_1 + \theta_2 + \phi) \\
& - m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} r \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_1 \\
& + m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) l_2 \\
& + m_k l_1^2 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \dot{\phi} \cos(\phi) \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\phi}^2 r \\
& - 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\psi} l_1 \cos(-\theta_1 + \theta_2 + \phi) \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\phi} \cos(\phi) r \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) l_2 \\
& + m_k l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\phi} \cos(\phi) r \\
& - m_k l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} l_2 \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} l_1 \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \sin(\phi) \dot{\phi} l_1 \\
& + 2m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \\
& - m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} l_2 \\
& - m_h l_3 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} r \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} r \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} l_2
\end{aligned}$$

$$\begin{aligned}
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} l_1 \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} r \\
& + m_k l_1^2 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \sin(\phi) \dot{\phi} \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \dot{\phi} \cos(\phi) l_1 \\
& + m_k l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} r \\
& - m_k l_1^2 \cos(-\theta_1 + \theta_2 + \phi) \dot{\theta}_2 \sin(\phi) \dot{\phi} \\
& + 2m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \dot{\psi} \dot{\phi} l_1 \cos(-\theta_1 + \theta_2 + \phi)
\end{aligned}$$

$$\begin{aligned}
C_4 = & -m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) r \dot{\phi}^2 \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \sin(\phi) \dot{\phi} l_1 \dot{\psi} \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \sin(\phi) \dot{\phi} l_2 \dot{\psi} \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \sin(\phi) \dot{\phi} r \dot{\psi} \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\phi} \dot{\psi} \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \sin(\phi) \dot{\phi} r \dot{\theta}_1 \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \cos(\phi) \dot{\phi} r \dot{\psi} \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) l_1 \sin(-\theta_1 + \theta_2 + \phi) \dot{\theta}_1 \dot{\psi} \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \cos(\phi) \dot{\phi} l_1 \dot{\psi} \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) l_1 \cos(-\theta_1 + \theta_2 + \phi) \dot{\phi} \dot{\psi} \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \sin(\phi) \dot{\phi} l_2 \dot{\theta}_1 \\
& - m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \sin(\phi) \dot{\phi} r \dot{\theta}_2 \\
& + m_k l_4 \cos(-\theta_1 + \theta_2 + \psi + \phi) \sin(\phi) \dot{\phi} l_1 \dot{\theta}_2 \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \cos(\phi) \dot{\phi} l_2 \dot{\theta}_2 \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \cos(\phi) \dot{\phi} l_2 \dot{\theta}_1 \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \cos(\phi) \dot{\phi} r \dot{\theta}_2 \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \cos(\phi) \dot{\phi} r \dot{\theta}_1
\end{aligned}$$



$$\begin{aligned}
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \cos(\phi) \dot{\phi} l_1 \dot{\theta}_2 \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \cos(\phi) \dot{\phi} l_2 \dot{\psi} \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) \cos(\phi) \dot{\phi} l_1 \dot{\theta}_1 \\
& + m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) r \dot{\phi} \dot{\theta}_1 \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) r \dot{\phi} \dot{\psi} \\
& - m_k l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi) r \dot{\phi} \dot{\theta}_2
\end{aligned}$$

$$\begin{aligned}
G_1 = & m_t g \sin(\phi) r - m_t g \sin(\phi) l_2 \\
& - m_t g \sin(\phi) l_1 + m_t g l_5 \sin(\theta_1 - \phi) \\
& + 2m_h g \sin(\phi) r - 2m_h g \sin(\phi) l_2 \\
& - 2m_h g \sin(\phi) l_1 \\
& + m_h g l_3 \sin(\phi) \\
& + m_h g l_3 \sin(-\theta_1 + \theta_2 + \phi) \\
& + 2m_k g \sin(\phi) r + m_k g \sin(\phi) l_4 \\
& - 2m_k g \sin(\phi) l_2 \\
& - m_k g \sin(\phi) l_1 + m_k g l_1 \sin(-\theta_1 + \theta_2 + \phi) \\
& - m_k g l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi)
\end{aligned}$$

$$\begin{aligned}
G_2 = & -m_t g l_5 \sin(\theta_1 - \phi) \\
& - m_h g l_3 \sin(-\theta_1 + \theta_2 + \phi) \\
& - m_k g l_1 \sin(-\theta_1 + \theta_2 + \phi) \\
& + m_k g l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi)
\end{aligned}$$

$$\begin{aligned}
G_3 = & m_h g l_3 \sin(-\theta_1 + \theta_2 + \phi) \\
& + m_k g l_1 \sin(-\theta_1 + \theta_2 + \phi) \\
& - m_k g l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi)
\end{aligned}$$

$$G_4 = -m_k g l_4 \sin(-\theta_1 + \theta_2 + \psi + \phi)$$

# Appendix B

## Simulation analysis

The dynamics of RunBot are modelled as shown in figure B.1, in which,  $L_1 = 0.11m$ ,  $L_2 = 0.12m$ ,  $L_3 = 0.09m$ ,  $L_4 = 0.08m$ . For detailed dynamics equations of the robot and Poincaré map method, see chapter 2.

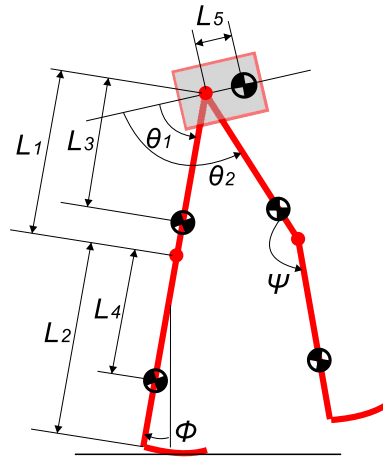


FIGURE B.1: Model of the dynamics of our robot. Sizes and masses are the same as those of the real robot.

The values of the neuron parameters in the simulation are chosen to be identical to those in the real robot. Moreover, to simplify the problem, we also fix the gain of the motor neurons of the hip joints, i.e.,  $G_{M,h} = 2.5$  (at the middle of the stable area in figure 3.6). Thus, we only need to adjust the value of  $\Theta_{ES,h}$  to change the properties of the gaits.

To see how the location of the mass centre ( $L_5$  in figure B.1) of the trunk affects the stability and the speed of the gaits, we also change the value of  $L_5$  in the simulation. With each set of  $L_5$  and  $\Theta_{ES,h}$ , we use a multi-dimensional Newton-Raphson method

solving equation éPoincaréo find the fixed point (Garcia, 1999). Then we compute the Jacobian matrix  $J$  of the fixed point using the approach described in Garcia (1999), and evaluate the stability of the fixed point according to its eigenvalues. The simulation results are shown in figure B.2.

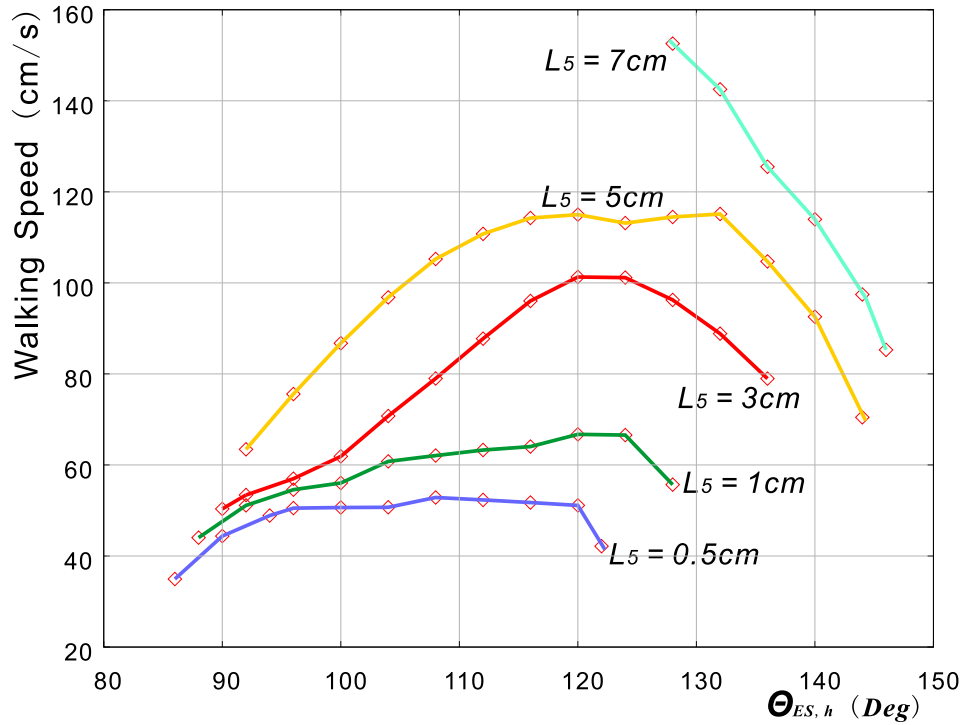


FIGURE B.2: Change of walking speed while  $G_{M,h}$  is fixed at 2.5 and  $\Theta_{ES,h}$  is changed in its stable range. Each curve corresponds to a different location of the mass center of the trunk (see figure B.1), i.e.,  $L_5 = 0.5cm, 1cm, 3cm, 5cm, 7cm$ .

Because some details of the robot dynamics such as uncertainties of the ground contact, nonlinear frictions in the joints and the inevitable noise and lag of the sensors cannot be modelled precisely, the results of the simulation suggest a larger stable range than the real experiments. For example, in the real robot, the mass centre of the trunk is located about 3 cm forward. With  $G_{M,h} = 2.5$ , stable gaits can appear when  $\Theta_{ES,h}$  is in the range of  $95^\circ - 122^\circ$  (see figure 3.6). But in the simulation, the stable range of  $\Theta_{ES,h}$  is somewhat bigger,  $90^\circ - 136^\circ$  (see the curve indicated with  $L_5 = 3cm$  in figure B.2). However, the simulation results have shown that the location of the mass center of the trunk does have a drastic influence on the stability and the speed of the gaits:

1. A small value of  $L_5$  (see figure B.1 and B.2) is helpful to the stability of the gaits at slow walking speeds.
2. If the mass center of the trunk is located appropriately forward (e.g.,  $L_5 = 3cm, 5cm$

in figure B.2), stable range and walking speed can both be improved.

3. But, if the mass center is located too far forward (e.g.,  $L_5 = 7cm$  in figure B.2), the stable range for the neuron parameters will become quite small, though the walking speed can be very high.

# Bibliography

- Akazawa, K., Aldridge, J., Steeves, J., and Stein, R. (1982). Modulation of stretch reflexes during locomotion in the mesencephalic cat. *Journal of Physiology*, 329:553–67.
- Alexander, R. and Jayes, A. (1983). A dynamic similarity hypothesis for the gaits of quadrupedal mammals. *Journal of Zoology*, 201:135–152.
- Andrew, M. and Rymer, W. (1997). Role of intrinsic muscle properties in producing smooth movements. *IEEE Transactions on Biomedical Engineering*, 44:165–176.
- Azevedo, C., Andre, N., and Arias, S. (2004). Bipedal walking: from gait design to experimental analysis. *Mechatronics*, 14:639–665.
- Beer, R. and Chiel, H. (1992). A distributed neural network for hexapod robot locomotion. *Neural Computation*, 4:356–365.
- Beer, R., Quinn, R., Chiel, H., and Ritzmann, R. (1997). Biologically inspired approaches to robotics. *Communications of the ACM*, 40(3):30–38.
- Boone, G. and Hodgins, J. (1997). Slipping and tripping reflexes for bipedal robots. *Autonomous Robots*, 4(3):259–271.
- Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23.
- Brown, I. and Loeb, G. (1999). *Biomechanics and Neural Control of Movement*, chapter 3. Springer-Verlag, New York.
- Cham, J., Bailey, S., and Cutkosky, M. (2000). Robust dynamic locomotion through feedforward-preflex interaction. In *Proceedings of ASME annual International Mechanical Engineering Congress*, pages 231–237, Orlando, Florida.
- Chestnutt, J., Lau, M., Cheung, G., Kuffner, J., Hodgins, J., and Kanade, T. (2005). Footstep Planning for the Honda ASIMO Humanoid. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 629–634, New York.

- Chevallereau, C., Abba, G., Aoustin, Y., Plestan, F., Westervelt, E., Canudas-de Wit, C., and Grizzle, J. (2003). Rabbit: A testbed for advanced control theory. *IEEE Control Systems Magazine*, 23:57–79.
- Chiel, H. and Beer, R. (1997). The brain has a body: adaptive behavior emerges from interactions of nervous system, body, and environment. *Trends in Neuroscience*, 20:553–557.
- Cohen, A. H. and Boothe, D. L. (1999). Sensorimotor interactions during locomotion: principles derived from biological systems. *Autonomous Robots*, 7:239–245.
- Collins, S., Ruina, A., Tedrake, R., and Wisse, M. (2005). Efficient bipedal robots based on passive-dynamic walkers. *Science*, 37:1082–1085.
- Cruse, H., Kindermann, T., and Schumm, M. (1998). Walknet—a biologically inspired network to control six-legged walking. *Neural Networks*, 11(7):1435–1447.
- Cruse, H. and Warnecke, H. (1992). Coordination of the legs of a slow-walking cat. *Experimental Brain Research*, 89:147–156.
- Delcomyn, F. (1980). Neural basis of rhythmic behavior in animals. *Science*, 210:492–498.
- Dickinson, M. and Farley, C. (2000). How Animals Move: An Integrative View. *Science*, 288(5463):100–105.
- Dietz, V., Gollhofer, A., Kleiber, M., and Trippel, M. (1992). Regulation of bipedal stance: dependency on load receptors. *Experimental Brain Research*, 89(1):229–231.
- Donelan, J. and Kram, R. (1997). The effect of reduced gravity on the kinematics of human walking: a test of the dynamic similarity hypothesis for locomotion. *Journal of Experimental Biology*, 200:3193–3201.
- Duysens, J. and Van de Crommert, H. (1998). Neural control of locomotion. Part 1: the central pattern generator from cats to humans. *Gait Posture*, 7:131–141.
- Duysens, J., Van de Crommert, H., Smits-Engelsman, B., and Van der Helm, F. (2002). A walking robot called human: lessons to be learned from neural control of locomotion. *Journal of Biomechanics*, 35:447–453.
- Ferrell, C. (1995). A comparison of three insect-inspired locomotion controllers. *Robotics and Autonomous Systems*, 16:135–159.
- Forsberg, H. (1979). Stumbling corrective reaction: a phase-dependent compensatory reaction during locomotion. *Journal of Neurophysiology*, 42(4):936–953.

- Fukuoka, Y., Kimura, H., and Cohen, A. (2003). Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts. *International Journal of Robotics Research*, 22:187–202.
- Full, R. J. and Tu, M. S. (1990). Mechanics of six-legged runners. *Journal of Experimental Biology*, 148:129–146.
- Funabashi, H., Takeda, Y., S., I., and Higuchi, M. (2001). Disturbance compensating control of a biped walking machine based on reflex motions. *JSME International Journal Series C - Mechanical Systems, Machine Elements and Manufacturing*, 44:724–730.
- Gallagher, J., Beer, R., Espenschied, K., and Quinn, R. (1996). Application of evolved locomotion controllers to a hexapod robot. *Robotics and Autonomous Systems*, 19:95–103.
- Garcia, M. (1999). *Stability, scaling, and chaos in passive-dynamic gait models*. PhD thesis, Cornell University.
- Geng, T., Porr, B., and Wörgötter, F. (2006). A reflexive neural network for dynamic biped walking control. *Neural Computation*, 18:1156–1196.
- Gerritsen, K. (1998). Intrinsic muscle properties facilitate locomotor control - a computer simulation study. *Motor Control*, 2:206–220.
- Grillner, S. and Wallen, P. (1982). On peripheral control mechanisms acting on the central pattern generators for swimming in the dogfish. *Experimental Biology*, 98:1–22.
- Hardt, M. and von Stryk, O. (2002). The role of motion dynamics in the design, control and stability of bipedal and quadrupedal robots. In *Proceedings of RoboCup International Symposium*, pages 206–223, Bern.
- Hirai, K. (1997). Current and future perspective of honda humanoid robot. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 500–509, Nagoya, Japan. IEEE.
- Hurmuzlu, Y. (1993). Dynamics of bipedal gait; part ii: Stability analysis of a planar five-link biped. *ASME Journal of Applied Mechanics*, 60(2):337–343.
- Iida, F. and Pfeifer, R. (2004). Self-stabilization and behavioral diversity of embodied adaptive locomotion. *Lecture notes in artificial intelligence*, 3139:119–129.
- Ijspeert, A. J. (2001). *The Hand Book of Brain Theroy and Neural Networks*, chapter 3, Locomotion, Vertebrate. MIT Press.

- Inoue, H. and Tachi, S. (2000). Hrp: Humanoid robotics project of miti. In *Proceedings of the First International Conference on Humanoid Robots*, pages 421–426, Tokyo. IEEE.
- Kajita, S. and Kobayashi, A. (1987). Dynamic walk control of a biped robot with potential energy conserving orbit. *Journal of Japanese Society of Instrument and Control Engineers*, 23:281–287.
- Klavins, E., Komsuoglu, H., Full, R., and Koditschek, D. (2002). *Neurotechnology for Biomimetic Robots*, chapter 2, The Role of Reflexes Versus Central Pattern Generators in Dynamical Legged Locomotion. MIT Press.
- Kohl, N. and Stone, P. (2004). Policy gradient reinforcement learning for fast quadrupedal locomotion. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2619–2624, Lyon.
- Kuroki, Y., Ishida, T., Yamaguchi, J., Fujita, M., and Doi, T. (2001). A small biped entertainment robot. In *Proceedings of International Conference on Humanoid Robots*, pages 181–186, Sydney. IEEE.
- Lewis, M. (2001). Certain principles of biomorphic robots. *Autonomous Robots*, 11:221–226.
- Lewis, M., Etienne-Cummings, R., Hartmann, M., Xu, Z., and Cohen, A. (2003). An in silico central pattern generator: Silicon oscillator, coupling, entrainment, and physical computation. *Biological Cybernetics*, 88:137–151.
- Linde, V. d. R. Q. V. (1998). Active leg compliance for passive walking. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2339–2345, Orlando, Florida.
- Loffler, K., Gienger, M., and Pfeiffer, F. (2003). Sensors and Control Concept of Walking “Johnnie”. *The International Journal of Robotics Research*, 22(3):229–239.
- Miyakoshi, S. and Cheng, G. (2002). Ballistic walking by compass-like biped walker - exploiting physical dynamics in achieving human-like walking. In *Proceedings of 5th International Conference on Climbing and Walking Robots*, pages 445–452, Munich.
- Miyazaki, F. and Arimoto, S. (1987). A control theoretic study on dynamical biped locomotion. *ASME Transactions of Dynamic Systems, Measurement, and Control*, 102:233–239.
- Morasso, P., Bottaro, A., Casadio, M., and Sanguineti, V. (2005). Preflexes and internal models in biomimetic robot systems. *Cognitive Processing*, 6(1):25–36.



- Moritz, C. and Farley, C. (2004). Passive dynamics change leg mechanics for an unexpected surface during human hopping. *Journal of Applied Physiology*, 97:1313–1322.
- Nishiwaki, K., Sugihara, T., Kagami, S., Kanehiro, F., Inaba, M., and Inoue, H. (2000). Design and development of research platform for perception-action integration in humanoid robot. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 1559–1564, Taipei. IEEE.
- Park, J. and Kim, K. (1998). Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 174–179, Edinburgh.
- Porr, B. and Wörgötter, F. (2005). Inside embodiment—what means embodiment for radical constructivists? *Kybernetes*, 34:105–117.
- Poulakakis, I., P. E. and Buehler, M. (2003). On the stable passive dynamics of quadrupedal running. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1368–1373, Shanghai.
- Pratt, J. (2000). *Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots*. PhD thesis, Massachusetts Institute of Technology.
- Printer, M. and Dimitrijevic, M. (1999). Gait after spinal cord injury and the central pattern generator for locomotion. *Spinal Cord*, 37:531–537.
- Purves, D. (2004). *Neuroscience*. MIT Press.
- Raibert, H. and Hodgins, J. (1993). *Biological Neural Networks in Invertebrate Neuroethology and Robotics*, chapter 3, Legged robots, pages 319–354. Academic Press, Boston.
- Sano, A. and Furusho, J. (1990). 3d dynamic walking of biped robot by controlling the angular momentum. *Journal of Japanese Society of Instrument and Control Engineers*, 26:459–466.
- Seyfarth, A., Geyer, H., and Herr, H. (2003). Swing-leg retraction: a simple control model for stable running. *Journal of Experimental Biology*, 206(15):2547–2555.
- Seyfarth, A., G. H. G. M. and Blickhan, R. (2002). A movement criterion for running. *Journal of Biomechanics*, 35:649–655.
- Taga, G. (1995). A model of the neuro-musculo-skeletal systems for human locomotion. *Biological Cybernetics*, 73:97–111.

- Vaughan, C. L. (2003). Theories of bipedal walking: an odyssey. *Journal of Biomechanics*, 36:513–523.
- Vaughan, C. L. and O'Malley, M. J. (2005). Froude and the contribution of naval architecture to our understanding of bipedal locomotion. *Gait and Posture*, 21:350–362.
- Vukobratovic, M., Borovac, B., Surla, D., and Stokic, D. (1990). *Biped locomotion: dynamics, stability, control and application*. Springer-Verlag.
- Wadden, T. and Ekeberg, O. (1998). A neuro-mechanical model of legged locomotion: Single leg control. *Biological Cybernetics*, 79:161–173.
- Wagner, H. and Blickhan, R. (2003). Stabilizing function of antagonistic neuromusculoskeletal systems: an analytical investigation. *Biological Cybernetics*, 89:71–79.
- Wisse, M. and van Frankenhuyzen, J. (2003). 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*, Kyoto, Japan.
- Yamaguchi, J., Soga, E., Inoue, S., and Takanishi, A. (1999). Development of a bipedal humanoid robot: control method of whole body cooperative dynamic biped walking. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 368–374, Boston.
- Yang, J., Stephens, M., and Vishram, R. (1998). Infant stepping: A method to study the sensory control of human walking. *Journal of Physiology*, 507:927–937.
- Zehr, E. P. and Stein, R. B. (1999). What functions do reflexes serve during human locomotion? *Progress in Neurobiology*, 58:15–205.
- Zheng, Y. and Shen, J. (1990). Gait synthesis for the SD-2 biped robot to climb sloping surface. *IEEE Transactions on Robotics and Automation*, 6(1):86–96.