Technische Universität München – Germany
The Institute for Cognitive Systems

und

Université de Versailles Saint-Quentin-en-Yvelines – France
Laboratoire d'Ingénierie des Systèmes de Versailles

Doppelpromotion / Joint Doctorate

Success-Failure Learning for Humanoid: study on bipedal walking

John Nassour

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. Werner Hemmert
Prüfer der Dissertation:	Univ.-Prof. Dr. Gordon Cheng, Ph.D. Prof. Dr. Fethi Benouezdou, Université de Versailles / Frankreich Univ.-Prof. Dr.-Ing Tamim Asfour, Karlsruher Institut für Technologie

Die Dissertation wurde am 11.09.2013 bie der Technische Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 19.12.2012 angenommer.

Acknowledgements

I would like to thank all people who have supported and inspired me during my doctoral study. I would like to express my sincere gratitude to all of them.

Foremost, I would like to thank my advisors Prof. Gordon Cheng and Prof. Fethi Benouezdou for providing me the opportunity to complete my PhD thesis at the Technical University of Munich and the University of Versailles. I especially want to thank my advisor, Prof. Cheng, for his unlimited support, patience, motivation and guidance. He has been actively interested in my work and has always been available for advising me. I am very glad to have worked with him for many years. I hope that one day I would become as good advisor to my students as Gordon has been to me.

My deepest gratitude is to my French advisor, Prof. Fethi Benouezdou, for the academic support and the facilities provided to reach the goal of my research. He has been very encouraging and supportive. I would like to express my gratitude to him. My French co-advisor, Dr. Patrick Hénaff, has been always there to listen and provide recommendation. I am thankful to him for the discussions that helped to sort out the technical details of my work. I am deeply grateful to Dr. Vincent Hugel for helping me during the validation phase of my work on the NAO humanoid robot, I appreciate his support. Besides my advisors, I would like to thank the reviewer of my thesis, Prof. Tamim Asfour, for accepting reviewing my thesis, and for his encouragement and insightful comments. I would like to thank all administration staff at the Technical University of Munich and the University of Versailles who completed successfully the joint-PhD agreement.

The generous support from the German Research Foundation DFG, and the support of the Center of Franco-Bavarian University Cooperation CCUFB are greatly appreciated. Furthermore, I would like to acknowledge the French National Research Agency ANR as my research was supported in part by them. I am also grateful to the staff of the Institute for Cognitive Systems ICS at the Technical University of Munich, for their various forms of support– Anja Delanoff, Ilona Nar-Witte, Brigitte Rosenlehner and all ICS researchers. I am thankful to the staff of the LISV laboratory in France, Dominique Maillet and Tuyet Touchais, and the PhD students for encouraging me during my stay in France.

Many friends, Samer Alfayad, Elmira Amrollah, and Abbass Al-Ali have supported me through these difficult years. Their support and care helped me overcome setbacks and stay focused on my thesis. I greatly value their friendship and I deeply appreciate their trust in me. I am also grateful to the German family that hosts my stay in Munich in very good conditions.

None of this would have been possible without the love and patience of my family. My immediate family, to whom this dissertation is dedicated to, has been a constant source of love, concern, support and strength all these years. I would like to express my heartfelt gratitude to my mother, father, sister and brother.

I also dedicate this Ph.D. thesis to my powerful source of inspiration and energy: my joyful son, Khalil, and my affectionate daughter, Cham. They have supported me during all these years. I love you a lot.

Last but not least, thanks to God for my life through all the tests in the past five years. You have made my life more bountiful. May your name be exalted, honored, and glorified.

John

Abstract

We develop our skills through continuous interaction with the world, by learning from our experience of successes as well as failures. Neurological understanding of the mechanisms involved in this mental process are beginning to emerge to a level where they can be validated on robots functioning in the real world.

The Anterior Cingulate Cortex (ACC) has shown to contribute to cognitive control by modulating the error-related signals for both positive and negative past experiences, thus acting as an *early warning system* (EWS). The notion of *Vigilance* affects the manner we learn, in other word, the way we make decisions. The Orbitofrontal Cortex (OFC) plays a role in our learning process by representing the effective value of reinforcements – coding of rewards, thus, regulating decision-making and expectation. These neural mechanisms play an underpinning role in cognitive development and learning. Furthermore, it has been shown that robust and complex motor patterns are generated from the spinal cord via a *central pattern generator* (CPG).

This thesis presents computational models of these neural mechanisms, implemented on a robot that can acquire and learn from its experiences. A framework for success-failure learning based on the studies of the ACC is presented. Validation of this work was first conducted on a 2D planar robot simulation. Based on the OFC, this success-failure learning framework was extended to support the coding of reward which enhances robot's performance. A low-level controller based on studies of CPGs was developed to support the motor learning. This new CPG was built by extending mathematical models of well-established models of CPGs. The proposed multi-layered multi-pattern CPG model (MLMP-CPG) provides a diverse pattern generator for the production of motor patterns for learning. Bringing these mechanisms together, we validated the success-failures learning framework with the support of the extended central pattern generator on a humanoid robot NAO. The goal was to learn walking under varying conditions. The obtained results showed that the robot was able to adapt to different slopes as well as to deal with disturbances.

Contents

List of Figures

Chapter 1

Introduction and Motivation

Contents

For centuries humans have been dreaming of replacing themselves with agents in order to perform tasks that are ill-suited for humans. From the robot "Knight" designed by Leonardo da Vinci (1495) to today's robots, immense developmental milestones have been made in term of materials and actuators as well as in terms of control and information processing. While vast advances have been made in analyzing human behaviors: from the "theory of reflex movement" (1664) proposed by Descartes that introduces *a reflex circuit* of human behavior that was triggered by flame, to today's sophisticated models of the human brain that have in a large part been accompanied by new technologies for measuring brain activity. Cognitive neuroscience contributes a large part to our self-understanding. Nevertheless, we still slightly appreciate the myriad of internal processes that are at the heart of our intelligence. Perhaps we will never completely understand the human brain, but we stand to pursue this understanding.

1.1 Perspectives

Taking a global perspective, we take the view that building humanoid cognitive systems based the understanding of how human biological cognitive system works brings together common interests across several disciplines. It provides information for the engineering of more robust artificial systems that are based on biological systems. Furthermore, it holds promises of making robots more interactive, useful, and even more suitable for our society, as these systems embrace similar human-like judgment in addition to human knowledge [Cheng et al., 2007].

By channeling ideas from different disciplines regarding an ultimate goal of understanding the brain, humanoid robotics combines mechanical engineering, electronic engineering, computer sciences, artificial intelligence, psychology, biomechanics, and neuroscience. The combination of neuroscience research and robotics research is increasingly used to solve open problems in both fields. One big challenge of today's robotic research consists of endowing robots with the capability to *learn* new skills by self-exploration while adapting to environmental changes. Furthermore, they must be able to improve their skills and behaviors to meet varying demands (e.g.: performing tasks under different conditions). Neuroscience provides useful descriptions about information processing in the nervous system of humans and animals. NeuroRobotics is the combined field of neuroscience and robotics. One key aspect of this field is to exploit brain-modeled algorithms to improve the learning capabilities of robots. Unlike classical approaches, neurorobotics carefully accounts for a similar path to its biological counterpart, thus providing a new and proven way to allow robots to improve their skills and behaviors.

Inevitably, skills and behaviors need to be possessed by robots to function and be helpful in our world. In this way, they should be able to react and adapt existing capabilities with environmental changes. This is why learning on the fly by sequencing existing behaviors is indispensable for autonomous robots. Learning allows humanoid robots to acquire new skills for tasks that they have never done before, by self-exploration or by observation. In goal directed learning, humanoid robots should be told what to do, rather than how to do it. The ultimate goal is to transfer the human mechanisms of learning, perception, acquisition, and coordination into humanoid robots. We focus in this thesis on a learning framework. In this thesis, learning is oriented towards physical tasks learning for humanoid robots. Our goal is to propose a learning model based on human learning from experience in order to perform walking task in humanoid robot.

1.2 Problem Definition

Robots' ability to acquire new skills can be separated into two key aspects: 1) the ability to represent skills in memory with self-exploration and evaluation mechanisms; and 2) the ability to generate the related motion patterns.

An evaluation phase is indispensable to explore and to acquire new skills while learning physical tasks. For instance, how well was the task achieved in the current trial? The evaluation phase associates improving the experience of the robot when performing a new trial to learn the task.

Most classical learning approaches require prior information about the environment before starting the learning process (e.g. reinforcement learning models that are widely used for learning tasks require prior information about the reward signals before starting to learn). Besides learning techniques, different algorithms have been proposed to solve skill acquisition problems. Evolution methods like Genetic Algorithms are widely used for optimization in robotics. However, these off-line techniques require an environmental model that is assumed to match environment properties. Policy Gradient Method, which is the most prominent reinforcement learning technique, does not ensure convergence to the global optimum, like many other methods that only guarantee a convergence to local optima.

Beside the problem of high-level control for skill acquisition, the robot must be able to generate related motion patterns. Most Central Pattern Generator (CPG) models used in robot locomotion cannot explain the complex behaviors in human and animal locomotion. They are based on interconnected neurons that show oscillatory patterns. These models are unable to show complex behaviors that can mix both periodic and non-periodic patterns.

In order to overcome these issues in robot and human skill acquisition, an efficient controller for learning and achieving various tasks by robots must consider two essential elements: i) an efficient and adaptive learning mechanism (high-level control); supported by ii) a diverse pattern generator (low-level control).

1.3 Our Contribution

The purpose of this thesis is to validate neural models on robots in order to provide them with better mechanisms for learning tasks, and better neural models for pattern generation.

The presented work includes three essential parts. First, based on neurobiological studies devoted to observe the Anterior Cingulated Cortex (ACC) activation in human brain prior to and during mistakes, a learning mechanism that accounts for mistakes as well as for success is presented. ACC is involved not only in external error detection, but also in internal error prediction. ACC works as an early warning system (EWS) that adjusts the behavior in accordance with prior experiences, especially to avoid negative consequences. The proposed model is called “Success-Failure learning”. It learns good experiences as well as bad ones. Based on psychological studies in risk taking behavior, the concept of vigilance is introduced to manage the risk tendency in Success-Failure learning that allows the agent to switch between exploration and exploitation.

Second, the success-failure learning is improved with a model of the Orbitofrontal cortex (OFC), based on its role of adaptive reward coding in the human brain. This allows to qualify the good experience according to each trial outcome. Before being introduced into the learning process, the efficiency of trials is coded adaptively depending on the available outcomes; the coded rewards will be altered with a newly elevated outcome. Then, it is represented as a reward signal that plays an important role in the learning cycle, providing it with the qualitative metrics in formulating successes. We refer to this technique as *Qualitative Adaptive Reward Learning* (QARL). Unlike most reinforcement learning techniques, the key feature of this approach is that it does not require any prior information for gaining efficiency.

Third, an efficient low-level controller, called multi-layered multi-pattern CPG model (MLMP-CPG), is introduced to generate diverse patterns. This controller is based on an extended mathematical model of the Central Pattern Generator (CPG) in the spinal cord. This model is supported by two neuroscience studies: 1) the first study consists of a two-levels CPG, in which Pattern Formation neurons (PF) and Rhythm Generation neurons (RG) are produced at different layers. This separation between these layers explains the observation of animals’ locomotion behaviors, where the rhythm of pattern is independent from the shape (e.g. non-resetting deletion). 2) The second study draws on a neural model that can generate different activities, including oscillation. This neural model is used within Rhythm Generation neurons of the CPG, thus enabling it to produce different motion patterns for periodic and discrete motion. In order to allow high-level control during goal-directed actions, a technique for dimensionality reduction was proposed to establish a space of patterns. Complex locomotion behaviors can then be presented in this pattern space, allowing to switch between behaviors.

To validate these mechanisms, both low and high-level controllers are implemented, first on a simulated planar biped, and then on a real NAO humanoid robot. The robot learned to walk under varying conditions, on flat as well as sloped terrains. The effectiveness of the CPG model is shown by its ability to switch from oscillatory to different motion patterns in order to react against external perturbation. Due to the diversity of the CPG patterns, it provides further improvements and robustness in the walking task. Furthermore, the effectiveness of the proposed QARL algorithm resides in its ability to learn without prior information about outcomes, and in the matching of efficient trials even when starting from scratch. We show that robots with QARL are able to improve their skills adaptively by self-exploration and exploitation of past experiences.

1.4 Organization

In the next chapter we review the state of the art in the related areas, including: Artificial neural networks models; the central pattern generator models; and bipedal robot learning. Different locomotion models are also presented with different oscillator models. In Chapter 3 we present our framework for Success-Failure learning. In Chapter 4, our multi-layered multi-pattern Central Pattern Generator (MLMP-CPG) model that generates diverse locomotion patterns is presented. Furthermore, the contribution that combined the low and high level controller for humanoid robot locomotion is presented in Chapter 5. The success-failure learning approach is improved by proposing the concept of Qualitative Adaptive Reward Learning (QARL) and the notion of vigilance adaptation. Finally, we conclude with a discussion of the presented thesis work and its future perspectives. The organization of this thesis is shown in Figure 1.1.

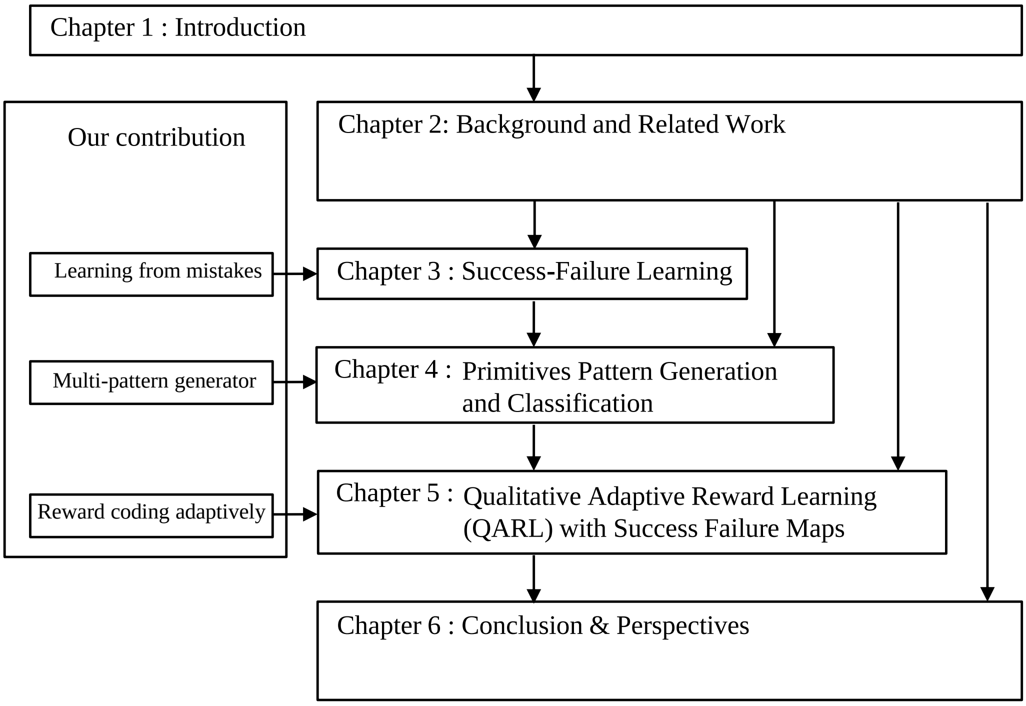

Chapter 1 : Introduction
Our contribution
Chapter 2: Background and Related Work
Learning from mistakes
Chapter 3 : Success-Failure Learning
Multi-pattern generator
Chapter 4 : Primitives Pattern Generation and Classification
Reward coding adaptively
Chapter 5 : Qualitative Adaptive Reward Learning (QARL) with Success Failure Maps
Chapter 6 : Conclusion & Perspectives

Figure 1.1: Thesis' organization.

Chapter 2

Background and Related Work

Contents

2.1 Cognitive Systems

The notion of “cognition” aims to capture the capability of mental activities of human beings for abstracted information from real world. It refers furthermore to their representation and storage in memory. It includes various mental processes like perception, attention, reasoning, learning, recognition, decision, as well as task coordination. The transaction between cognitive states is described by cognitive cycles. Cognitive model can be defined by human thinking and interaction that produce an internal representation of the external world. [Patnaik, 2007] introduces a model that includes various mental processes and their transitions, see Figure 2.1.

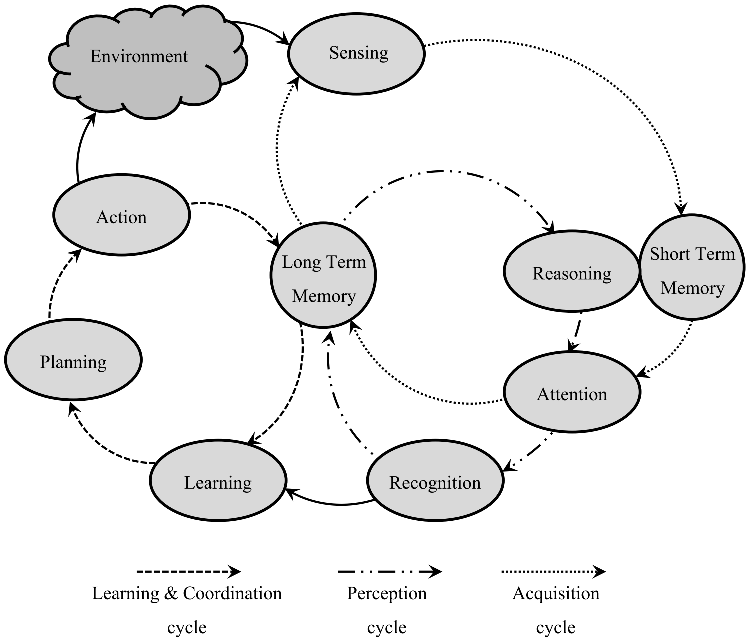


Figure 2.1: A cognitive model includes mental states and three embedded cycles, modified from [Patnaik, 2007].

In such an acquisition cycle, the sensed information are stored in short-term memory and compared with stored information in long-term memory. Perception cycle, with three cognition states: reasoning, attention, and recognition, interprets information stored by acquisition cycle in "STM" with stored knowledge in "LTM". Learning and coordination cycle, with three cognition states: learning, planning, and action, which help the agent to plan its action in the environment after the perception of an action.

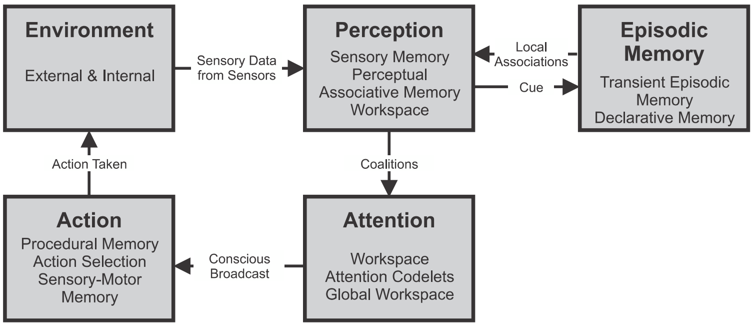


Figure 2.2: The Simplified LIDA cognitive model, extracted from [Snaider et al., 2012].

LIDA A cognitive model from low-level perception/action to high-level understanding and reasoning, called "LIDA" ([Franklin et al., 1998, Snaider et al., 2012]), introduces an integrated cognitive system for Artificial General Intelligence (AGI), see Figure 2.2.

The LIDA cognitive model can be divided into three main cycles: 1) perception (understanding) cycle; 2) attention cycle; and 3) action selection cycle. Sensory Memory module holds incoming sensory information, i.e. external sensory stimulus from the environment and internal sensory stimulus from the proprioception. The Perception cycle is responsible for “making sense” of these sensory stimuli, e.g. object recognition, object classification, and understanding situations (e.g. the cup is empty). The perceptual knowledge-base of this model is called Perceptual Associative Memory (PAM). PAM works as a pass filter for the most relevant part of sensory information stored in Sensory Memory and pass it to the working memory (the workspace). All previous processes in the perception cycle are preconscious.

In the attention cycle, Attention Codelets module brings information to consciousness. Each attention codelet has its own set of information to bring to consciousness. In attention learning, new attention codelets are learned and existing attention codelets are reinforced. The attention codelet is learned whenever it can match the kind of information to bring to consciousness.

In the action selection cycle, the sensory-motor automatism carries out the selected action. Sensory information from sensory memory is required and will be retrieved directly without the aid of the consciousness mechanism. Therefore, the sense-act-sense-act... cycles work much faster than do more complex cognitive cycles. The cognitive cycle is completed when the selected action affects the environment, then another cognitive cycle runs.

Cognitive psychology All cognitive models have in general the same objective, analyze how human think, reason, remember, perceive, and learn. Cognitive psychology has different purposes, such as improve human memory and learning, and increase decision-making accuracy [Neisser, 1967]. Cognitive psychology is the current dominant school of thought in psychology. That is because it focuses on the internal mental states and cycles rather than the observable behaviors as in behaviorism school (1950s).

Cognitives Neurosciences After the cognitive revolution in 1950s, and the birth of cognitive sciences by the convergence of several scientific disciplines interested in all about the human mind, cognitive neuroscience was derived. By studying the behavioral consequences of the brain damage, cognitive neuroscience promises to delineate the connections between the brain anatomy and the functionality of the human mind that is studied in cognitive psychology [Banich and Compton, 2010].

One of the most famous cases was an American memory disorder patient, nicknamed HM, whose Hippocampus and Amygdala were surgically removed in an attempt to treat his epilepsy crises. Without anticipation, HM became amnesic and he could not remember the new information past a period of a few tens of seconds. HM has proved the existence of different types of memory that was supposed by cognitive psychology theories. The patient HM was widely studied from 1957 until his death, which has revolutionized the understanding of the organization of human memory.

The study of human brain regions and its dysfunctionalities has continued to yield new insights for better understanding the structure of human brain and the rules of its regions in human behaviors (e.g. memorizing, risk taking, ...).

Brain recording One of the most important elements that participate in the development of cognitive neuroscience was the electrophysiologically techniques that allows the recording of electric neuronal activity in animals by implanting electrodes. Since 1990s, the Functional Magnetic Resonance Imaging (FMRI) technique takes an important part in cognitive neuroscience researches in animal and human brain mapping. It is a medical imaging technique used in radiology to visualize the internal structure of human brain/body. FMRI measures the brain activity by detecting associated changes in blood flow [Buxton, 2009]. It does not require any surgery nor to undergo shots neither to be exposed to radiation. An important part of this thesis is based on researches on behaviors related to cerebral activity in the human brain measured by FMRI scanner [Brown and Braver, 2005, Cohen et al., 2005]. Another technique for recording brain activity along the scalp is called Electroencephalography (EEG). In addition to the medical objectives, these techniques are widely used to analyze human brain activity and mental action and to understand how human cognitive system work.

Cognitive Robotics - NeuroRobotics The research on cognitive neuroscience adds a biologically-inspired intelligence dimension into robotics research. In contrast to traditional robotics control, which focuses on programming robots to solve one specific task in one environment (sense-act, sense-act, sense-act, ...), cognitive robotics control aims to generate intelligence and adaptive behaviors based on animal or human thinking and learning processes (sense-test-learn-act, sense-test-learn-act,...). To be considered as a human partner, human-like robots and human interactive robots should be provided with sophisticated cognitive systems based on open-ended learning toward developmental robotics. Learning can be considered to be open-ended if it handles tasks that are unknown or even not well-defined previously, [Gomes, 2011].

The human brain develops and learns in open-ended way across its lifespan. A human child can learn tasks that he never did before, this can be referred to the mental and physical development (see Figure 2.3). In contrast to human being, the physical development in robotics is not yet addressed to be in parallel with the mental one.

Traditional robots can percept and act only with the external environment, while robot based cognition can be intrinsically motivated by the internal environment, therefore, it can percept and act with the external and the internal environment to reach an intrinsic motivation (e.g. search for missing knowledge in the word-model

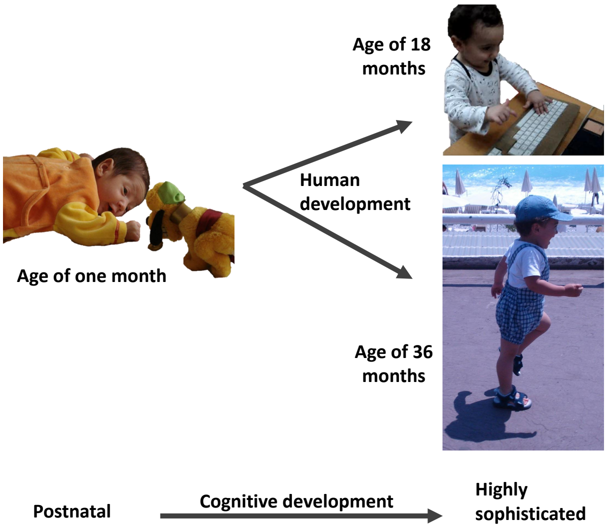


Figure 2.3: A human child development.

and trigger a learning process when needed.) [Oubbati and Palm, 2009].

Figure 2.4 provides a conceptual representation of the intersections between the cognition, the neuroscience, and robotics.

Recent debates in Cognitive Robotics bring about ways to seek a definitive connection between cognition and robotics, ponder upon the questions: "Do robots need cognition? Does cognition need robot?" – [EUCogIII, 2012]

"Cognition say to Robot: You have everything to learn from us, and we have nothing to learn from you!" – (Michael Arbib)

It has been suggested that Cognitive scientists probably need a physical experiment platform like a robot that has quantifiable and measurable capabilities in appropriate dimension to solve their scientific problem that cannot be solved by simulation (Michael Arbib)[Arbib, 2006, Kuniyoshi et al., 2004].

Furthermore, Arbib also brings about the point of view that some robots do not need cognition, and others do. Robots that interact with living organisms need to understand their behaviors (e.g. human robot interaction). To implement cognition in robot, the dimension with which cognition can be characterized needs to be defined while taking into account the capabilities, the requirements, the performance, and the adaptivity with the environment dynamics (Michael Arbib).

Automated reasoning [Wos et al., 1985], non-supervised learning [Barlow, 1989], environment perception [Gonzalez-Aguirre et al., 2011], decision taking [Bicho et al., 2011], and action selection [Ridderinkhof et al., 2004] are some of the shared objectives of cognitive robotic research.

Figure 2.5 shows an example of cognitive architecture that has been developed specifically for a humanoid robotic system, the iCub humanoid robot. Based on the relevant studies on human brain regions (e.g. Hippocampus, Basal Ganglia, Amygdala ...), this cognitive architecture proposed a modulation circuit to effects the

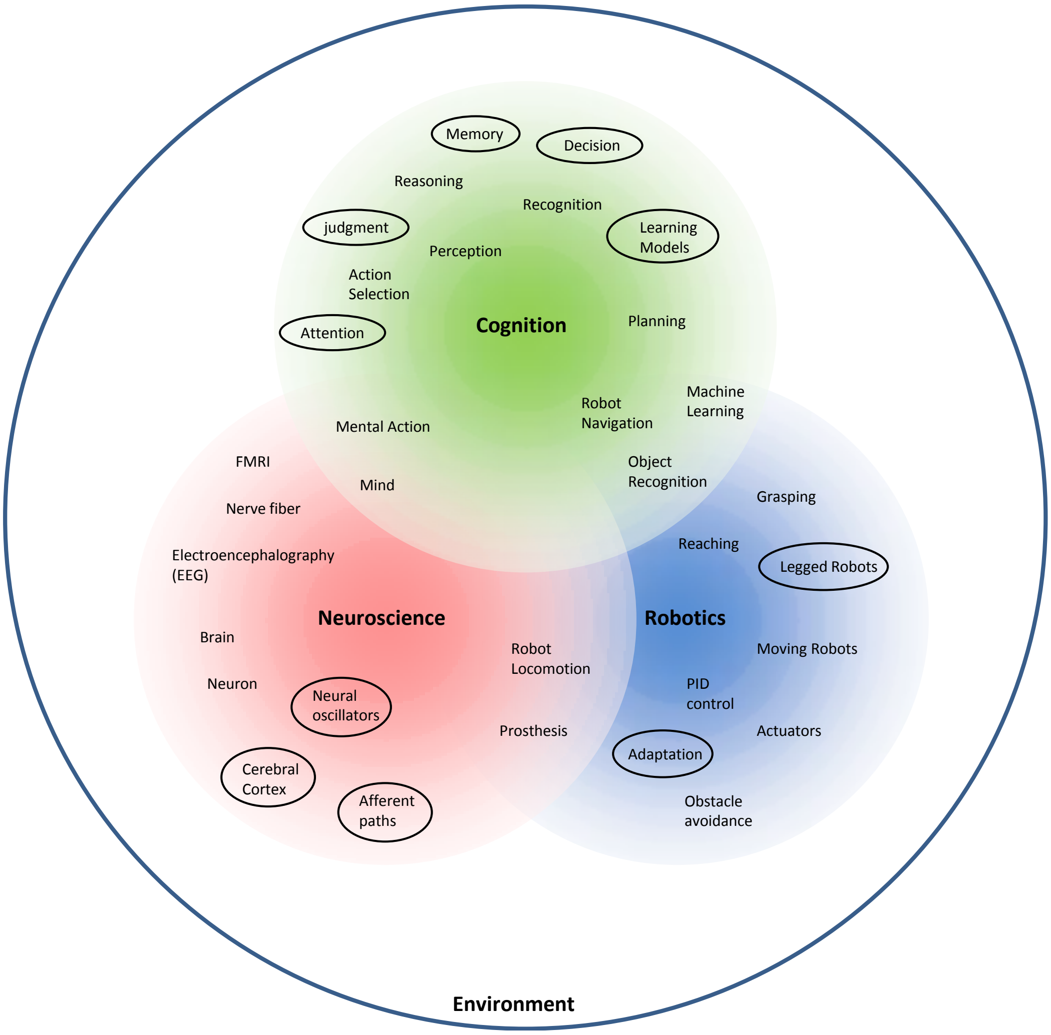


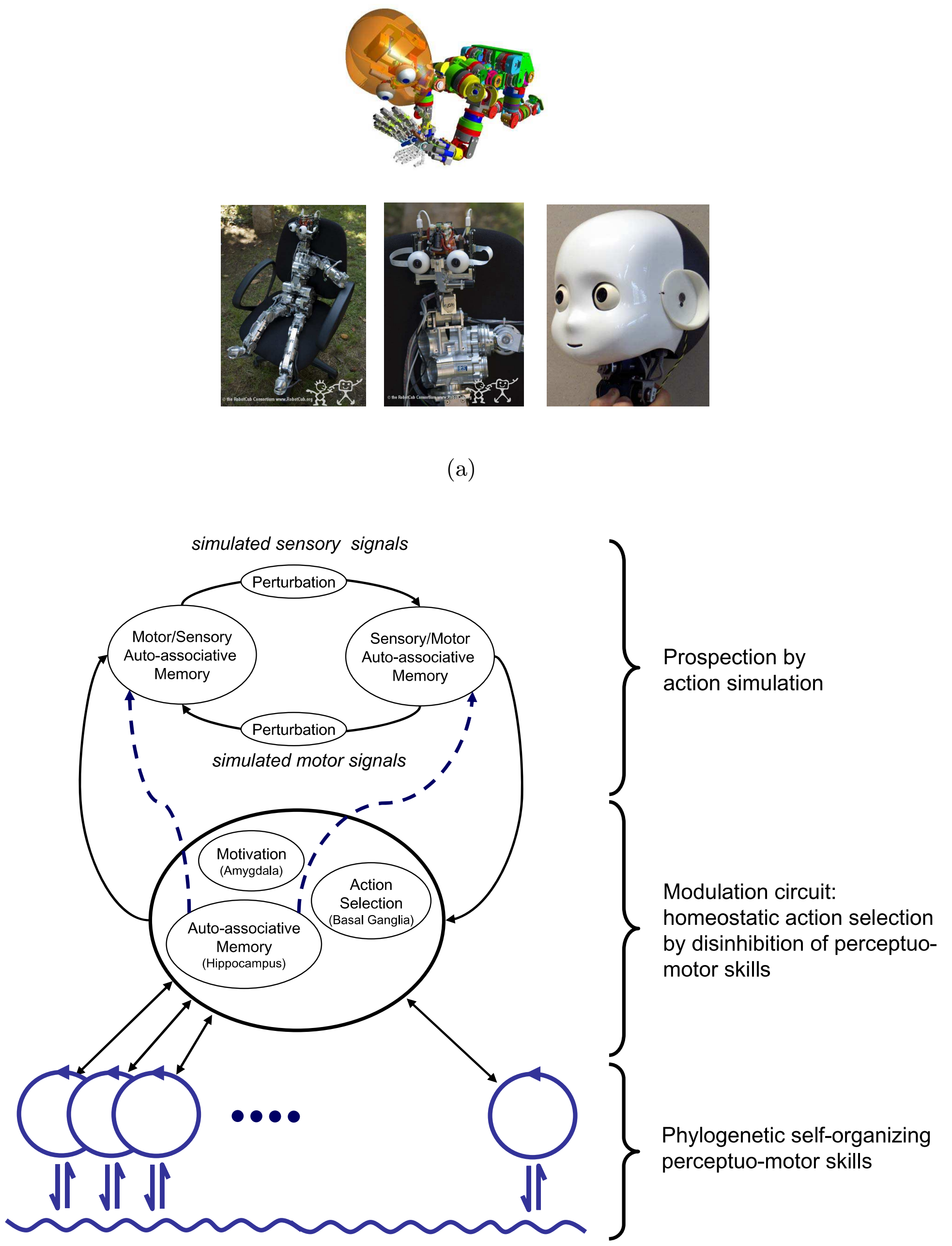
Figure 2.4: CNR = Cognitive Science + Neuroscience + Robotics.

action selection process by disinhibition of perception-action circuits. Competitive and cooperative networks are necessary to implement such cognitive architecture and other brain inspired mechanisms.

Important elements of any Cognitive Systems is its capability to store data, represent, adapt and learn from them. Artificial Neural Networks (ANN) are widely used as an implementation tool for cognitive systems. In the next section, we describe ANN development and focus on the architecture they are based on.

2.2 Artificial Neural Networks

From the engineering point of view, an artificial neural network is a system of data structures associated with each other according to roles that approximate the operation of the biological brain. To date, Artificial Neural Networks are used



(a)

(b)

simulated sensory signals
Perturbation
Motor/Sensory Auto-associative Memory
Sensory/Motor Auto-associative Memory
Perturbation
simulated motor signals
Motivation (Amygdala)
Action Selection (Basal Ganglia)
Auto-associative Memory (Hippocampus)
Prospection by action simulation
Modulation circuit: homeostatic action selection by disinhibition of perceptuo-motor skills
Phylogenetic self-organizing perceptuo-motor skills

Figure 2.5: The iCub robot design (a), cognitive architecture (b), extracted from [Vernon et al., 2007].

by connectionists to perform tasks that require some intellectual or mental state. Technically speaking, the big challenge is always to build an autonomous system that able to discover the environment with minimum of tuning and predefinitions of system parameters. However, a completely independent artificial brain for humanoid robots is the ambition of many studies.

Numerous models of Artificial Neural Networks have been proposed to try to emulate the inner working of the brain. For instance, Teuvo Kohonen proposed the following one:

"Artificial Neural Networks are massively interconnected networks in parallel of simple elements (usually adaptable), with hierarchic organization, which try to interact with the object of the real world in the same way that the biological nervous system does." [Kohonen, 1988]

The simple element in the artificial neural network called node or computational neuron aims to be equivalent to the biological neuron, especially in term of response activity. As in the biological nervous system, nodes are interconnected hierarchically and laterally.

Due to the structure of ANN, massively distributed neurons work in parallel to produce the ability to learn and generalize [Jain et al., 1996]. The ANN is able to give a solution for complex problems that are difficult to be solved or approximated with the traditional systems, pattern recognition [Bishop, 1996], navigation [Tani and Fukumura, 1994], data mining [Bigus, 1996], central pattern generators [Ijspeert, 2008], and many other applications.

Artificial Neural Networks have shown to be flexible and adaptable, which allow greater fault tolerance in comparison to classical control approaches [Nascimento, 1994].

By adapting its internal parameters to the environments' parameters, ANNs have shown that they can acquire knowledge with a production of responses to unknown situation [Hang and Woon, 1997, Salomon, 2003].

A computational neuron can produce a linear or a non-linear activity. A non-linear artificial network is made by the interconnection of non-linear neurons. Non-linear systems have outputs that are not proportional to the inputs. This function allows the network to efficiently acquire knowledge through learning. This is a distinct advantage over a traditionally linear network that is insufficient when it comes to modeling non-linear data [Jain et al., 1996]. The ANN can response correctly to learned samples even if the samples exhibit variability or noise, this represents the fault tolerance aspect of ANN.

The learning process of the ANNs must be able to learn from its surroundings and improve their performance. We can understand learning as the modification of the behavior as consequence of building from experiences. [Mendel and McLaren, 1970] define learning in this way: "Learning is a process by which, the free parameters from a neural network are adapted, through a stimulation process, by the environment in which a network is contained. The kind of learning is determined by the way in which the change in parameter has place."

Two types of learning process are defined: 1) *supervised learning*; and 2) *unsupervised learning*. In the supervised learning the training is controlled by an external element (a supervisor). The supervisor compares the output of the network and the expected output to determine the amount of modification in the weights, e.g. error correction learning. In case of unsupervised learning, no external element is needed, and the network is able to self-organize. Hebbian learning and competitive learning are two kinds of unsupervised learning [Hebb, 1949, Rumelhart and Zipser, 1985].

2.2.1 Biological Background

Research in psychology and neuroscience focuses on the discoveries of how brain works and especially on its process of learning. Human brain consists of a large number of interconnected neurons. The neuron sends out spikes of electrical activity along its axon, which splits into thousands of branches. At the end of each branch, a structure called synapse converts the activity from the axon into electrical effects that inhibit or excite activity between the connected neurons (see Figure 2.6). A spike of electrical activity is sent through the axon when the excitatory inputs are higher than the inhibitory inputs. Learning occurs by changing the efficiency of the synapses that influence the transmission of the spike from a neuron to another.

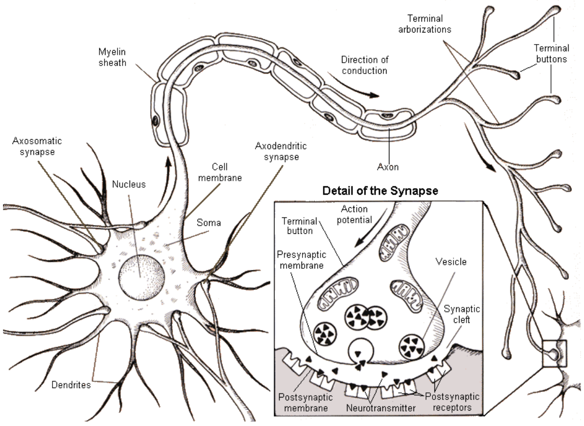


Figure 2.6: Biological neuron and synapse. Image found on internet without copyrights (ww2.coastal.edu/kingw/psyc415/html/detail_synapse_white_bg.gif)

The essential features of neurons and their interconnection and learning are transferred into mathematical model in order to produce a brain-like system that emulates intelligence information processing in the brain. However, the knowledge of the brain structure and of the interconnection between neurons is still far from complete. Different models inspired from biological brain have been proposed, always with the constraint of limitations in computing and of the large number of neurons to modulate.

2.2.2 ANN Models

A neuron The first neuron model was proposed by McCulloch and Pitts [McCulloch and Pitts, 1943]. It has one output channel that represents efferent axon and many input channels that represent afferent axons, all channels are binary. The state of the neuron is given by linear summation of all afferent x_i and comparison of the sum with a threshold s. The neuron is excited if the sum crosses a threshold value. Excitatory and inhibitory input signals are modulated by binary synapses $w = \pm 1$. The activity of this sort of neuron can be described by

$$y = \theta\left(\sum_i w_i.x_i - s\right) \tag{2.1}$$

Here $\theta(x) = 1$ for $x \geqslant 0$ and $\theta(x) = 0$ for $x < 0$. It has been shown that with an appropriate combination of neurons that follow this model; it is able to construct different logical functions. However, it was later shown that it is unable to construct a function like XOR. Furthermore, McCulloch and Pitts did not explain how the connections between neurons can be modified to yield learning.

Hebb's rule The psychologist Donald O. Hebb [Hebb, 1949] suggests that synapses are plastic and that the plasticity changes according to the activity correlation between the presynaptic and the postsynaptic neurons. Hebb's rule can be formulated as:

$$\triangle w_{ji} = \varepsilon.x_i.y_j \tag{2.2}$$

Where w_{ji} is the weight between the pre-synaptic neuron i and the post-synaptic neuron j. ε is a parameter that measures the size of a single learning step, considered as the learning rate. w_{ji} increases if the two neurons i and j activate simultaneously and reduces if they activate separately.

Perceptron In 1957, Rosenblatt made an important step in the domain of artificial neural networks by proposing the *perceptron*, [Rosenblatt, 1957]. The perceptron model is structured as N elements, each of them is preceded by L channels that code T input patterns. The features vector of a pattern is $x = (x_1, x_2, ..., x_L)$. During the *"training phase"*, the perceptron learns to classify the patterns into N classes according to classification examples. The output value for each neuron n from N is computed according to

$$y_n = \theta\left(\sum_{i=1}^{L} w_{ni}.x_i + b\right) \tag{2.3}$$

During the training phase, each neuron adjusts its synapses w_{ni} in the way that it reacts only to the corresponding input patterns (patterns of its class C_n) with an output value $y_n = 1$, b is the bias. This network is able to separate the space of input patterns into $L - 1$ hyperplanes. However, it cannot achieve the separation if the space of patterns has too convoluted classes [Jain et al., 1996]. This limitation of one layer perceptron has been solved by introducing multilayer networks [Rosenblatt, 1961]. Different training approaches have been proposed afterward in order to adjust the weights of the synaptic connections with hidden layers neurons. In 1986, Rumelhart *et. al.* found a method known as “Backpropagation” that associates among the input patterns and its classes [Rumelhart et al., 1986]. In this network, there are no defined classes in the hidden layer, the errors attached to the neurons of this layer are determined by back-propagating the errors from the neurons of the output layer (classes are defined for this layer). This method is considered as a generalization of a delta rule for supervised learning in perceptron [MacKay, 2003]. In supervised learning, the desired output is always required to update the synapses (weights) [MacKay, 2003].

Delta rule A delta rule is a gradient descent learning rule also considered as a Least Mean Square (LMS) method, developed by Windrow and Hoff. It is one of the most widely accepted learning rules [Widrow and Hoff, 1960]. In the case of perceptron with a linear activation function, the simplified form of the delta rule can be defined as

$$\triangle w_{ji} = \alpha(t_j - y_j).x_i \tag{2.4}$$

Where α is the learning rate, t_j is the target output, y_j is the actual output, and x_i is the ith input. The delta rule uses gradient descent to minimize the error from the perceptron network's weights, which is calculated according to

$$E = \sum \frac{1}{2}(t_j - y_j)^2 \tag{2.5}$$

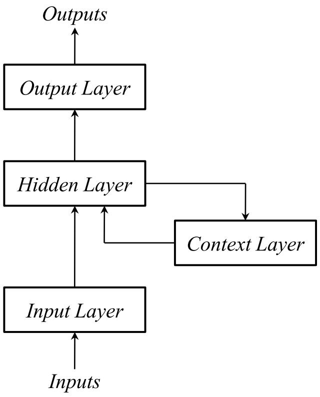


Figure 2.7: Recurrent neural network architecture.

Recurrent neural network The classical feedforward neural network has an input layer, an output layer, and one hidden layer. In order to make this network able to perform temporal processing and learn sequences, a “context” layer is added to the structure to retain information between successive observations [Elman, 1990, Jordan, 1990]. At each time step, a new input vector is presented to the input layer and the content of the hidden layer from the previous time step is passed to the context layer, which then provide feed back to the hidden layer in the next time step (see Figure 2.7). This network is called “**recurrent neural network** (RNN)”. RNN is widely used in sequence recognition, sequence prediction, and temporal classification.

Hopfield network In 1982, John Hopfield introduced RNN with symmetric connections between all neurons [Hopfield, 1982]. The idea behind the Hopfield network is that patterns are encoded by a weight matrix. With an input that contains a part of these patterns, the dynamics of the network is able to retrieve the patterns encoded by the weight matrix. This is referred to as Content Addressable Memory (CAM).

CTRNN In 1995, Randall D. Beer introduced the model of Continuous-Time Recurrent Neural Network (CTRNN) [Beer, 1995]. Neurons of this network update the internal state by a differential equation,

$$\tau_i(dy_i/dt) = -y_i + \sum_{j=1}^{N} w_{ij}.\sigma(y_j - b_j) + I_i \qquad i = 1, 2, ..., N \tag{2.6}$$

Where τ_i is the time constant, b_i is the bias of the neuron i. I_i is an external input for neuron i, w_{ij} is the weight between neuron i and neuron j. $\sigma(x) = 1/(1+e^{-x})$ is the standard logistic activation function. This model is widely used to modulate the motor neuron activation in robot locomotion [Gallagher et al., 1996, Manoonpong et al., 2007, Hoinville et al., 2011].

MTRNN Tani *et. al.* propose an action generation model based on Continuous-Time Recurrent Neural Network with multiple timescale (MTRNN) [Yamashita and Tani, 2008, Namikawa et al., 2011] (see Figure 2.8). The network receives two different inputs, proprioceptive and vision inputs, and generates predictions of the future state that is related to the capacity of the CTRNN in preserving the internal state. These predictions of the proprioception are sent to the robot as target joint angles. In Figure 2.8, context unit is divided according to the time constants into two groups, a group for fast context units and a group for slow ones. As in previous models of Artificial Neural Networks, every unit in the CTRNN is connected to the other units, including it. In the training mode, the synaptic connections are updated with the function of the error between the proprioceptive prediction and the generated behavior based on synaptic weights.

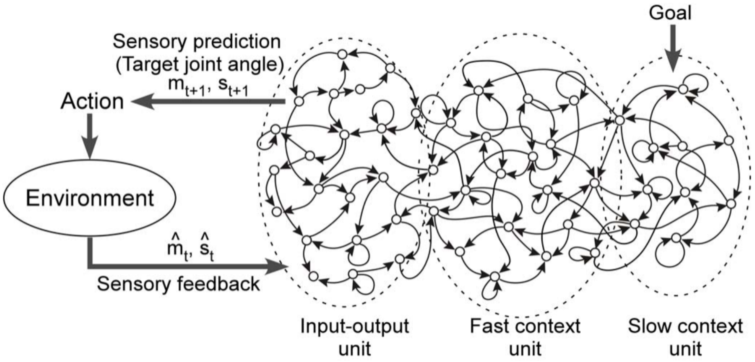


Figure 2.8: Action generation model, extracted from [Yamashita and Tani, 2008]. Where $\widehat{m}_t$ is the proprioception, and $\widehat{s}_t$ is the vision sense.

The model proposed by Tani *et, al.* learns to generate temporal patterns of sensory motor sequences in order to coordinate the movements of a humanoid robot through a high dimensional sensori-motor control while achieving upper body tasks (see Figure 2.9).

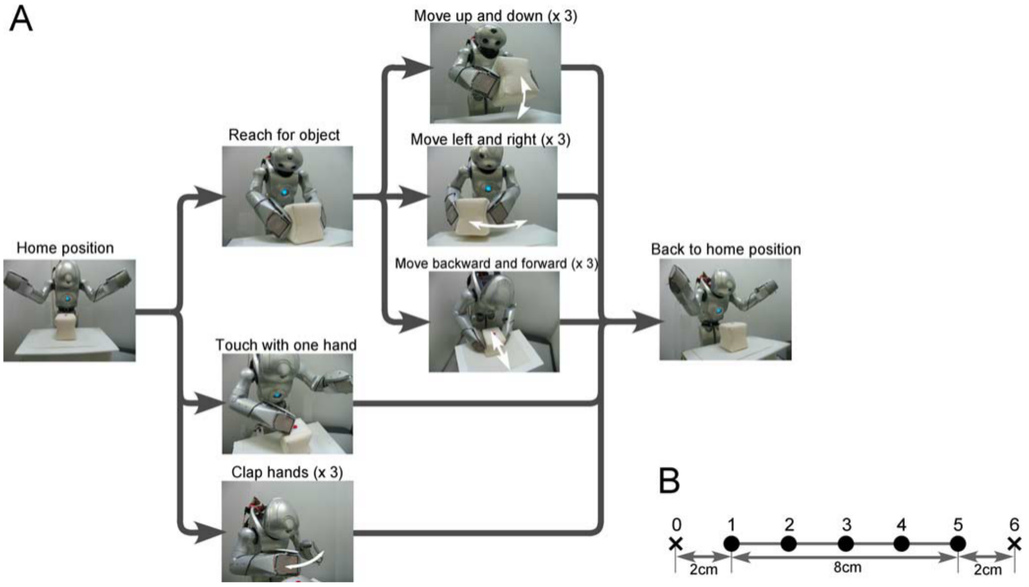


Figure 2.9: A humanoid robot achieving upper body tasks designed by MTRNN, extracted from [Yamashita and Tani, 2008].

Namikawa *et. al.* improves the MTRNN, shown in Figure 2.8, in the aim to correspond to the hierarchy of the prefrontal cortex, the supplementary motor area, and the primary motor cortex involved in action generation [Namikawa et al., 2011]. As supposed the action is a combination of primitives or chunks in a specific consequence, each chunk can be reused in building other actions. [Byrne, 2003] shows

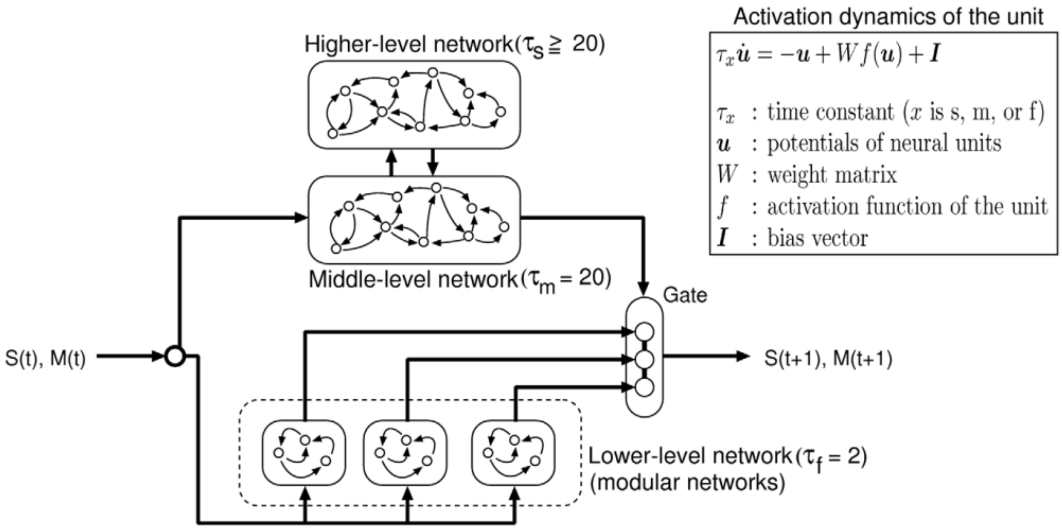

Activation dynamics of the unit
$\tau_x \dot{\boldsymbol{u}} = -\boldsymbol{u} + Wf(\boldsymbol{u}) + \boldsymbol{I}$
τ_x : time constant (x is s, m, or f)
$\boldsymbol{u}$: potentials of neural units
W : weight matrix
f : activation function of the unit
$\boldsymbol{I}$: bias vector
Higher-level network($\tau_s \geqq 20$)
Middle-level network($\tau_m = 20$)
Gate
S(t), M(t)
S(t+1), M(t+1)
Lower-level network($\tau_f = 2$)
(modular networks)

Figure 2.10: Hierarchical neural network for action generation, with three levels and a time constant, extracted from [Namikawa et al., 2011].

how Gorillas' actions can be segmented into elements that can be obtained by extracting statical structures by imitation. The Hierarchical Neural Network shown in Figure 2.10 is based on different cortical studies. A monkey electrophysiological study showed that some neurons of the presupplementary motor area (preSMA) become active at the beginning of each motion primitive in the generation of trained sequences [Nakamura et al., 1998]. This study suggests that the prefrontal cortex and the preSMA are involved in segmenting sequences into motion primitives and in selecting the next primitive, while motor-related areas, including the premotor cortex and the primary motor cortex, are involved in cognitive control within each primitive [Sakai et al., 1999]. This hierarchical organization in the cortical areas is discussed in [Namikawa et al., 2011] (Figure 2.10).

CMAC In 1975, [Albus, 1975] proposed the Cerebellar Model Articulation Controller "CMAC", a neural network that models the role of cerebellum in motor control of body's parts. The cerebellum is a brain region involved in regulation of muscle tone and motor activity [Banich and Compton, 2010]. *"In large part, it is the region of the brain that allows a pianist to play a piece of music seamlessly or a pitcher to throw a ball fluidly"*. Any damage in the cerebellum affects not only the equilibrium, but also the precision of movements, see Figure 2.11.

The CMAC model is a form of associative memory that was designed to control robotic manipulators; it is able to learn motor behavior. It merges the input command from high center and the feedback from joint's sensors, muscles, and skin into a set of memory addresses where the correct motor responses are stored, [Albus, 1975]. The CMAC generates different strings of responses for different commands from high center even for same stimuli. Each command will select a corresponding region in

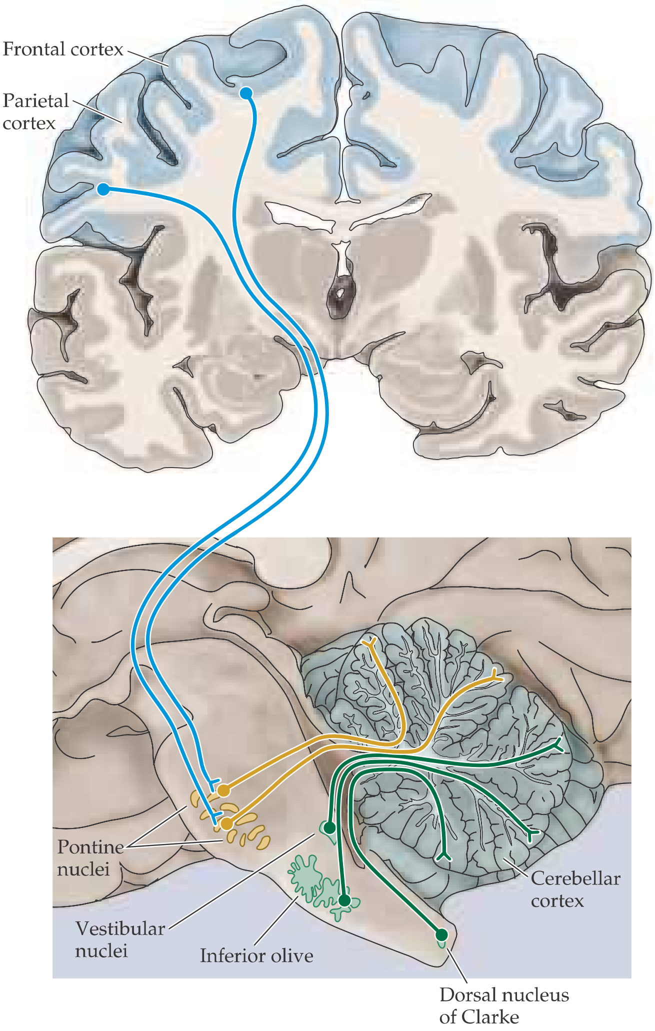


Figure 2.11: Functional organization of the inputs to the cerebellum. A coronal section of the brain and a sagittal section through the cerebellum that show the major inputs of the cerebellum from the motor cortex “high centers”, vestibular system, spinal cord, and brainsteam. Extracted from Cognitive Neuroscience book [Banich and Compton, 2010].

the memory where motor primitives for this command are stored.

2.2.3 The SOM (Self-Organizing Maps)

Teuvo Kohonen [Kohonen, 1984] introduced the “Self-Organizing Map (SOM)” that can convert the similarity of patterns into a proximity of activated neurons. In SOM, the spatial distribution of the neurons plays an important role in the response of the network. The research presented in this thesis uses SOM as associative memory for walking patterns. Let us explain the architecture of this model and its algorithm.

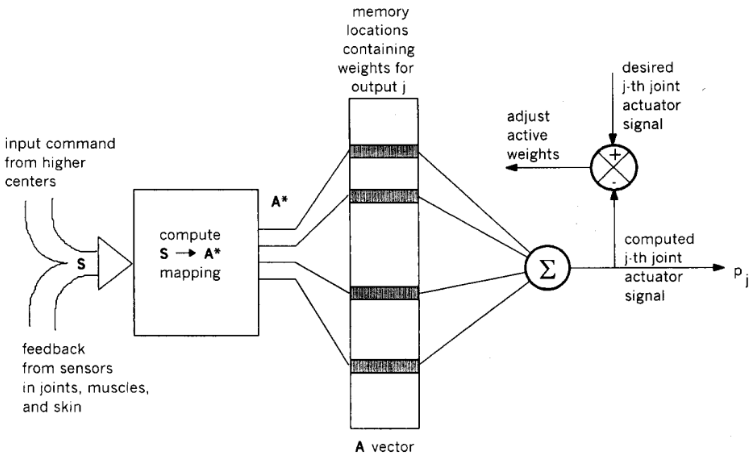

memory locations containing weights for output j
input command from higher centers
S
compute S → A* mapping
A*
feedback from sensors in joints, muscles, and skin
A vector
adjust active weights
desired j-th joint actuator signal
+
-
Σ
computed j-th joint actuator signal
p j

Figure 2.12: A block diagram of the CMAC system for a single joint. S is the combination of high centers inputs and feedback from sensors and joint muscles. A^* is memory locations selected by mapping. The output signal defines the desired signal of the joint actuator. Extracted from [Albus, 1975].

The structure of a SOM network is biologically inspired: stimulus with same nature excite same brain region. Neurons are organized in the cortex in a manner that interprets all possible types of stimulus. In the same way, the self-organizing map unfolds in order to represent a data set of input patterns. Each neuron will be responsible to represent a group of patterns that are close to each other in the pattern space. The map divides the space into different areas according to the number of its neurons. Each area can be assigned into a "reference vector" that represents the weights of the corresponding neuron.

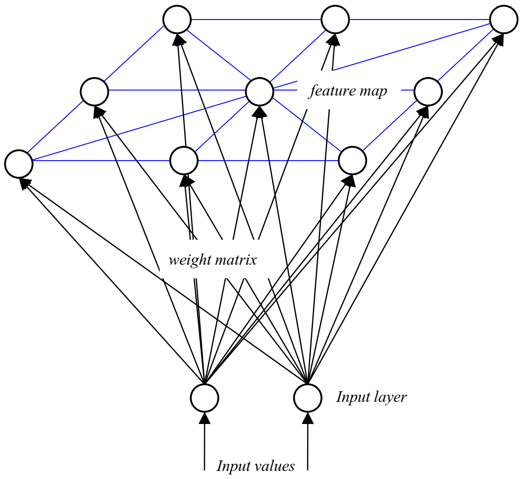


Figure 2.13: A self-organizing map.

The neurons of the map are organized as a grid (uni-, bi-dimensional, or three-dimensional). Each neuron is associated with a reference vector that is responsible for an area in the data space (input space). All these vectors divide together the entire input space into areas, one area per vector (neuron).

In a self-organizing map, the reference vectors provide a discrete representation of the input space. They are positioned in a way that preserves the topology of input space. Keeping the neighborhood relations in the grid allows easy indexing according to the coordinates in the grid. This is useful in various fields such as classification of textures, interpolation between data, and visualization of multidimensional data.

We can formulate this by: let A be the rectangular grid of neurons in a self-organizing map. The map assigns to each input vector $v \in \omega$ a neuron $r \in A$ whose weight vector w_r is closest to v. Mathematically, this association is expressed by a function:

$$r = \arg\min_{r \in A} \|v - w_r\| \tag{2.7}$$

SOM as a learning mechanism We can then formulate SOM as a learning process in the following manner. In the absence of any prior information, the weights vector for each neuron in the map will be initialized randomly. According to the weights vector for the neuron, one neuron that is closest to the stimulus is rewarded with a weights change to respond better to another stimulus of the same nature as the previous. Thereby, neighboring neurons of the winner will be rewarded with a multiplicative gain less than one. At the end of learning phase, the neurons do not move, or move very little after each iteration, thus the self-organizing map covers all presented stimulus.

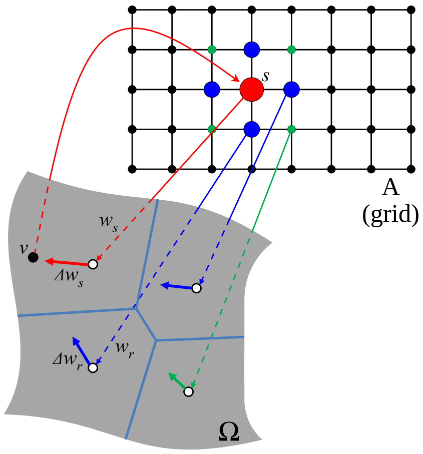


Figure 2.14: The adaptation step in Kohonen's model.

Mapping the input space is achieved by adapting the reference vectors w_r. The adaptation for a stimulus is made by the learning algorithm based on the competition between neurons while taking into account the degree of neighborhood between neurons. A random sequence of input vectors is presented for learning. With each vector, a new adaptation cycle is started. For each vector v in the sequence, we determine the winner neuron, that is to say, the neuron whose reference vector is the nearest to v.

$$s = \arg\min_{r \in A} \|v - w_r\| \tag{2.8}$$

The winner neuron s and its neighbors (defined by a neighborhood function) move their reference vectors toward the input vector.

$$w_r^{t+1} = w_r^t + \Delta w_r^t \tag{2.9}$$

with

$$\Delta w_r^t = \varepsilon \cdot h \cdot (v - w_r^t), \tag{2.10}$$

Here $\varepsilon = \varepsilon(t)$ represents the learning rate. $h = h(r, s, t)$ is the neighborhood function that defines the connections between neurons.

The neighborhood function describes how neurons in the vicinity of the winner s are trained in the learning of the vector v. To express this proximity, a Gaussian function is widely used, such that

$$h(r, s, t) = \exp\left(-\frac{\|\vec{r} - \vec{s}\|}{2\sigma^2(t)}\right) \tag{2.11}$$

Where σ is the coefficient of neighborhood. Its role is to determine a radius of the neighborhood around the winner neuron.

The neighborhood function h forces the neurons located in the vicinity of s to align their reference vectors with the input vector v. The more the neighbor neuron is close to the winner in the grid, the more its displacement is important. The amount of correction of the reference vectors is related by the distance to the winner in the grid. During learning, the map shifts from a random state into a stable state that describes the topology of input space with respect to the connection order in the grid.

The map reflects the distribution of points in the input space. Areas where training vectors v are learned with a high probability are mapped with better resolution than the areas in which training vectors v are learned with a small probability of occurrence. By preservation of topology of the input space, neurons tend to discretize the space in an orderly manner.

The presentation of artificial neural networks and learning approaches set the stages in which they can be applied to cognitive systems to represent data and learn it either by supervised or non-supervised learning. In the next section, a focus is given on a special type of neural network that is employed as central pattern generators (CPG). Models of CPG are commonly used to model biological as well as robot locomotion. First, biological evidences of CPG are presented, and then computational CPG models are described. The use of CPG in legged robot locomotion research is presented.

2.3 The Central Pattern Generator

Central Pattern Generators (CPGs) are set of interconnected neurons that can produce motion patterns without input from higher centers and without sensory feedback. Biological evidences showed that the Central Pattern Generators of the spinal cord play an important role in the control of animals' locomotion. Longstanding animals studies suggest that locomotion is mainly generated in the spinal cord, by a combination of a central pattern generator (CPG) and reflexes receiving adjustment signals from the cerebellum [Brown, 1911, Orlovsky et al., 1999, McCrea and Rybak, 2008].

Motivated by the adaptive and robustness of biological locomotion mechanisms, much of these studies have been taken into account in robot's locomotion gait generations, in order to emulate such mechanisms, especially in legged robots [Taga et al., 1991, Kimura et al., 1999, Endo et al., 2008, Morimoto et al., 2008, Ijspeert, 2008]. These works based on biologically-inspired walking of legged robots have the advantage of not requiring a perfect knowledge of the robot's dynamics, while still achieving robust locomotion. In the next sections, biological background of the CPG is given. Next, the utilization of the CPG in robot locomotion is presented with different selected models and robots.

2.3.1 Biological Approach

The coordination of body movements plays an important role to keep animals alive and help them to explore their environment [Purves et al., 2004]. Local circuits (central pattern generator) were identified in the spinal cords of vertebrates as responsible for control of locomotion movements. The CPG is a set of *sensory neurons, interneurons, and motor neurons* localized repetitively in the spinal cord. They control locally the sequence of contraction / relaxation of body's muscles. The sensory neurons detect the stretching and the contraction of muscles, while the interneurons fire rhythmically and coordinate sensory information and motor neurons signals. The three types of the above-mentioned neurons are also involved in reflexive responses of stimuli. The higher centers are involved in locomotion by controlling the spatial and temporal activity patterns of the individual limbs. However, it has been shown that a cat's limb can produce walking patterns even with a cut of the spinal cord at the thoracic level (see Figure 2.15).

Neurobiological studies on de-cerebrated cats have proposed computational spinal circuitry models responsible for animal locomotion [Brown, 1911, Orlovsky et al., 1999, Rybak et al., 2006, McCrea and Rybak, 2008]. The rhythmic

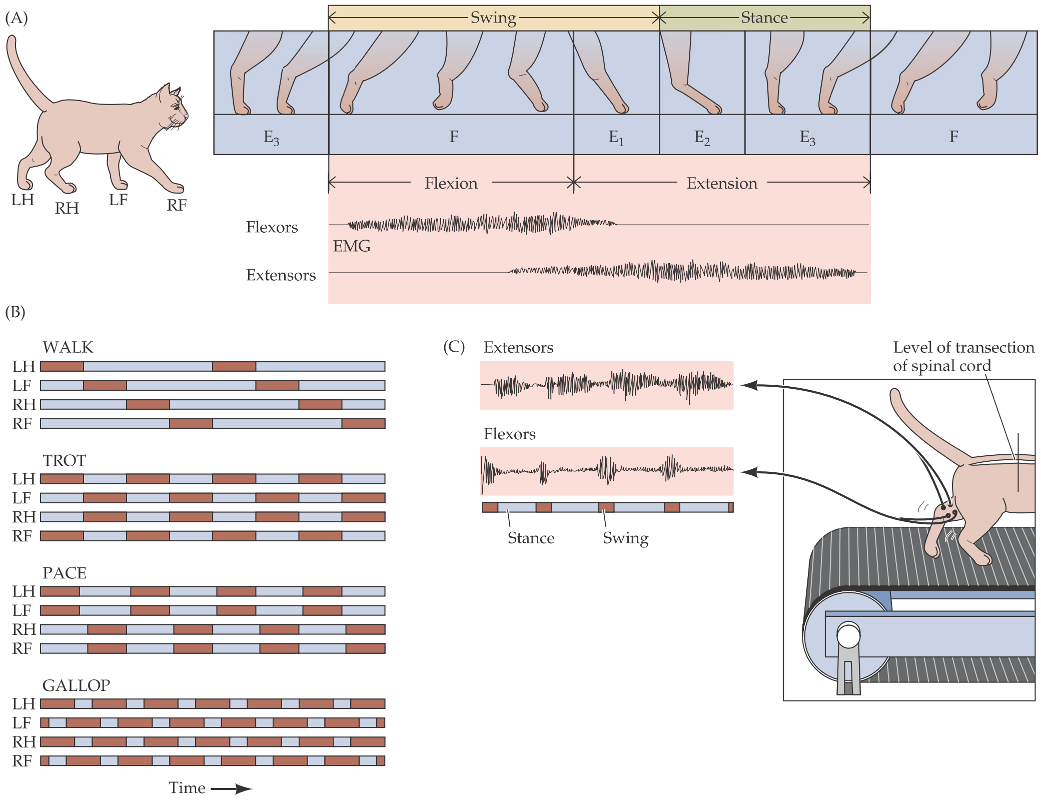


Figure 2.15: The locomotion cycle organized by central pattern generator in terrestrial mammals. (A) The step cycle, showing the relation between leg extension and flexion, and swing and stance phases. EMG indicates electromyographic recordings. (B) Stepping movements for different gaits. (C) The cat is able to walk on treadmill even with transection of the spinal cord at the thoracic level that isolates hindlimb segments from the cord. Extracted from [Purves et al., 2004].

patterns in cat limbs can be generated in the absence of high center signal and is able to control the timing and the coordination of limbs motion [Brown, 1911]. Each joint appears to have its own CPG, which can be coupled to the CPG of another joint in order to achieve complex movements. These CPGs controlling such behaviors in animals locomotion can be responsible for rhythmic movements in human locomotion [Choi and Bastian, 2007].

Several schemes for the spinal CPG have been proposed to generate rhythmic movements: "half-center CPG" proposed by Brown [Brown, 1914], "half-center CPG" with more complex patterns of motorneuron activity was introduced by Perret et al. [Perret et al., 1988] and "half-center CPG" with sensory input proposed by Orlovsky et al. [Orlovsky et al., 1999]. One drawback of these models is the direct excitatory connection between the rhythm generator interneurons and motorneurons that any changes in the interneurons layer will affect simultaneously the motorneurons layer. A more sophisticated architecture is required to face the adaptation with the environment changes. Two and three levels CPGs with rhythm generation and pattern formation circuitry have been proposed by [McCrea and Rybak, 2008] and [Koshland and Smith, 1989]. These models separate cycle timing and motoneurons activation. [Rybak et al., 2006] propose a model of CPG with two levels, a half center rhythm generator neurons RG, and a pattern formation neurons PF, see Figure 2.16.

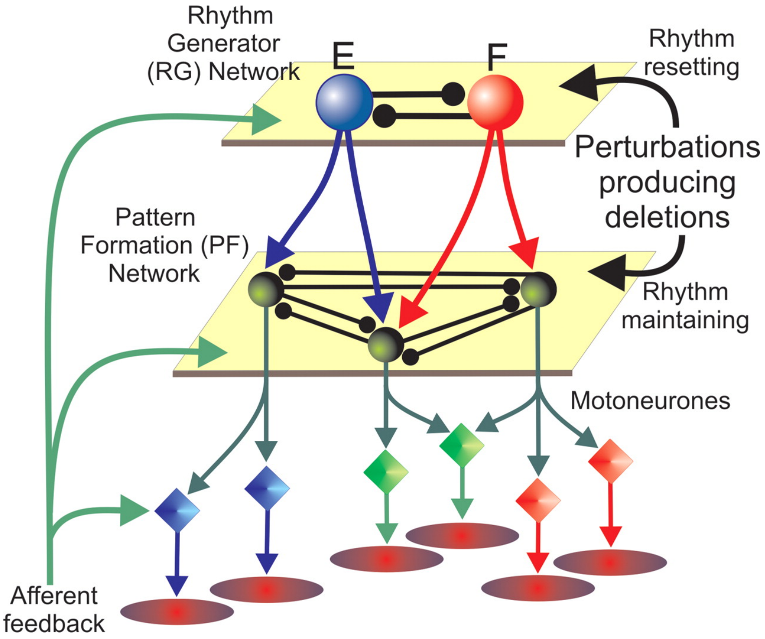

Rhythm Generator (RG) Network
E
F
Rhythm resetting
Perturbations producing deletions
Pattern Formation (PF) Network
Rhythm maintaining
Motoneurones
Afferent feedback

Figure 2.16: Schematic illustration of the two-level central pattern generator (CPG) concept, extracted from [Rybak et al., 2006].

The hypothesis of a two-level organization of the CPG allows to control separately the rhythmic patterns and the activation of these patterns. The CPG shown in Figure 2.16 is composed of half-center rhythm generator (RG) that controls the rhythm and duration of extension and flexion phase of muscle, and a pattern formation interneuron layer that excites motorneurons population and inhibits other PF population. The PF network distributes rhythmic input from RG network among the motorneurons pools. Activation of a particular PF population will activate the corresponding motorneurons population and therefore the corresponding muscle [Rybak et al., 2006].

Afferent feedback may affect the CPG at the RG level; this can produce alterations in the generated locomotor rhythm in term of phase shifting or in term of phase resetting. If the effect of the afferent feedback occurs at the pattern formation level, it may affect the activation and the timing of phase transition without phase shifting or resetting.

When afferent feedback affects the motorneuron population without the intermediate of RG or PF levels, reflex action can be produced. The sensory-motor circuitry is involved in different reflex types (e.g. knee jerk reflex, stretch reflex,

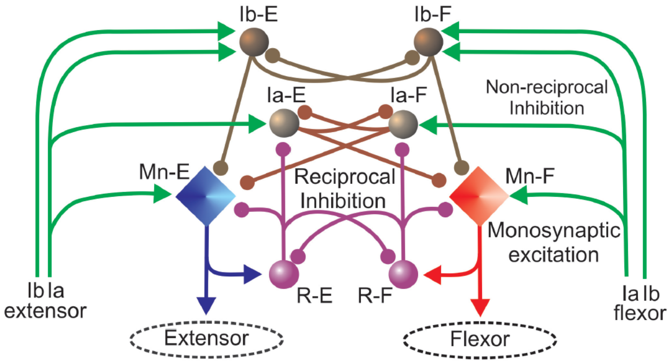


Figure 2.17: Sensori-motor Circuitry, extracted from [Rybak et al., 2006].

flexion reflex). Figure 2.17 shows the sensory-motor model that makes part of the CPG model.

The neuronal structures of CPG have inspired its usage in the modulation of robot locomotion control. [Ijspeert et al., 2007] use rhythm generator to generate the patterns, while [Manoonpong et al., 2007] use a sensori-motor model to produce walking pattern. These two works are two key exemplars of detailed neural models applied to robotics.

2.3.2 CPG models

In legged animal locomotion, the control of muscle-skeletal system is ensured in a hierarchical manner by a high-level nervous system and spinal cord nervous system (CPG) for rhythmic locomotion patterns. Several mathematical models have been proposed for rhythmic pattern generation (like, Matsuoka [Matsuoka, 1985], Taga [Taga et al., 1991], Rowat & Selverston [Rowat and Selverston, 1991]) and some of them were implemented in legged robots locomotion [Endo et al., 2004, Righetti and Ijspeert, 2006, Ijspeert et al., 2007, Endo et al., 2008].

Among many models of motor pattern generator, Matsuoka oscillator is widely used in robotic research. The original model was developed by Brown to modulate the activation of flexor and extensor cat limbs muscles in walking [Brown, 1914]. The Matsuoka model is based on mutual inhibition of two neurons with self-inhibition effect. Due to these connections, each neuron can produce rhythmic activity. The mathematical representation of the model is as follow:

$$\tau.\frac{du_1}{dt} = -u_1 - w.y_2 - \beta.v_1 + u_0 \tag{2.12}$$

$$\tau.\frac{du_2}{dt} = -u_2 - w.y_1 - \beta.v_2 + u_0 \tag{2.13}$$

$$\tau'.\frac{dv_1}{dt} = -v_1 - y_1 \tag{2.14}$$

$$\tau'.\frac{dv_2}{dt} = -v_2 - y_2 \tag{2.15}$$

$$y_i(u_i) = max(0, u_i), i = 1, 2. \tag{2.16}$$

Where u_i is the state of the ith neuron. y_i is the output of the ith neuron. v_i represent the degree of adaptation of self-inhibition of the ith neurons. u_0 is an external input with a constant rate. τ and τ' are time constants of the inner state and the adaptation effect, respectively. The constant input u_0 changes the amplitude, and the time constants change the frequency of oscillation. w is the inhibitory connection weight between the two neurons.

Taga et al. used the Matsuoka oscillator to control a biped robot in a 2D simulated environment [Taga et al., 1991]. Their controller is based on the Matsuoka model as a unit oscillator, a neural rhythm generator was constructed for bipedal locomotion (see Figure 2.19). Each oscillator unit produces a torque to be applied to a specific joint.

The neural rhythm generator is described by the following differential equations:

$$\tau_i.\frac{du_i}{dt} = -u_i + \sum_{i,j=1}^{N} w_{ij}.y_j - \beta.v_i + u_0 + feed_i \tag{2.17}$$

$$\tau'_i.\frac{dv_i}{dt} = -v_i - y_i \tag{2.18}$$

$$y_i(u_i) = max(0, u_i), i = 1, 2, ..., N. \tag{2.19}$$

Where N is the number of neurons (N=12 in Figure 2.19).

$feed_i$ is an external input that represents the feedback sensor signal and the interaction between the robot and the environment, forming the closed-loop part of the CPG model. Thus, it represents the proprioceptive and exteroceptive information. The former is the sensory feedback from the musculo-skeletal system, while the latter can be postural sensory information, visual, somatic, vestibular information that represent the interaction with the environment.

This model has been widely used in legged robots locomotion to generate walking patterns [Endo et al., 2004, Matsubara et al., 2006, Liu et al., 2008]. In Matsuoka

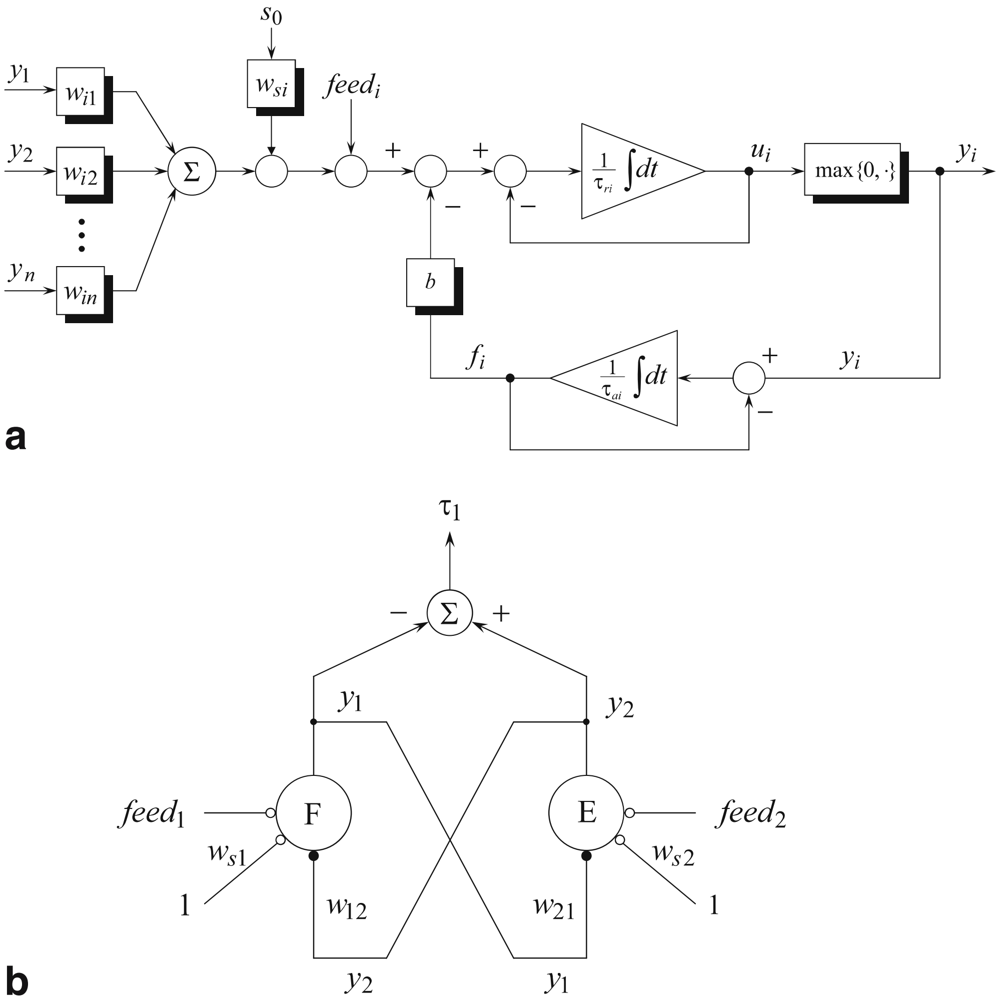


Figure 2.18: (a) Diagram of Matsuoka neural oscillator.(b) Matlab Simulink model of one neuron. Extracted from [Liu et al., 2008].

and Taga models the oscillatory behaviour arises from the mutual connection between neurons, however, each neuron cannot produce oscillation activity without coupling. [Rowat and Selverston, 1997] propose an oscillatory neural model based on two cells with self-rhythmic generation ability. Each cell can independently generate its own pattern according to two parameters, which are related to the membrane conductivity for fast and slow currents.

The membrane currents of the neuron are separated into two classes, fast and slow, according to their time responses. The sum of all fast currents is modeled by a single fast one, and a single slow current is used to model the sum of all slow ones. This model cell has two differential equations, one for membrane potential V, derived from current's conservation, and one for lumped slow current q, derived from current's activation (see equations (2.20 and 2.21)).

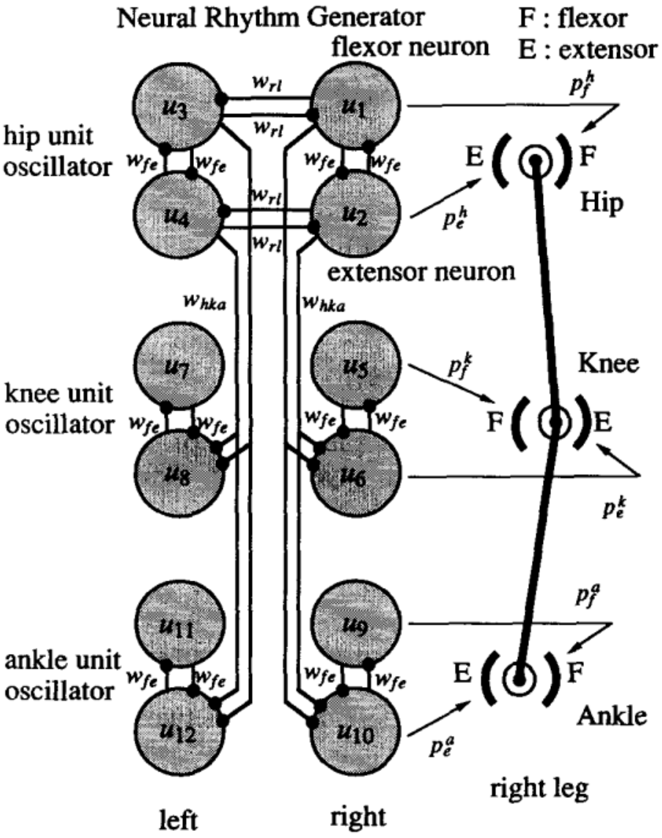


Figure 2.19: Neural rhythm generator for bipedal locomotion, extracted from [Taga et al., 1991].

$$\tau_m.\frac{dV}{dt} = -(fast(V, \sigma_f) + q - i_{inj}) \tag{2.20}$$

$$\tau_s.\frac{dq}{dt} = -q + q_\infty(V) \tag{2.21}$$

$$\tau_m < \tau_s \tag{2.22}$$

While the fast current is supposed to activate immediately, the membrane time constant τ_m is assumed to be significantly smaller than the slow current's time constant for activation τ_s. The ratio of τ_s to τ_m was fixed to 20 in [Rowat and Selverston, 1997], but when the ratio is as small as 1.5 most model patterns still arise. The injected current is i_{inj}. An idealized current-voltage curve for the lumped fast current is given by:

$$fast(V, \sigma_f) = V - A_f.tanh((\sigma_f/A_f)V) \tag{2.23}$$

The fast current represents the sum of a leak current and an inward Ca^{++}. The dimensionless shape parameter for current-voltage curve is given by:

$$\sigma_f = \frac{g_{Ca}}{g_L} \tag{2.24}$$

Where g_L is a leak conductance and g_{Ca} is the calcium conductance.

Figure 2.20(a) shows the fast IV curve Equation (2.23) with different values for σ_f that define the slope of the reverse part of N shape.

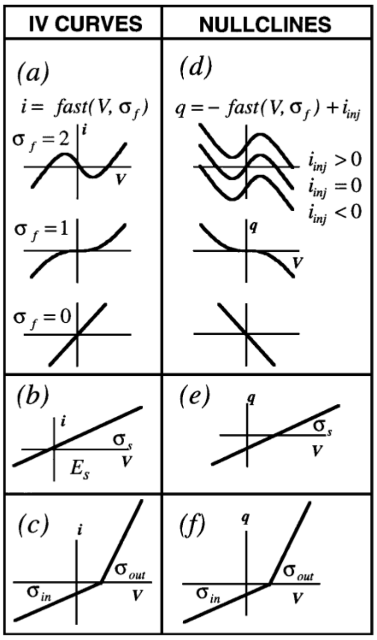


Figure 2.20: IV curves and nullclines in the cell model. Extracted from [Rowat and Selverston, 1997].

$q_\infty(V)$ is the steady state value of the lumped slow current, which is given by:

$$q_\infty(V) = \sigma_s(V - E_s) \tag{2.25}$$

$q_\infty(V)$ is linear in V with a reversal potential E_s (see Figure 2.20(b)). σ_s is the potassium conductance g_K normalized to g_L. σ_s is given by:

$$\sigma_s = \frac{g_K}{g_L} \tag{2.26}$$

This model can be extended to show two different conductances for inward and outward, with conductance σ_{in} for inward slow current smaller than for outward slow current σ_{out}, see Figure 2.20(c). The steady state value of the lumped slow current is given by:

$$q_\infty(V) = \begin{cases} \sigma_{in}(V - E_s) & if \quad V < E_s \\ \sigma_{out}(V - E_s) & if \quad V > E_s \end{cases} \tag{2.27}$$

q and i_{inj} have the dimension of an electrical potential. A true current is obtained by multiplying the model current by a leak conductance g_L. V, E_s, i_{inj}, and q are given in millivolts while τ_s and τ_f are expressed in milliseconds.

Figure 2.21 shows the (V,q) phase plane for the cell model and the rhythmic activity generated with the corresponding values of σ_s and σ_f. The figure shows four points on the phase plane with their corresponding positions on the generated pattern.

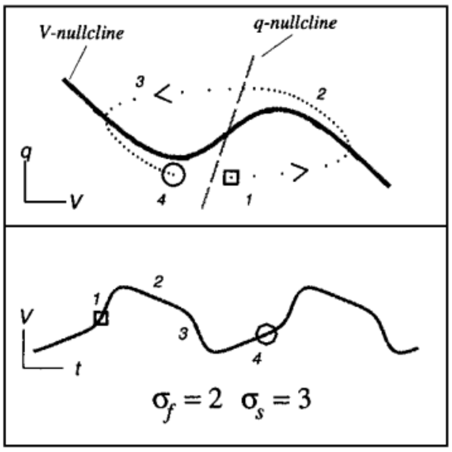


Figure 2.21: Phase plane for the cell model and an example for a rhythmic pattern. Extracted from [Rowat and Selverston, 1997].

With different values of the cell parameters, different intrinsic behaviors can be generated: quiescence (Q), almost an oscillator (A), endogenous oscillator (O), depolarization (D), hyperpolarization (H), and plateau (P), as shown in Figure 2.22.

Rowat & Selverston cell model were used in robotic locomotion problem in order to design (with genetic algorithms) neurocontrollers for various multi-legged robots [Hoinville, 2007]. This work showed that the RS neuron model is very well suited to generate adaptive rhythmic locomotion for legged robots because it may show properties of plasticity through its parameters. [Amrollah and Henaff, 2010] has implemented this cell model to control a two joint planar simulated leg that slips on a rail in order to show the role of sensory feedback on a CPG model to improve the locomotion task.

Our neural control architecture is based on Rowat & Selverston cell model in the generation of walking pattern in order to learn walking tasks on simulated planar biped in Chapter 4 and on a NAO humanoid robot in Chapter 5.

2.3.3 Legged Robot Locomotion

Various CPG architectures were proposed to achieve different locomotion tasks on legged robot locomotion. In this section, we present a few examples of these locomotive robots.

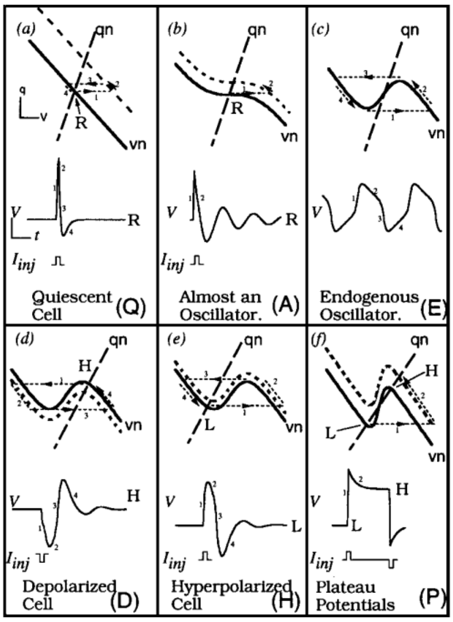


Figure 2.22: The six primitives patterns of the rhythm generator proposed by Rawat et al. Extracted from [Rowat and Selverston, 1997].

A Spinal Cord Model (Salamander) [Ijspeert et al., 2007] proposed a spinal cord model implemented on a salamandar robot. They demonstrated how the robot is able with such CPG model to switch between swimming and walking. The spinal cord model receives drive signals from the mesencephalic locomotor region located in the brainstem (see Figure 2.23). Like most CPG based locomotion, this model is based on some pragmatic rules or hypothesis. These rules ensure the generation of the patterns that produce traveling waves activated with a tonic drive for the switching from walking to swimming with increasing the drive signal from MLR.

Biped Locomotion based on Hopf Oscillator [Righetti and Ijspeert, 2006] present a system of coupled nonlinear oscillator to control the locomotion of a humanoid robot (see Figure 2.24).

This oscillator can learn the frequency of a periodic input signal by using this periodic input to perturb the state variable to converge into another state that corresponds to one of the frequency components of the periodic input. Therefore, the oscillator will be synchronized with the perturbation periodic input signal. After the convergence to the new state, the learnt frequency is encoded in the oscillator system. The formulation of the oscillator is as follows:

$$\dot{x} = \gamma(\mu - r^2)x - \omega y + \varepsilon F(t) \tag{2.28}$$

$$\dot{y} = \gamma(\mu - r^2)y - \omega x \tag{2.29}$$

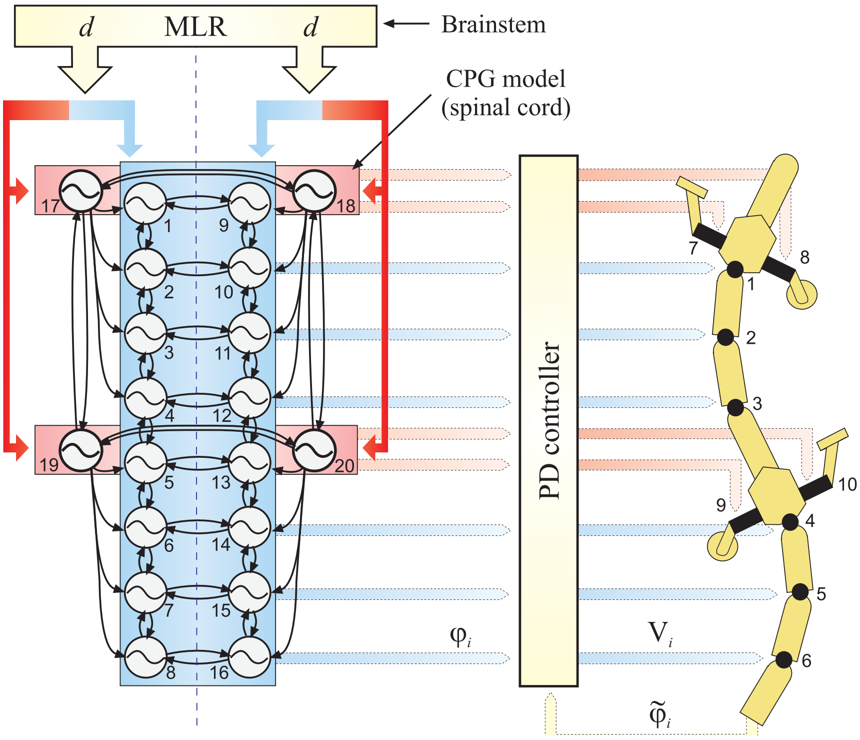


Figure 2.23: Configuration of the CPG model of the salamander robot, extracted from [Ijspeert et al., 2007].

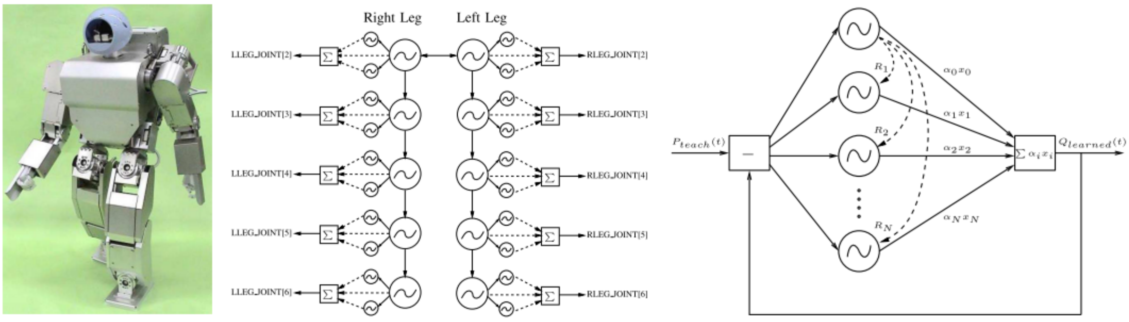


Figure 2.24: The Hoap-2 robot and the CPG structure for legs with the the structure of the network of adaptive Hopf oscillator, extracted from [Righetti and Ijspeert, 2006].

$$\dot{\omega} = -\varepsilon F(t)\frac{y}{r} \tag{2.30}$$

$$r = \sqrt{x^2 + y^2} \tag{2.31}$$

Where μ controls the oscillation's amplitude, ω represent the frequency, $F(t)$ is the periodic input signal that oscillator must adapt to. $\varepsilon > 0$ is a coupling constant.

Reflexive Walking Controller (RunBot) [Geng et al., 2005] built a reflexive controller based sensory motor connection in order to control the bipedal robot RunBot, see Figure 2.25. Without any rhythm generation this reflexive model based on some pragmatic rules is able to control the biped and achieve the walking task. In Chapter 3, the principles of this model are detailed, as we have adapted this model for our initial study on Success-Failure learning.

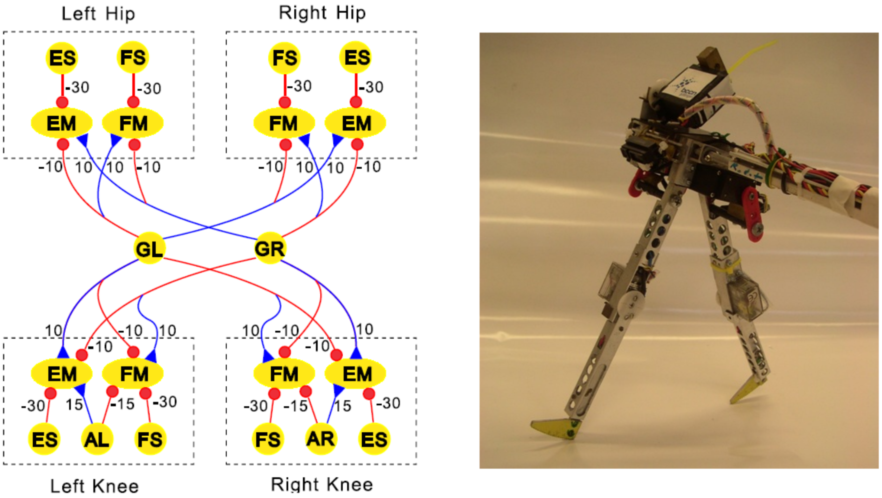


Figure 2.25: The Reflexive Neuronal controller for bipedal walking on the left. RunBot robot on the right, extracted from [Manoonpong et al., 2007].

To extend the walking capability on sloped terrains, a servo motor with fixed mass is fixed on the top of RunBot in order to compensate the slope effect by a high body control. IR sensor is used to detect the ramp and react for the postural reflex that is triggered and executed through the actuation of a servomotor responsible for this reflex. After a few trials the robot will be able to optimize the weights in the postural reflex sensory motor neurons by learning the synaptic weights in the sensory motor connections for this servo (see Figure 2.26).

Neural Oscillator for Biped Locomotion [Endo et al., 2004] propose a CPG model based on Matsuoka neural oscillator model to control planar robot with lateral movement constrained by a boom, see Figure 2.27.

The robustness of this controller and its adaptation capability are shown for walking on surfaces with different friction properties. The model decomposes walking into stepping motion in place and swing motion. Stepping motion is driven by Linear Motion oscillator (LM) (see Figure 2.27), while swing motion is driven by Swing Motion oscillator (SM). Walking motion is produced when linear motion and swing motion cooperate together with the proper phase. However, feedback pathway drives the oscillators to make the robot interact with the environment and stay in balance, e.g. in double support phase, the body pitch angle is fed back to

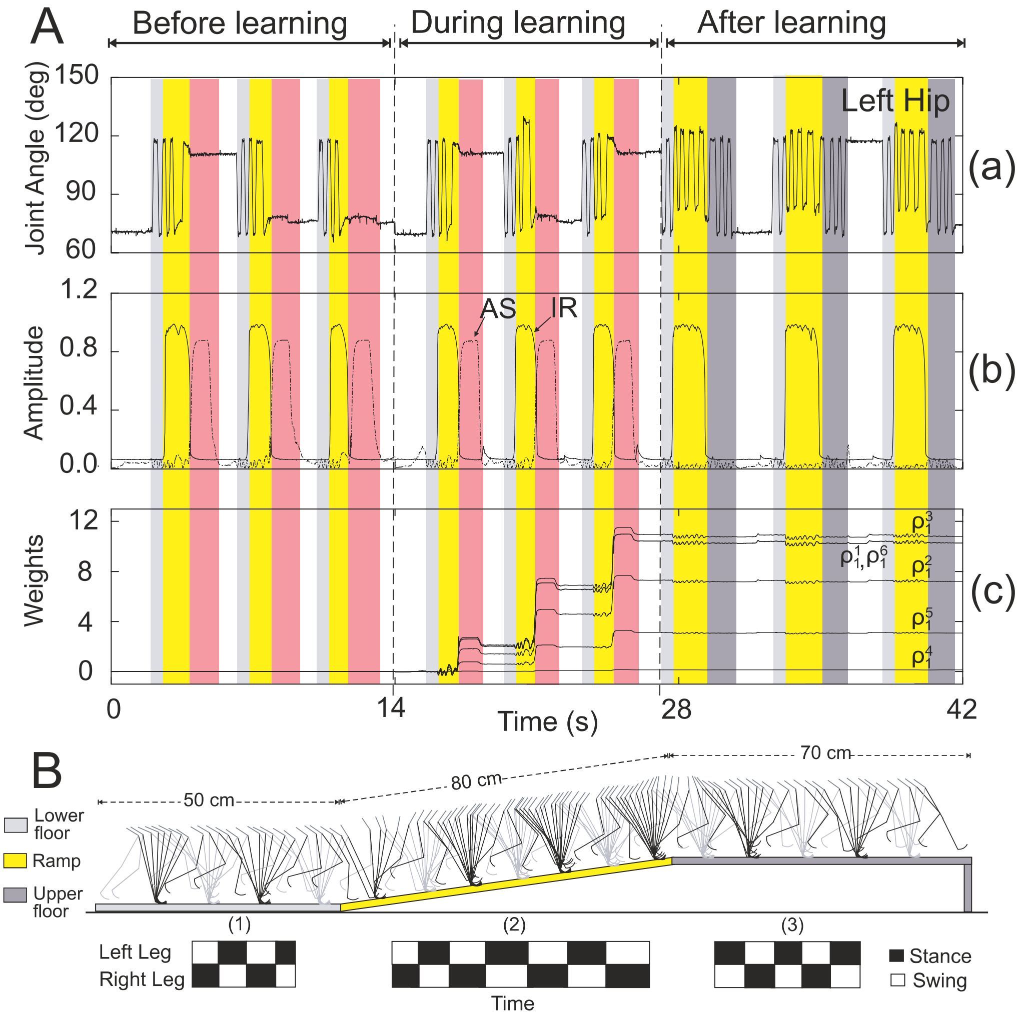


Figure 2.26: Adaptive walking experiment for RunBot robot. (A) left hip angle (a), IR sensor signal and stretch reflex sensor neuron AS (b), for learned synapses; comparing between its values before, during, and after learning (c). (B) A diagram of RunBot walking on different slopes. extracted from [Manoonpong et al., 2007].

LM oscillators in order to avoid falling by controlling lengths of the front and hind legs. In the experimental phase, a problem of symmetric motion appears: there is a difference in step length variation between left and right legs. This difference in step length is related to the initial condition. Walking efficiency is improved by investigating gait variation with changing an intrinsic constant c in the CPG mathematical model. c is a positive constant between 2 and 4 added to Matsuoka equation (see Equation 2.12).

As in [Manoonpong et al., 2007] neural connection and feedback pathway in [Endo et al., 2004] were investigated by empirical exploration.

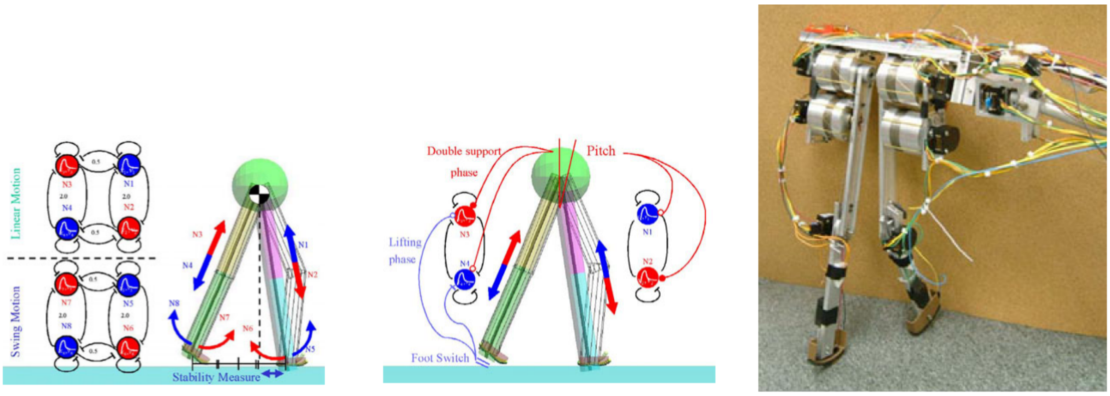


Figure 2.27: Neural oscillator arrangement in swing and linear motion and many sensory feedback for linear motion oscillators, extracted from [Endo et al., 2004].

In [Morimoto et al., 2006], authors modulate simple sinusoidal patterns for biped walking. Based on coupled oscillators, the phase of the desired joint trajectory was adjusted. The phase in lateral motion is detected by using the velocity and the location of the center of pressure. The oscillator model is described by:

$$\dot{\phi}_c = \omega_c + K_c.sin(\phi_r - \phi_c) \tag{2.32}$$

$$\dot{\phi}_r = \omega_r + K_r.sin(\phi_c - \phi_r) \tag{2.33}$$

Where ϕ_c is the phase of the biped controller, ϕ_r is the phase of the robot dynamics, ω_c is the frequency of the controller, ω_r is the natural frequency of the robot dynamics, K_c and K_r are positive constants for coupling circuitry.

The phase of the robot dynamics is calculated as

$$\phi_r(X) = -arctan\left(\frac{\dot{x}}{x}\right) \tag{2.34}$$

Where x and $\dot{x}$ are the position and the velocity of the center of pressure (see Figure 2.28.(a)). $X = (x, \dot{x}, \psi, \dot{\psi})$ is the state space of an equivalent inverted pendulum dynamic. ψ is the phase difference between oscillators ($\psi = \phi_r - \phi_c$).

Nominal trajectories for stepping motion were designed by a combination between a controller for side-to-side movement and a controller for foot clearance (see Figure 2.28(b)). An additional sinusoidal trajectory is designed for forward walking. Body roll angle is used to stabilize the controller in the ankle joint in side-to-side motion.

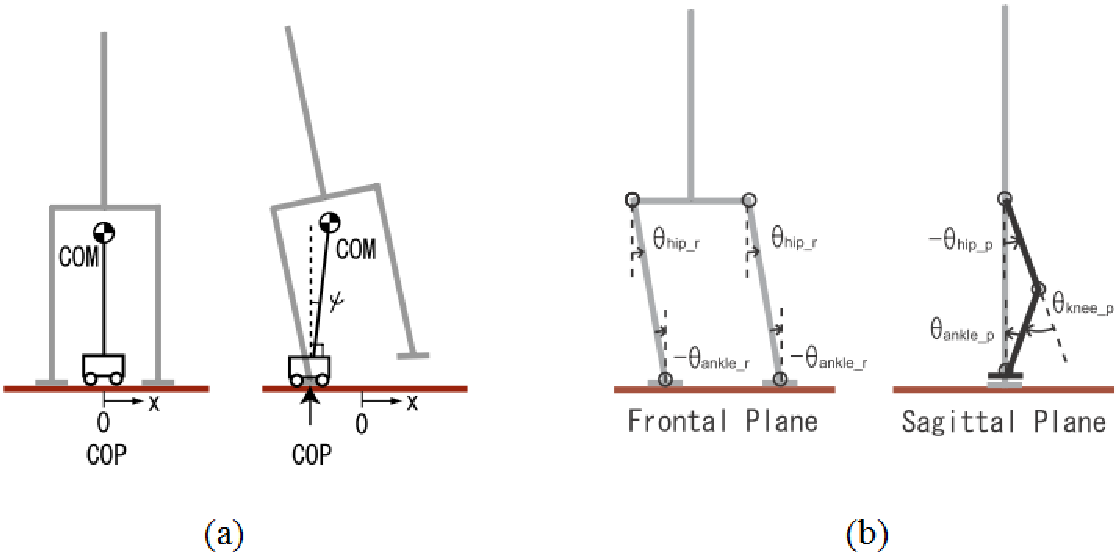


Figure 2.28: (a) Inverted pendulum model. (b) Stepping controller by side to side motion and foot clearance motion. Extracted from [Morimoto et al., 2006].

This controller was tested on a small humanoid robot developed by Sony and also on human sized hydraulic humanoid robot (developed by SARCOS) to generate stepping and walking patterns [Morimoto et al., 2006]. The robustness of the controller is also presented with different ground friction.

This study used a linear approximation to design stabilizing controllers for lateral movement. However, a nonlinear approximation model and optimization methods such as reinforcement learning are necessary to acquire a nonlinear feedback controller, which can stabilize sagittal and lateral movements.

2.4 Learning to Walk for Bipedal Robots

This section presents some of the learning techniques that have been developed specifically to address the important challenge of robot bipedal locomotion. In general, these learning techniques can be grouped into two classes: 1) supervised learning; and 2) reinforcement learning (RL).

Data to learn are always labeled with error signal in supervised learning, while they are labeled with reward in reinforcement learning techniques [Sutton and Barto, 1998]. Learning in the absence of these two feedback signals “Error / Reward” is called unsupervised. In all bipedal learning techniques an evaluation of the achieved task is indispensable to converge to an optimal solution.

Unsupervised learning builds a hidden structure for unlabeled data. There is no potential solution to evaluate due to the absence of error or reward signal. Approaches like clustering, dimensionality reducing feature extraction can be employed in unsupervised learning.

2.4.1 Neuronal-based Bipedal Supervised Learning

The Cerebellar Model Articulation Controller “CMAC”, proposed by [Albus, 1975] and described previously, was used in biped robot locomotion as an associative memory that stores patterns and to generate joint trajectories of the swing leg, see Figure 2.29 [Sabourin et al., 2006].

The control strategy has two stages: *learning* and *using* stages. In the first stage, a set of “pragmatic rules” is used to achieve a desired dynamic walking gait. The CMAC is used to learn the joints’ trajectories of the reference gait. In the second stage, the CMAC is used to generate the trajectories learned during the first stage.

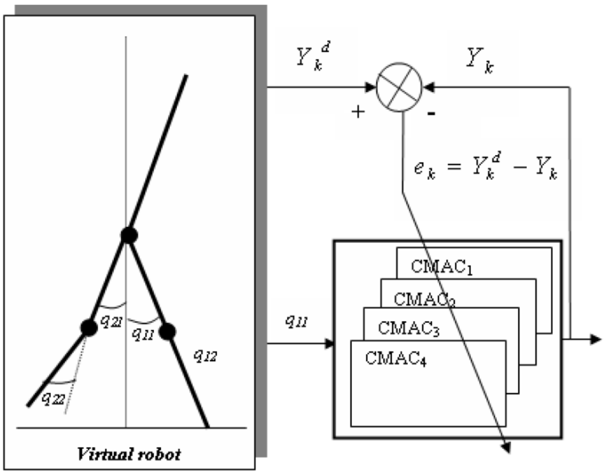


Figure 2.29: The walking control strategy for a RABBIT biped robot during training of CMAC. Extracted from [Sabourin et al., 2006].

It has been shown that CMAC can be used as a model-free function approximation. [Lin et al., 2006] introduced a technique of CMAC-based fault tolerant control to recover the nonlinear faults of the biped robot (see Figure 2.30). This system has two parts. The first is the fault estimation model where CMAC is used as online estimator that monitors and provides information about the off-nominal behavior due to the nonlinear faults. The other is the computed torque controller model.

In the previous studies, CMAC is used as an associative memory that can store joints trajectories or computed torques. Compared to Self-Organizing Maps, CMAC does not hold a topological structure that makes sense for neighborhood, which can be used to switch between memory cells. However, CMAC can be used as a memory not only for joints’ trajectories or torques but also for feedbacks. It can learn therefore to compensate the dynamic interaction between the robot and the environment, which is part of our future work.

2.4.2 Reinforcement Learning

Policy Gradient Method [Endo et al., 2008] propose a learning framework for CPG-based biped locomotion controller using a policy gradient method. They used

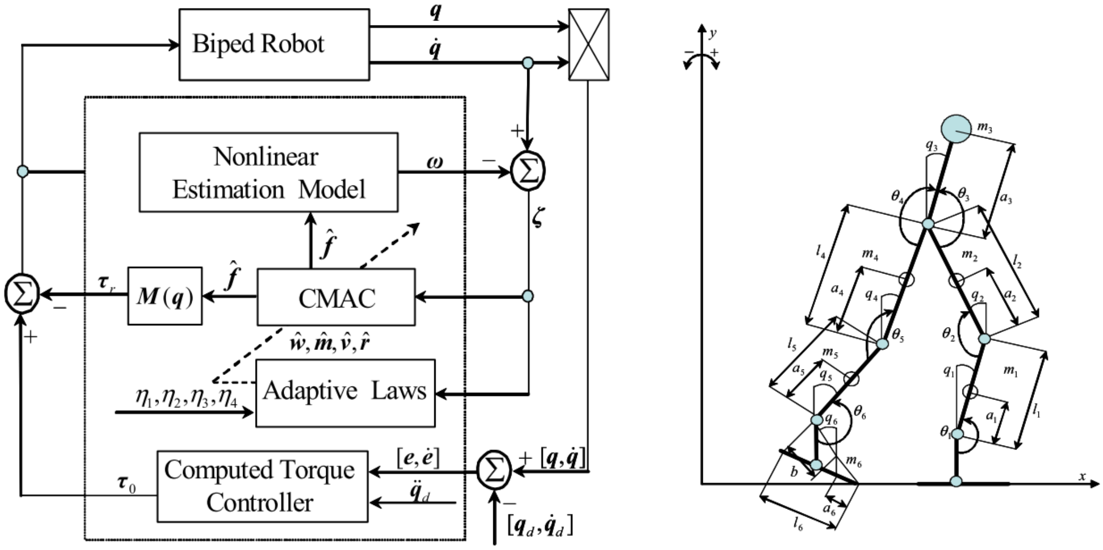


Figure 2.30: Architecture of CMAC fault tolerant control (left). Nine-link biped robot (right). Extracted from [Lin et al., 2006].

numerical simulation to acquire an appropriate feedback controller with few thousand of trials before transferring the result into the real robot. Typically, feedback pathway to neural oscillators are designed, while the parameter of the oscillators are tuned manually or by genetic algorithm [Lewis et al., 1992]. However, [Endo et al., 2008] propose a reinforcement learning model to optimize the CPG open parameters.

The basic framework for the CPG controller is presented in Figure 2.31(a). Using the robot sensory information, the CPG feedback controller generates feedback signal to the neural oscillators that control motors.

[Endo et al., 2008] separate the feedback controller into two parts: one is for oscillators responsible for motion in forward direction X; and the other is for oscillators responsible for motion in vertical direction Z. The first is optimized by reinforcement learning algorithm while the second is done with the inspiration of human walking as a biologically-inspired feedback (e.i. vestibulospinal reflex and extensor response reflex).

The vestibulospinal reflex is one of the basic posture-control of humans, where the contralateral muscles are activated by the vestibular system that measures the body's inclination in order to stabilize the upper body. [Kimura H, 2007] demonstrate the effectiveness of this feedback pathway with a quadruped robot. The extensor reflex, described by [Cohen and Boothe, 1999], explains the cat's stomping response when a vertical perturbation force is applied to its planter during extensor muscle activation. Such models for reflex are taken into account in our CPG design for biped robot, see Chapter 4.

Figure 2.31(c) represents the stepping in place motion control in the frontal plane. *Extensor response* [Cohen and Boothe, 1999] and *vestibulospinal reflex* are

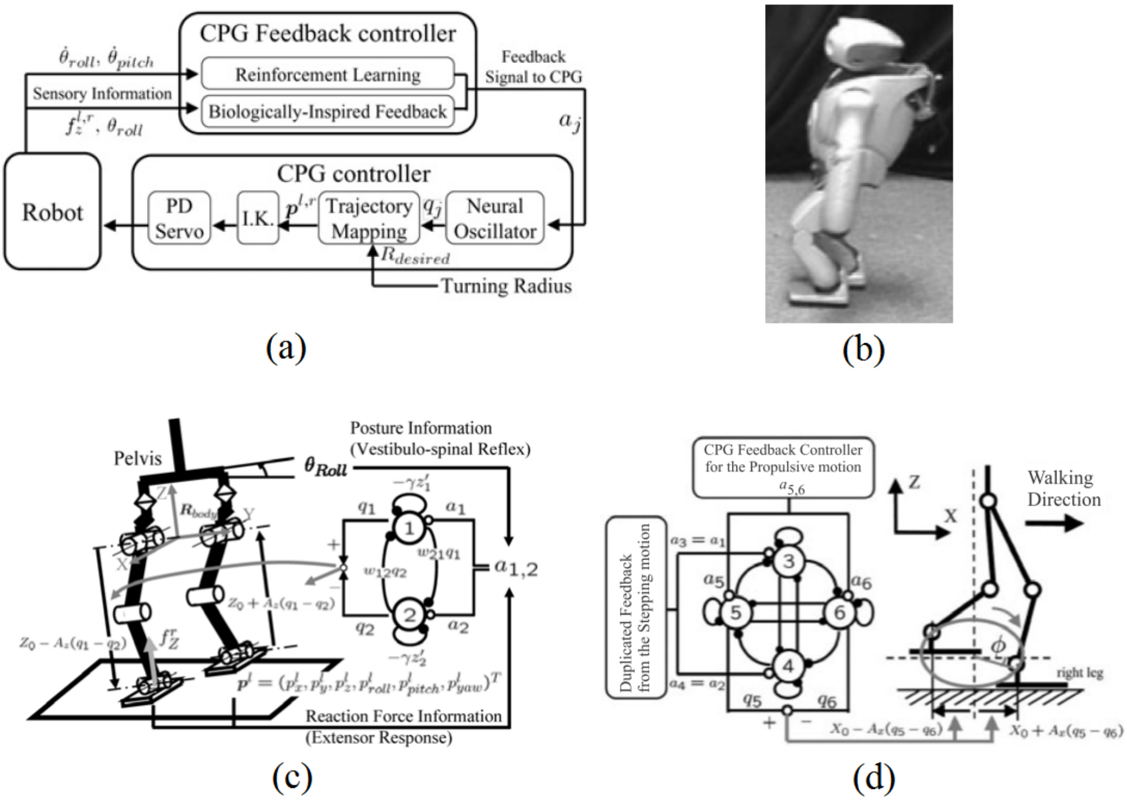


Figure 2.31: CPG control model. (a) General framework of the controller. (b) 3D full-body humanoid robot used in experiment. (c) Neural oscillator for stepping motion and biologically inspired feedback pathways. (d) Neural oscillators for sagittal plane motion. Figures were extracted from [Endo et al., 2008].

used as a biologically inspired feedback for the oscillators responsible for stepping motion. Quad-element neural oscillator controls the propulsive motion in the sagittal plane (Figure 2.31(d)). The objective is to make foot form an ellipsoidal trajectory.

In learning phase, a reasonable number of state variables were selected. The location of the COM is approximated to the position of the pelvis. The CPG generates leg trajectory in the direction of walking. As with any reinforcement learning technique, the reward information and the state of the robot will be sent back to the learning mechanism after the robot interaction with the environment (see Figure 2.32). Based on temporal difference error estimation in the learning of the value function, an actor generates the CPG feedback signal for the oscillators responsible of the motion in the walking direction.

Reward function is designed as

$$r(x) = k_H(h_1 - h_t) + k_S.v_x \tag{2.35}$$

Where h_1 and h_t are the pelvis height and its threshold. v_x is forward velocity. k_S and k_H are constants that represent the importance of height and speed factors

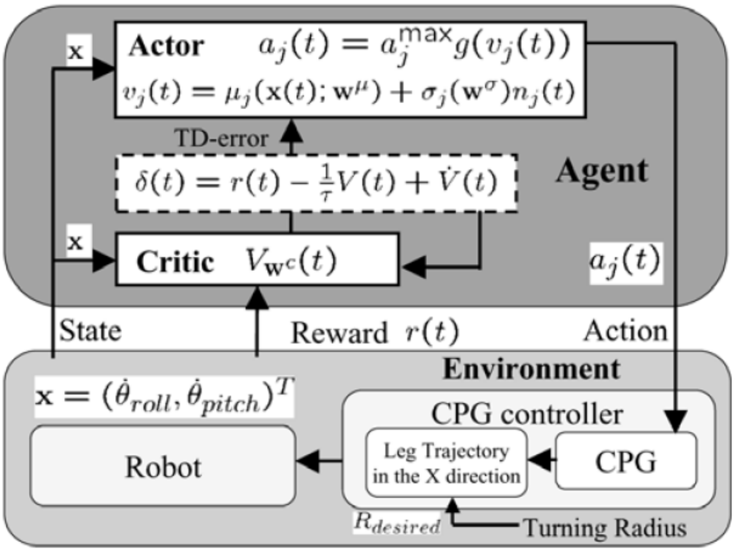

Actor $a_j(t) = a_j^{\max} g(v_j(t))$
$v_j(t) = \mu_j(\mathbf{x}(t); \mathbf{w}^{\mu}) + \sigma_j(\mathbf{w}^{\sigma}) n_j(t)$
x
TD-error
$\delta(t) = r(t) - \frac{1}{\tau}V(t) + \dot{V}(t)$
Agent
x
Critic $V_{\mathbf{w}^c}(t)$
$a_j(t)$
State
Reward $r(t)$
Action
Environment
$\mathbf{x} = (\dot{\theta}_{roll}, \dot{\theta}_{pitch})^T$
CPG controller
Robot
Leg Trajectory in the X direction
CPG
$R_{desired}$
Turning Radius

Figure 2.32: General diagram of CPG feedback learning, extracted from [Endo et al., 2008].

in the reward. The reward is designed to keep the height of the pelvis over the threshold h_t and achieve a forward motion. The parameters were predefined before learning, $h_t = 0.272$, k_S in the range $0.0 - 10.0$, and $k_H = 10$. In case of failure trial, the robot receives a negative reward as a punishment $r = -1$.

Authors propose an online learning algorithm that is able to adapt to environmental changes. With an initial policy, the controller was improved on a hardware robot with only 200 iterations. However, this model performs optimization only for the parameters responsible for the motion in the forward direction X, while the parameters responsible for the motion in the vertical direction Z were predefined. Furthermore, the reward function parameters were also predefined prior to the start of learning. Therefore, producing an online optimization method for all motion directions without any prior information about reward function parameters (e.g. gain, range, etc...) remains an important challenge toward autonomous robots.

Q-learning [Lee and Oh, 2009] use reinforcement learning in the generation of biped walking pattern (see Figure 2.33). Q-learning is used as a learning method and CMAC is used as a generalization method, the objective is to find walking patterns that satisfies both stable walking and the required position for foot placement. CMAC is used to store the various Q-value, which represent the actual experience or trained data.

States definition is based on the linear inverted pendulum model that was used widely to control the walking gaits of biped robots. Body position and acceleration were chosen as states, as the associated ZMP position can be used as a criterion for dynamic stability. The reward is divided as "fall down or not" and "how good it is"

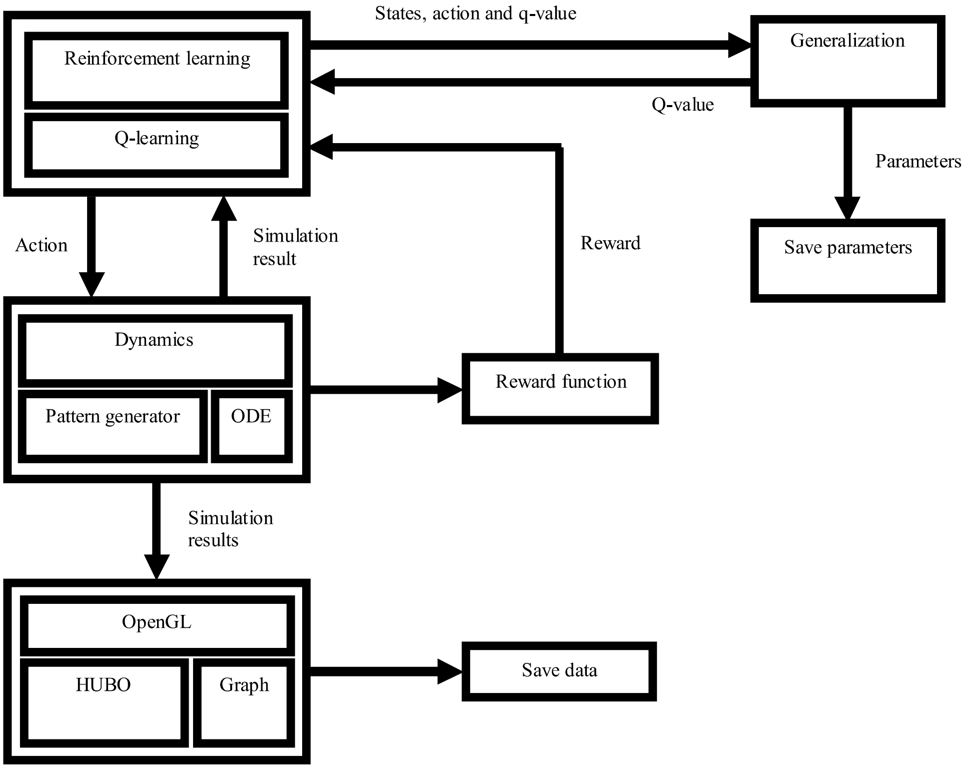


Figure 2.33: Walking pattern generation based on Q-learning, structure of the HUBO simulator. Extracted from [Lee and Oh, 2008].

according to the body rotation angle (less rotation much better).

To define the walking pattern and the wanted posture, initial and final positions and velocities of joints are selected as boundary conditions. This model was tested only in simulation on a planar biped, and requires knowing the desired posture in advance.

Ordinary Q-learning is generally considered for problems where state and action are discrete. Furthermore, Q-learning is used for problems where the target's reward is constant; it is always fixed before learning. In other word, this technique shows a limitation when the reward needs to be adaptive [Maeda, 2002]. This learning method is usually only for learning of a single task with a known target. However, in the case of robot learning, the ability to learn multiple targets simultaneously would be more desirable, especially interesting if it is based on the same learning framework (e.g. robot learning to optimize energy consumption and displacement velocity with same policy, learning a task under different conditions).

Predefinition in Reward Function [Li et al., 2011] propose the implementation of a fuzzy motion controller based on reinforcement learning (see Figure 2.34). The policy gradient RL (PGRL) searches the set of parameters for fast walking motion. The reward function is composed of two parts. First part represents the walking speed that was estimated by a vision system, while second part represents the desired Zero Moment Point (ZMP) trajectory.

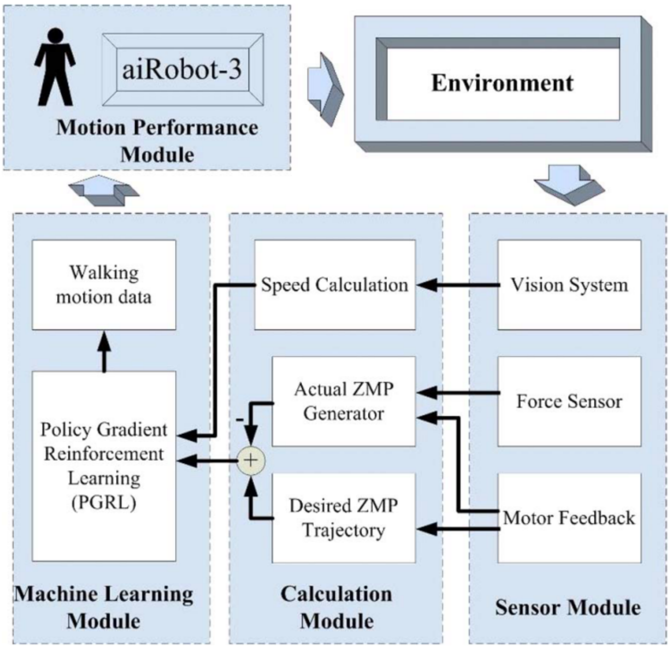


Figure 2.34: A policy gradient reinforcement learning (PGRL) controller, extracted from [Li et al., 2011].

Suppose that the zero moment point score in the reward function is Z_{score} and the velocity score is V_{score}. These scores are normalized by recording the maximum (V_M, Z_M) and the minimum (V_m, Z_m). Therefore, the reward function can be defined as follows:

$$\gamma_{score} = 100 \left(\alpha \left(\frac{V_{score} - V_m}{V_M - V_m} \right) + (1 - \alpha) \left(\frac{Z_{score} - Z_m}{Z_M - Z_m} \right) \right) \tag{2.36}$$

Where $\alpha \in [0, 1]$ defines the participation of the speed score and zero moment point score in the reward function. After learning, the controller succeeds in matching a stable and a fast walking pattern.

The drawback of a reward function definition lies in the predefinitions of the minimum and the maximum scores for walking speed and for the zero moment point. They cannot be estimated without experimentation on the robot. Thus, these parameters will need to be adjusted automatically and adapted during learning.

Monte Carlo Methods Monte Carlo method is a reinforcement learning algorithm that estimate value functions and discover optimal policies without requiring complete knowledge of the environment [Sutton and Barto, 1998]. Despite it was developed in 1949 by [Metropolis and Ulam, 1949], its variation tends to match particular steps, such as: defining the domain of possible input; random generation of input from a probability distribution, performing a deterministic computation; and gathering the results to approximate the policy.

Monte Carlo methods are widely used to estimate numerical quantities by repeated sampling. They were used to solve complicated optimization problems through randomized algorithms [D. P. Kroese and Botev, 2011, Rubinstein and Kroese., 2004]. However, this thesis proposes a searching algorithm that is able to do partially-randomized optimization, which significantly reduces optimization time in comparison with Monte Carlo methods (this will be shown in Chapter 5).

2.4.3 Evolutionary Computation

Evolutionary computation methods are widely used in robotics for parameters optimization [Eiben and Smith, 2008, Lipson et al., 2007, Ghiasi et al., 2010]. The common method of Evolutionary computation is the genetic algorithms (GAs) that generate solutions to optimization problems using techniques inspired by natural evolution [Koza, 1994]. They are likely to converge toward a global optimum compared with Policy Gradient techniques that converge to a local optimum [Rocha and Neves, 1999, Schmitt, 2004]; furthermore they can solve problems with multiple solutions [Tabandeh et al., 2006]. They are able to find a solution inside a high dimensional space with a requirement of huge number of iteration in simulation [Hase and Yamazaki, 1999].

A typical genetic algorithm requires a genetic representation of the solution domain and a fitness function to evaluate the solution domain. GAs provide the solution according to the fitness function that must be well described to measure the quality of the represented solution[Sivanandam, 2007]. It has been shown that GAs are able to find a solution for complex problems with a large number of parameters [Hase and Yamazaki, 1999, Inada and Ishii, 2003].

[Inada and Ishii, 2003] propose a CPG parameters searching method by genetic algorithm for behavior generation of bipedal robot (see Figure 2.35). The motion of human being and the trajectories of each joint are measured by motion captures. The CPG parameters are adjusted to generate the trajectories obtained by GA. Test was done first in simulation then on real robot.

Genetic algorithms work in general off-line, when a non-considered environmental change occurs during the using phase (after transfer the solution into the robot), an off-line readjustment of the parameters will be required without taking into account previous solutions.

2.5 Conclusion

Researches in biologically inspired robot locomotions are based on two parts, low level controller: “CPG”, and high level controller: “Cognition”. The high level with its cognitive cycles is able to perceive the environment and to react to surrounding events. Tuning the parameters of the “CPG” remains an important challenge in

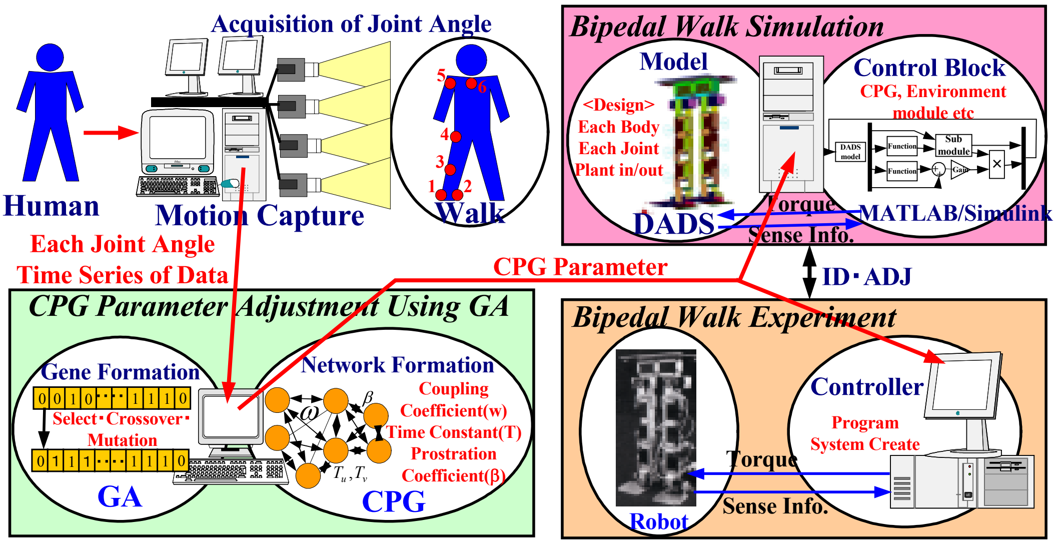


Figure 2.35: Behavior generation of bipedal robot using central pattern generator, CPG parameters searching method by genetic algorithm. Extracted from [Inada and Ishii, 2003].

order to have an auto-adaptive mechanism that is able to tune and readjust those parameters with external and internal changes.

Genetic Algorithms are widely used to tune low level controller parameters. However, it has limitations in robotics application where research is interested in the way to get the solution rather than the analytical solution in order to build auto-adaptive and autonomous robots, which can automatically adjust controller parameters in face of environments changes.

Policy Gradient Method is one of the most famous methods that were widely used in robotics and in walking control. This optimization technique guarantees the convergence at least to a local optimum, unlike other reinforcement learning search methods. However, the convergence to the global optimum is not guaranteed because it can depend on the initial condition.

Next chapters present the proposed searching mechanism that is inspired from human learning from mistakes. It is an on-line technique promised to be an auto-adaptive algorithm that can adjust the controller parameters in face of environment changes. Then a reward function model is proposed based on the inspiration from the orbitofrontal cortex (OFC) in coding reward adaptively. Our proposed reward model can solve the problem of predefinition of parameters in reward functions.

Chapter 3

Success-Failure Learning

Contents

Neurobiology studies showed that the role of the Anterior Cingulate Cortex of the brain is primarily responsible for avoiding repeated mistakes. According to vigilance threshold, which denotes the tolerance to risks, we can differentiate between a learning mechanism that takes risks, and one that averts risks. The tolerance to risk plays an important role in such learning mechanism. In this chapter, we propose a learning mechanism that is able to learn from negative and positive feedback. It is composed of two phases: i) an evaluation phase; and ii) a decision-making phase. In the evaluation phase, Self-Organizing Maps are used to represent success and failure. Decision-making is based on an early warning mechanism that enables a warning signal in order to avoid repeating past mistakes. Our approach is presented with an implementation on a simulated planar biped robot, controlled by a reflexive low-level neural controller. The learning system adapts the dynamics and range of a hip sensor neuron of the controller in order for the robot to walk on flat and slope terrain. Our results have shown the differences in learning capacity between risk-taking and risk avert behaviors. Results show that success and failure maps can learn better with a threshold that is more tolerant to risk. This gives rise to robustness to the controller even in the presence of slope terrain variations.

3.1 Introduction

Cognitive studies have identified an *early warning system* (EWS) in the human brain that can help to avoid making past mistakes again. It has been shown that the brain remembers details about past dangers [Singer et al., 2004]. Activities was founded in the Anterior Cingulate Cortex (ACC) after making mistakes [Brown and Braver, 2008]. This cortex area works as an early warning system that adjusts the current behavior to avoid dangerous situations. It responds not only to the sources of errors (external error feedback), but also to the earliest sources of error information available (internal error detection) [Mars et al., 2005]. It becomes active in proportion to the occurrence likelihood of an error [Gemba et al., 1986][Gehring et al., 1990][Hohnsbein et al., 1989]. Therefore, it can learn to identify situations where humans may make mistakes, and then help to avoid such situations from occuring [Brown and Braver, 2008]. It learns to predict error likelihood even for situations where no error occurs previously [Brown and Braver, 2005]. Through the observation of particular areas located in cerebral cortex in the brain responsible for cognitive control, neuropsychological studies demonstrated a switching in human learning strategies around the age of twelve years. This switch from learning with positive feedback to learning with negative feedback probably comes from the combination of brain maturing and experiences [Van Leijenhorst et al., 2008].

In this chapter, our aim is to produce an early warning mechanism that can help to avoid repeating past errors in the generation of walking patterns for humanoid robots. It is necessary for such a mechanism to have experience of mistakes and other experiences of success, in order to evaluate new situations before taking any decision and carrying out the test on the robot. This mechanism of selection allows estimation the zone of success and also the zone of conflict in the space of parameters. It is used to adapt the dynamics and range of a hip sensor neuron in a neural reflexive controller, proposed by F. Wörgötter [Geng et al., 2006] (see Section 2.3.3), for a simulated planar biped robot in order to avoid falls when the terrain slope varies.

The next section presents an Error Prediction Model (EPM) in the Anterior Cingulate Cortex [Brown and Braver, 2005]. Section 3.3 presents the principles of our learning mechanism in details, and introduces the concept of vigilance. Section 3.4 describes the neural reflexive controller based on sensor motor neurons proposed by F. Wörgötter. We show that this neural controller is able to generate a stable walk when its parameters are adjusted. In the Section 3.5, we show that this mechanism is able to detect the domain of viability of the controller and to allows the biped to walk on flat terrain and then on sloped terrain. We present therefore the interest of this method compared with others searching methods. In the conclusion of this chapter, improving this method will be discussed for both controllers, low and high level controllers.

3.2 Error Prediction Model in the Anterior Cingulate Cortex

The ACC and neighboring areas are involved in controlling and monitoring goal-direct-behavior to avoid repeating mistakes, [Brown and Braver, 2005]. They develop a computational model that shows how ACC not only detects errors, it may predict error likelihood before error occurs. The ACC is activated proportionally to the observed likelihood of the error. The error-likelihood hypothesis assumes the training signal that affects the ACC is acquired and dopaminergic. The phasic suppression of dopamine, which drives the error-related negativity (ERN) [Gehring et al., 1990][Gehring et al., 2012], may play the role of a training signal that make ACC activation stronger for contexts with more frequent error.

The computational neural model for error likelihood hypothesis, proposed by [Brown and Braver, 2005], take into account the development in ACC activation for error by experiences, see Figure 3.1.

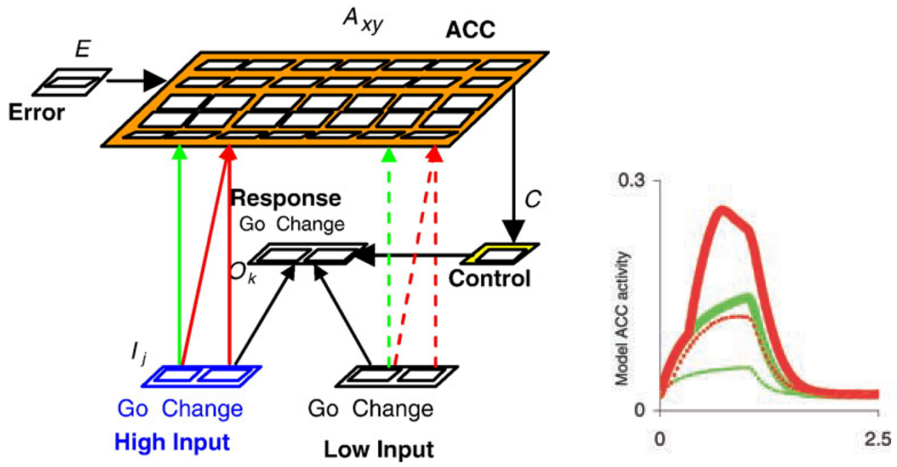


Figure 3.1: Computational model of error likelihood, extracted from [Brown and Braver, 2008].

This model is based on a neuroimaging study on the role of human ACC in expecting risk effects. It was tested by the incentive change signal task (ICST), where participants execute the task with monetary reward in case of successful trials. Tasks are achieved with four conditions in the variation of the error likelihood and the magnitude of the consequence of errors. As shown in Figure 3.2 each trial has four phases: color cue, target, response, and feedback.

In the first phase, a horizontal dash is displayed on the screen. The four possibilities with different colors are shown in Figure 3.2. Each color represents a combination between error likelihood and error consequence magnitude. In the second phase, an angle brace appears on the left or right side of dash to form an arrow that informs the subject in which direction the answer should be. For 33% of trials a change signal appears as an arrow pointing in the opposite direction that informs

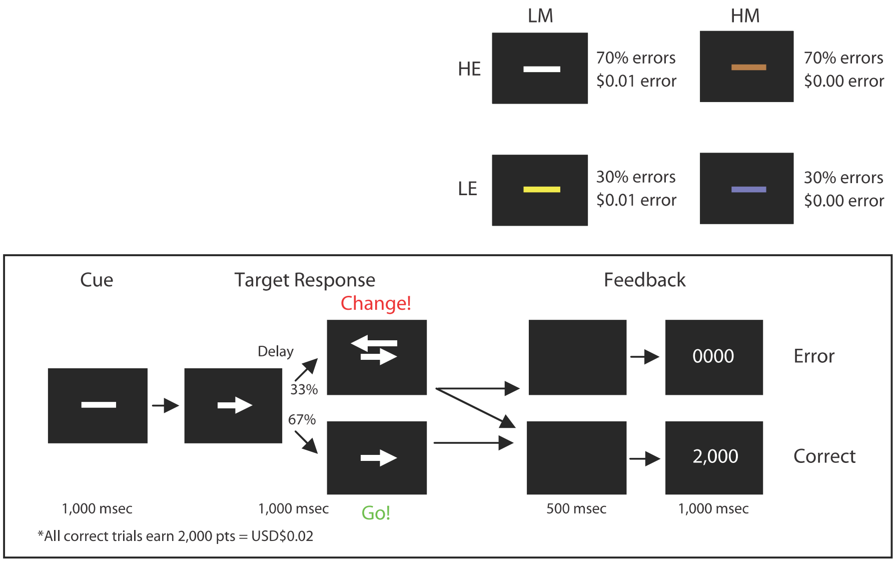


Figure 3.2: Incentive Change Signal Task. As a motivation to perform trials successfully , participants earned \$0.02 for each correct trial, nothing for incorrect trials in the condition of high error magnitude, and \$0.01 for an incorrect trial with low error magnitude. Error rate of 70% was used in high error likelihood condition and 30% for low error likelihood condition. (Left) extracted from [Brown and Braver, 2008]. (Right) extracted from [Brown and Braver, 2005].

the subject to hold back the response to the first arrow, and react with the pointing of the second arrow if possible. Delays between displaying the first and the second arrow, called change signal delays (CSDs), are adjusted differently for each color. CSDs are adjusted in function of committed errors in the rule that subjects achieve an error rate around 30% for low error likelihood conditions (yellow and blue), and around 70% for high error likelihood conditions (white and brown). After response deadline a blank screen for 0.5 second then visual feedback for gained points will be presented. As a result of FMRI observation of subjects' ACC, the ACC cells learn to respond with more activation for cues with high error likelihood, see Figure 3.1. The results suggest that the ACC is involved in cognitive control through its risk-related cortical activity.

We address this approach in learning locomotion task for humanoid robot (e.g. bipedal walking). The locomotion controller can be considered as two parts, low level controller and high level controller that can be responsible for mental process (see Section 3.5.1).

3.3 Learning Mechanism

The objectives of this learning mechanism is to adapt parameters of a low level controller and to detect its domain of viability, which is designated by Ω the state space of those influent parameters. The mechanism must be able to learn from negative feedback (failure) and positive feedback (success). Therefore it must have experience of success and other experience of failure in the state space Ω. As each action vector $\overrightarrow{v}$ from Ω leads to either success or failure, the mechanism will evaluate whether this vector belongs to a success case or to a failure case. The decision mechanism "go" or "no-go" described in [Matsumoto et al., 2003] works as an early warning system similar to that in the Anterior Cingulate Cortex [Brown and Braver, 2005, Brown and Braver, 2008]. The learning architecture is then based on these two mechanisms and works as shown in Figure 3.3.

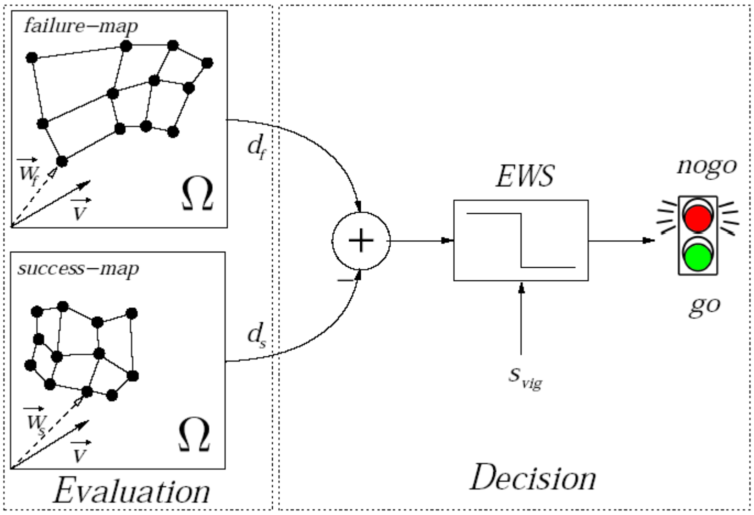


Figure 3.3: Success-Failure Learning mechanism with evaluation and decision phases.

3.3.1 Success-failure Evaluation

To represent the knowledge in success and in failure, we define two independent neural networks that are well-known Self Organizing Maps, proposed by Kohonen [Kohonen, 1984, Kohonen, 1995]. Success map S_m learns in case of success trials, and failure map F_m learns in case of failure trials. During the learning, the two maps will be self-organized in the state space that will be therefore divided into three zones: 1) a zone of success represented by success map; 2) a zone of failure represented by failure map; and 3) a zone of conflict that corresponds to the overlapping between the two maps. The evaluation of any vector $\overrightarrow{v}$ from space Ω belonging to success or failure is defined by the distance between $\overrightarrow{v}$ and each map. The distance of a vector with a map is the minimal euclidean norm between this vector and the closest, in the state space, neuron's weights vector (the winner neuron). For each $\overrightarrow{v}$ we have therefore two distances: one to S_m called d_s, and another to F_m called d_f. d_s and d_f are then used for the decision process.

3.3.2 Decision Mechanism

For a vector $\overrightarrow{v}$, the comparison between the distance with success map d_s and the distance with failure map d_f leads to an expected result in the case where the vector was passed to the low level controller (trial). According to expected results, if it may lead to failure, then an Early Warning Signal (EWS) becomes active to avoid the passing into the lower level controller, and the decision will be "no-go". When EWS is inactive the decision is "go". The decision mechanism is affected by the threshold of vigilance s_{vig}, which will be detailed in section 3.3.4.

3.3.3 Learning Algorithm

Success and failure maps represent the knowledge in success and in failure inside the state space. Maps will be initialized in the state space Ω. Then we take one vector $\overrightarrow{v}$ randomly from this space. In the phase of evaluation, we calculate the distance between this vector and all the neurons of both maps, (see Equation 3.1), where $\overrightarrow{d}_s^i$ is the distance between $\overrightarrow{v}$ and the i^{th} neuron in the success map, $\overrightarrow{w}_s^i$ is the weight vector of this neuron, $\overrightarrow{d}_f^i$ is the distance between $\overrightarrow{v}$ and the i^{th} neuron in the failure map and $\overrightarrow{w}_f^i$ is the weight vector of this neuron. For each map, the winner neuron corresponds to the smallest distance between $\overrightarrow{v}$ and the map, (see Equation 3.2), where d_s is the distance between $\overrightarrow{v}$ and success map while d_f is the distance with the failure map. In the decision phase, we compare d_s with d_f, by taking into account the threshold of vigilance s_{vig} (detailed in Section 3.3.4), which represents the tolerance to risks. If the threshold is higher than the difference between the distance to failure map and the distance to success map, the early warning signal becomes active, otherwise, this signal is inactive, see Equation 3.3.

The activation of EWS indicates that $\overrightarrow{v}$ will lead to failure if it is passed into the lower level. As maps are in the learning phase, it is possible that vector $\overrightarrow{v}$ can activate EWS at a time and inactivate it at another time, because the distances with the neurons change. A decision of "no-go"corresponds to active EWS and a decision of "go"corresponds to inactive EWS. In the case where decision is "no-go", we take another vector $\overrightarrow{v}$ randomly from Ω, then we look for expected results by evaluation and decision phases as detailed before. In case where decision is *go* ($\overrightarrow{v}$ may lead to success), the vector will be passed into the low level controller to run a trial.

There is a reward R for each trial, either negative (in case of failure) or positive (in case of success). Only one map learns $\overrightarrow{v}$. If the reward is negative the failure map learns, and if it is positive the success map learns.

Next, other vectors are randomly taken from Ω and execute the same steps until the convergence of maps. The convergence of the map occurs when any new vector

$\overrightarrow{v}$ will not cause a marked displacement of the neurons of this map in the parameter space. The displacement can be represented by the sum of weights changes squares for all the neurons of the map.

The flow diagram of the learning cycle is presented in Figure 3.5, and the following steps summarize it:

1. $\forall\ (S_m, F_m) \in \Omega$

2. $\forall\ \overrightarrow{v} \in \Omega$

   a) *Evaluation* :
   the distances to the neurons of the two maps:

$$\left\{ \begin{array}{l} \overrightarrow{d}_s^i = -\overrightarrow{w}_s^i + \overrightarrow{v} \\ \overrightarrow{d}_f^i = -\overrightarrow{w}_f^i + \overrightarrow{v} \end{array} \right. \tag{3.1}$$

   the distances to the neurons winners of the two maps:

$$\left\{ \begin{array}{l} d_s = min \parallel \overrightarrow{d}_s^i \parallel \\ d_f = min \parallel \overrightarrow{d}_f^i \parallel \end{array} \right. \tag{3.2}$$

   b) *Decision* :

$$EWS = \left\{ \begin{array}{lcr} 0 & (go) & if(d_f - d_s) > s_{vig} \\ 1 & (no - go) & otherwise \end{array} \right. \tag{3.3}$$

3. if $(no - go)$ go to 2 else if (go) test $\overrightarrow{v}$, and get a reward R

   if $(R : positive)$ learn S_m,

   else if $(R : negative)$ learn F_m,

   go to 2

In success-failure learning, the objective is to determine the cloud of success in the state space, success map can do this only by scanning all the space or by exploring the space around the succeeded trials. First solution is eliminated because the number of trials needed for scanning all the state space is huge. Failure map makes learning faster, because it avoids testing not only previously failed tested trials but also its surrounding areas. Even the training vector is randomly selected but the decision phase will reject it before the trial if it is incorporate to failure map area. As the state space is continuous, the vector will not be repeated, otherwise it will be needed to precise the "accuracy" for which we can judge that there is repetition for a previous tested vector.

3.3.4 Concept of Vigilance

Psychological research studies suggest that some people are more tolerant to risk than others who are more cautious [van Gelder et al., 2009][Pawlowski et al., 2008][Horvath and Zuckerman, 1993]. The vigilance is related to human learning in connection with decision making [Ahn and Picard, 2005]. In the standard psychological assessment of risk taking, people are classed as risk seeking or risk averse [Wang et al., 2007].

In our study for robot tasks learning by success and failure maps, we introduce the concept of vigilance in order to control the learning process in the two maps (success and failure) and manage the learning cycle while avoiding or taking risks according to the system's needs.

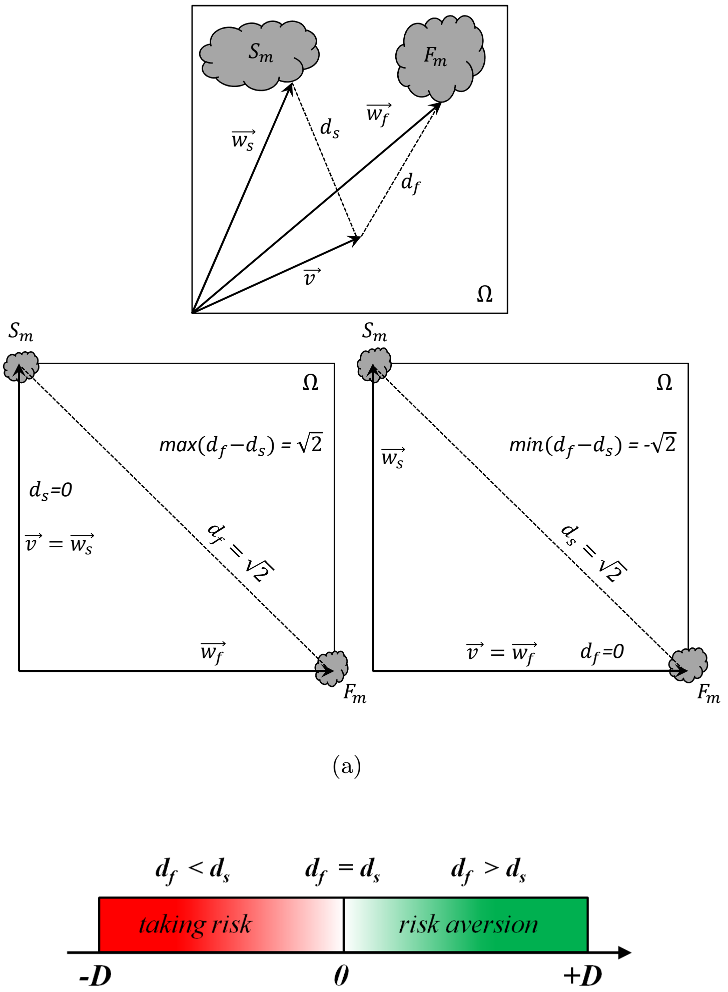
The vigilance is represented by a threshold s_{vig} that is used to adjust the early warning signal in the decision mechanism. This threshold describes the tolerance of risk, see Figure 3.3. By definition, the threshold of vigilance is the allowed margin of the difference between the distances of state space vector $\overrightarrow{v}$ with failure map (d_f) and with success map (d_s), for which the decision mechanism still responds with "go", see Equation 3.3. The threshold has a limited value according to the dimension of the state space.

As learning occurs inside a unit space (e.g. in a two dimensional state space, as in Figure 3.4(a)), the maximum difference between d_f and d_s is equal to the diameter of the unit space ($\sqrt{2}$ in Figure 3.4(a)), which corresponds to all $\overrightarrow{w}_s^i$ being in a corner and all $\overrightarrow{w}_f^i$ are in the opposite corner in the unit space, and $\overrightarrow{v}$ is closed to $\overrightarrow{w}_s^i$. The minimum difference between d_f and d_s corresponds to $\overrightarrow{w}_s^i$ for all success map neurons being in a corner and $\overrightarrow{w}_f^i$ for all failure map neurons and the randomly selected vector $\overrightarrow{v}$ are in the opposite corner. Therefore, the vigilance threshold $s_{vig} \in [-\sqrt{2}, +\sqrt{2}]$ in the two dimensional unit space, and $s_{vig} \in [-\sqrt{3}, +\sqrt{3}]$ in the three dimensional unit space. Therefore, as we move toward positive values of the threshold, the decision mechanism becomes more alert to risk (cautious). In the opposite, it has a tendency to take risks (courageous), see Figure 3.4(b), where D is the diameter of the space.

For instance, suppose that $s_{vig} = 0.1$, $d_s = 0.3$, and $d_f = 0.35$. In according to Equation 3.3, EWS become active and $\overrightarrow{v}$ will be rejected, and another vector will be selected, then the distances with the two maps will be measured. The randomly selected vector will then be tested on the robot when EWS is inactive.

In this chapter, we have fixed the threshold during learning. But we present the result for different values of the threshold.

In the next sections, this technique is used to learn intrinsic parameters of a low level controller for bipedal locomotion.



(a)

(b) The tolerance of risk.

Figure 3.4: (a) The distance to the neurons winner of success and failure maps. (b) The tolerance to risk.

3.4 Biological inspired neural controllers for walking

Biological inspired locomotion controllers are based on the simple circuit that is built from sensor neurons, motor neurons, and inter-neurons [Geng et al., 2006][Taga et al., 1991][Cruse et al., 1995][Wadden and Ekeberg, 1998]. Neurophysiological studies associate the rhythmic movement with the oscillation activity of a type of neurons, called neurons oscillators [McCrea and Rybak, 2008][Rowat and Selverston, 1991]. These oscillators can produce rhythmic activity without sensory input even without central input.

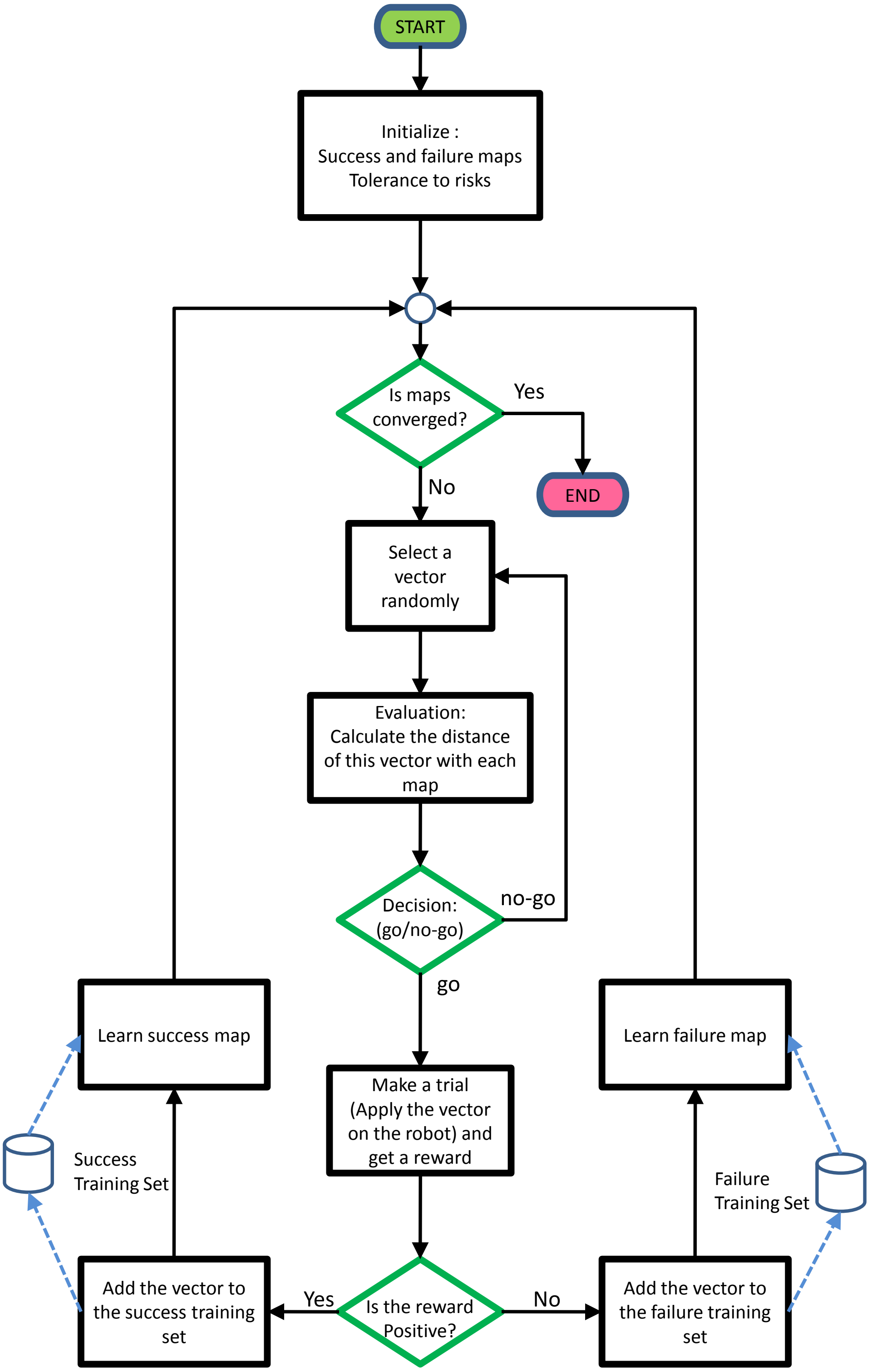


Figure 3.5: Flow diagram for success-failure learning.

But the sensory information is indispensable for walking because it allows to shape the rhythmic patterns in order to interact with the environment [Marder and Calabrese, 1996]. However, sensory information is mainly used to adapt the controller in front of changes and perturbations. Neurophysiologists have proved that biological controllers like Central Pattern Generators (CPG) have an adaptation mechanism that belongs to plasticity properties [McCrea and Rybak, 2008][Ishiguro et al., 2003]. Some robotics studies showed that plasticity allows the robot to adapt its rhythmic activity when environment changes [Ijspeert et al., 2007] [Hoinville et al., 2011].

To validate our proposed learning approach, we are interested in having the low level controller interact with the environment, like the neural reflexive controller proposed by [Geng et al., 2006] and tested on a real robot. This low level controller is based on the sensor motor approach. Our learning mechanism will regulate some parameters in this controller for walking, and to explore the domain of viability to give the ability of walking adaptation to the environment.

3.4.1 Neural model for Sensory-Motor Circuitry

In the neural model for sensory motor circuitry there are direct connections between neurons sensors and neurons motors, see Figure 3.6. A static model of sensor neuron was proposed by [Wadden and Ekeberg, 1998], it is described in the equation.3.4, where ρ_i is the activity of sensor neuron, α is a positive constant that denotes the dynamics of the neuron, θ is the amplitude and φ is the neuron input. φ can be an angular position, or a contact force [Geng et al., 2006]. In the other side, there is a model of motor neuron. Beer [Beer et al., 1992] has proposed a dynamic model that is described in equation.3.5 where y_j is the mean membrane potential of the j^{th} motor neuron, τ is a time constant, ρ_i is the activity of the i^{th} sensor neuron, w_{ij} is the synaptic weight between the i^{th} sensor neuron and the j^{th} motor neuron, u_j is the activity of this motor neuron, θ_m is the bias of this neuron. The neural reflexive controller is based on such neural model for sensory-motor circuitry.

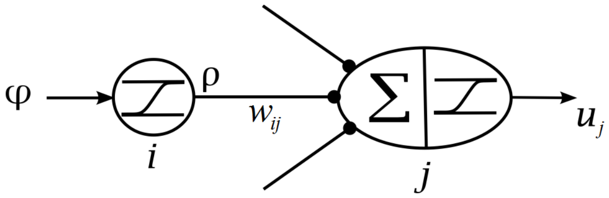


Figure 3.6: A neural model of sensory motor controller.

$$\rho_i = (1 + e^{\alpha(\theta-\varphi)})^{-1} \tag{3.4}$$

$$\begin{cases} \tau.\frac{dy_j}{dt} = -y_j + \sum_i w_{ij}.\rho_i \\ \\ u_j = (1 + e^{\alpha(\theta_m - y_j)})^{-1} \end{cases} \tag{3.5}$$

3.4.2 Neural Reflexive Controller

The neural architecture proposed by [Geng et al., 2006] to control a simulated biped is based on the sensory motor approach where sensor neurons are connected to extension and flexion motor neurons. Figure 3.7 shows the principles of this controller. A is a stretch receptor sensor neuron, G is a ground contact sensor neuron, FM is a flexion motor neuron, EM is an extension motor neuron. Lines with an arrow extremity indicate excitatory connections, and dotted lines terminated by a solid circle indicate inhibitory connections. Figure 3.7.(a) shows the interaction between the ground contact sensor neuron of the stance leg and the flexion and extension motor neurons in the same leg. Ground contact sensor neuron G in a leg excites the extension motor neuron EM in the knee and the flexion motor neuron FM in the hip of the same leg. Figure 3.7.(b) shows the interaction between ground contact sensor neuron and the flexion and extension motor neuron in the other leg. It excites the flexion motor neuron in the knee and the extension motor neuron in the hip. Figure 3.7.(c) shows the role of extension and flexion sensor neurons, E, F, to inhibit the corresponding motor neuron, which is the same for all joints. This behavior is referred as the articular reflex. Figure 3.7.(d) shows the role of the stretch receptor sensor neuron to excite the extension motor neuron in the knee of the same leg. This behavior is referred as the extension reflex.

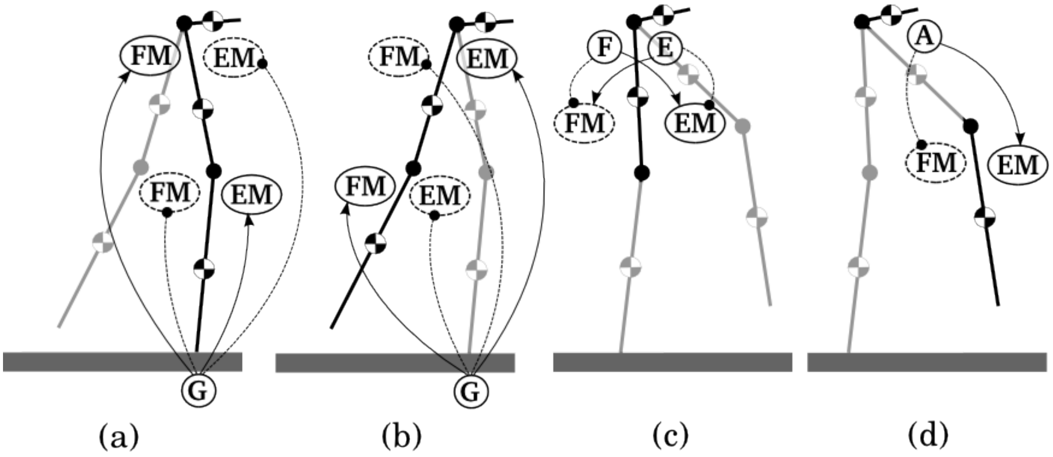


Figure 3.7: Principles of the neural reflexive controller proposed by Wörgötter.(a)Interaction with the stance leg. (b)Interaction with the swing leg. (c)Articular reflex. (d)Extension reflex.

The voltage of joint motor is done by

$$V = M(G_E.U_E + G_F.U_F) \tag{3.6}$$

Where V is input voltage of the motor, M is the servo amplification, U_E and U_F are the output of extensor and flexor motor neurons, G_E and G_F are the gains on motor neurons outputs.

In this study, we concentrate on two parameters of this low level controller. The first, α_{hip}, denotes the dynamics of rhythmic movement in the hip joint (dynamics of extensor sensor neuron). The second, $\theta_{hip-max}$, represents the amplitude of this movement (amplitude in the activity of extensor sensor neuron). By controlling these two parameters, the biped can walk and face environment changes, such as slope terrain variations.

3.4.3 Determination of viability domain of the neural controller

We have explored the domain of viability of the controller by varying the dynamics and the amplitude of the hip extensor sensor neuron (α_{hip} and $\theta_{hip-max}$) on a flat terrain. Inside a defined space for the two parameters, variations have been carried out with defined steps. For each couple (α_{hip},$\theta_{hip-max}$) the walking has been tested. According to definitions for success and failure we can know which couple leads to success or to failure. The biped has 10 seconds to walk, so if this time was passed and it was still standing, then it is a success. Otherwise, if it falls down before the time, it is a failure. In the simulation, we consider that the robot falls down when the gravity center of the trunk comes below the one of the two shanks. In such case the simulation will stop the trial. For all trials the robot has the same initial position in which one leg is in the stance phase and the other one is in the swing phase, because we are not interested here in the initial phase of walking. Figure 3.8 shows the results of this analytical studies related to flat terrain walk. The failure trials are represented by spots in surrounding area, while other spots represent the success trials.

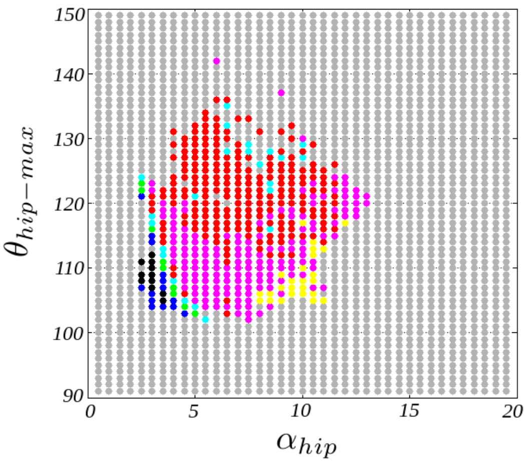


Figure 3.8: Domain of viability of the low level controller in space of α_{hip} and $\theta_{hip-max}$.

α_{hip} varies in [0 : 0.5 : 20], while $\theta_{hip-max}$ varies in [90° : 1 : 150°]. Walking velocity is limited in our case between $0.33[m/s]$ (black spots) and $0.66[m/s]$ (yellow spots). In the simulation, the walking velocity corresponds to the averaged velocity measured for the trunk.

3.5 Learn Walking and Adaptation Approach

First, we want the biped to learn walking on a flat terrain. The goal is to allow maps to explore the domain of viability in the state space. We will present the results for several values of vigilance threshold and discuss them. Second, we will present the results for a more complicated architecture devoted to learn how to walk on sloped terrain through an example that explains the adaptation approach for sloped terrain.

3.5.1 Learn Walking on Flat Terrain

We present how our learning approach makes success and failure maps to explore the space of parameters in order to find the domain of viability of the controller. The simulation is run for different values of vigilance threshold s_{vig}. Figure 3.9 shows the control diagram in case of learning on flat terrain. There are two loops of control: a low level control loop represented by the interaction between the biped and the sensory motor controller neuronal and a high level loop concerns the high level controller where the learning mechanism drives the low level controller and receives the result for each trial (success, failure).

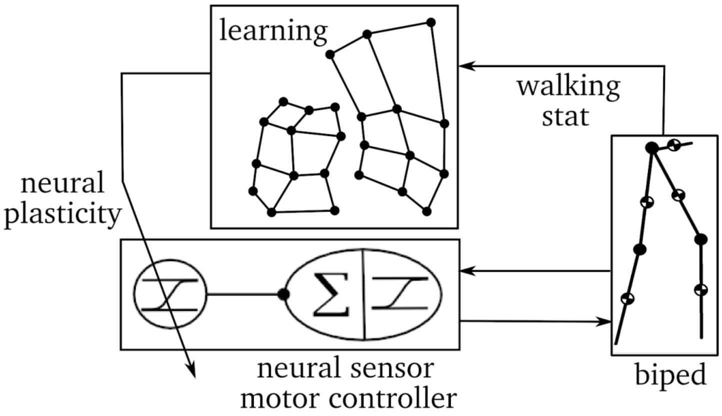


Figure 3.9: Walking control diagram, composed of two control levels: neural senror-motor controller (low level) and learning mechanism (high level).

In the learning algorithm, we initialize success-failure maps in the space of α_{hip} and $\theta_{hip-max}$. The same space has been used as studied previously for the domain of viability. The number of trials is fixed to 500 and the vigilance threshold determined. For a random vector $\overrightarrow{v}(\alpha_{hip}, \theta_{hip-max})$ from this space there are two processing phases, the evaluation phase and the decision phase. If the early warning signal EWS stays inactive for $\overrightarrow{v}$, then it may lead to success according to the past experience of this system represented by success map and failure map and also according to the risk tendency represented by vigilance threshold s_{vig}. Each vector that will lead to success has been passed to the controller sensory motor to run a trial on the biped. According to the result of each trial one map will learn, then another sampled vector $\overrightarrow{v}(\alpha_{hip}, \theta_{hip-max})$ from the space will be applied, and so on. After learning, all the vectors that had led to success have been incorporated into success map and all the vectors that had led to failure have been incorporated into failure map. The importance of this learning approach is double. First the maps themselves gain experience, second there is a specific technique of election that is used before each trial and controlled by the threshold of vigilance. Figures 3.10 to 3.12 show success map and failure map after learning, with 500 trials, for three different values of vigilance threshold. The state space in our studies is normalized between 0 and 1. Each map is composed of 100 neurons. Neuron weights values (w_1,w_2) denote a configuration of the low level controller ($w_1 = \alpha$, $w_2 = \theta_{hip-max}$). We have therefore 100 different configurations in each map that match 100 walking gaits stored in success map.

After learning with 500 trials, with $s_{vig} = 0.05$, we obtain 98% of succeeded trials, while 2% of failure. With another threshold $s_{vig} = 0$, we obtain 96% of succeeded trials, and 45% of success with $s_{vig} = -0.1$ and 28% of success with $s_{vig} = -0.2$. In the last case, as there is 72% failure the failure map was learned better than in the other cases.

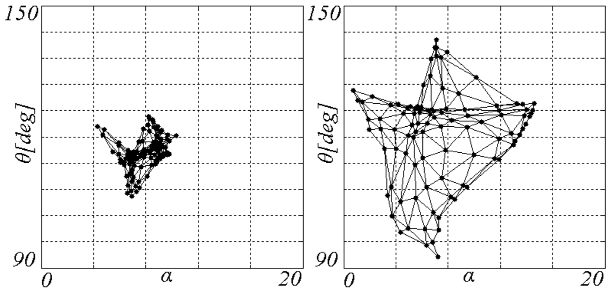
In Figures 3.10 to 3.12 all neurons in the success map lead to success (walk), but in the last map the domain of viability presented by the zone occupied by success map is bigger than in the other cases, which allows to have more stability and more walking gaits. So we can distinguish between two different behaviors for the system, risk taking and risk averse.

Thanks to the two behaviors the system can build experience in walking, and in case of risky behavior the system learns better. Figure 3.13 presents the rate of success as a function of vigilance threshold.

We can divide this figure into 3 zones. The first zone corresponds to $s_{vig} > 0.05$ where there is no decision, no trials, then no learning. The second zone corresponds to $s_{vig} < -0.4$, the system is more risky, and for a more negative threshold the decision will be *go* for all vectors. The middle zone is the most important, because it is a zone of switching between two different behaviors. In our studies we fixed the vigilance threshold during the learning phase, but changing this variable from a trial to another during learning is investigated in chapter 5.

3.5.2 Learning on Slope Terrain

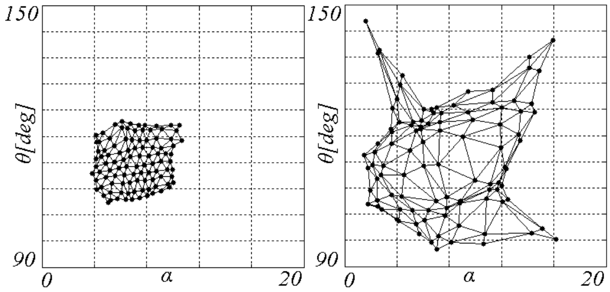
The objective from the previous study is to represent the zone of success in the state space by success map to justify the analytical study of the domain of viability. Our objective now is to generalize the controller for walking on sloped terrains. The modification in the maps structure consists of adding a third dimension to describe the terrain slope γ. Now the maps will learn in space of α_{hip}, $\theta_{hip-max}$ and γ. In



(a) Success map.

(b) Failure map.

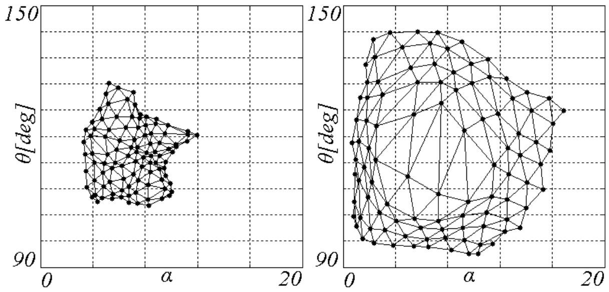
Figure 3.10: Success and failure maps after learning on flat terrain with vigilance threshold $s_{vig} = 0.05$.



(a) Success map.

(b) Failure map.

Figure 3.11: Success and failure maps after learning on flat terrain with vigilance threshold $s_{vig} = 0.0$.



(a) Success map.

(b) Failure map.

Figure 3.12: Success and failure maps after learning on flat terrain with vigilance threshold $s_{vig} = -0.2$.

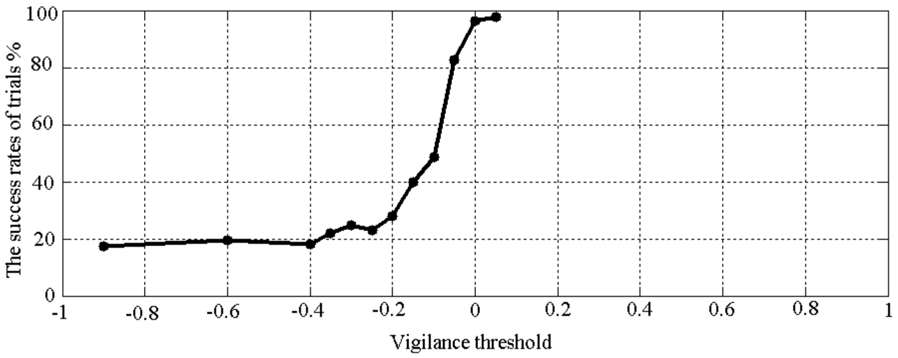


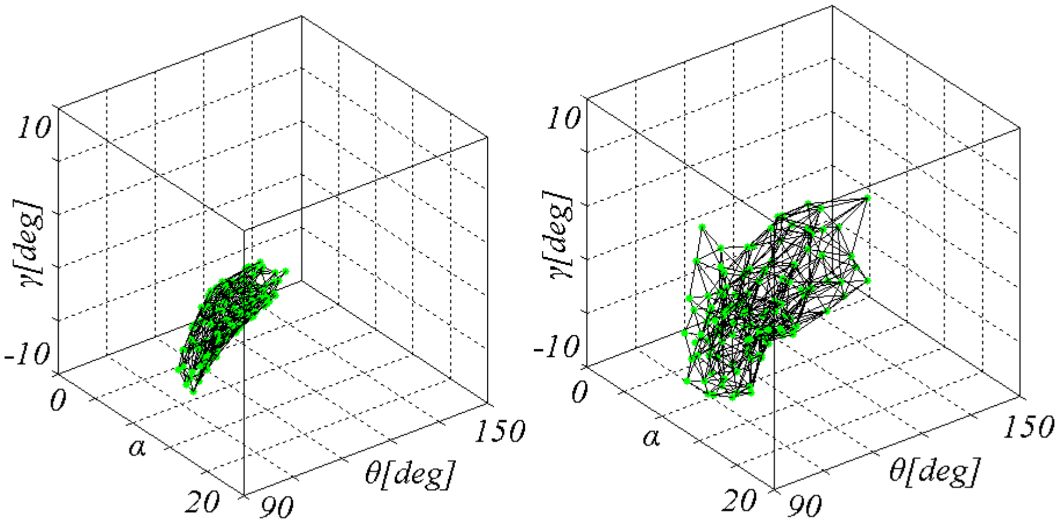
Figure 3.13: Rate of succeeded trials as a function of vigilance threshold.

our study the slope is limited between +10° and −10°. In the learning phase the biped learns to walk on terrains with different random slopes. After learning, the two SOM must be organized in the three dimension state space to represent success and failure experience. Figures 3.14 and 3.15 show success and failure maps after learning for different values of vigilance threshold.

Each map is composed of 125 neurons where each neuron has three weights (w_1,w_2,w_3) that denote a configuration of the low level controller ($w_1 = \alpha_{hip}$, $w_2 = \theta_{hip-max}$) for walking on determined sloped terrain ($w_3 = \gamma$). When $s_{vig} = 0$ there is a success in 86% of trials and a failure in 14%. Success and failure maps are shown in Figures 3.14(a) and 3.14(b) respectively. For the other value of vigilance, $s_{vig} = -0.2$, there is a success only in 15% of trials and a failure in 85%, as shown in Figures 3.15(a) and 3.15(b). The space occupied by success map in the second case is bigger than in the first case. This difference is referred to as the difference in the behavior according to vigilance threshold. As the failure rate in the second case is higher than in the first case, the failure map will learn better in the second case.

After learning, each neuron in the success map corresponds to a walking on a particular slope, including gait and speed. To walk on a terrain with a particular slope γ, a calculation occurs between all neurons to find the winner without taking the (w_1, w_2) values for neurons into account. The winner is the neuron whose w_3 is the closest to γ, while other weights of the neuron winner are used to configure the parameters (α_{hip}, $\theta_{hip-max}$) of the low level controller. Changing the terrain slope during walking causes switching into another neuron that corresponds to the new slope. This switch can be direct between the neurons or indirect by use of intermediary neurons. Figure 3.16 and Figure 3.17 show how the biped can walk on a sloped up and sloped down terrains with different configuration in success map neurons.

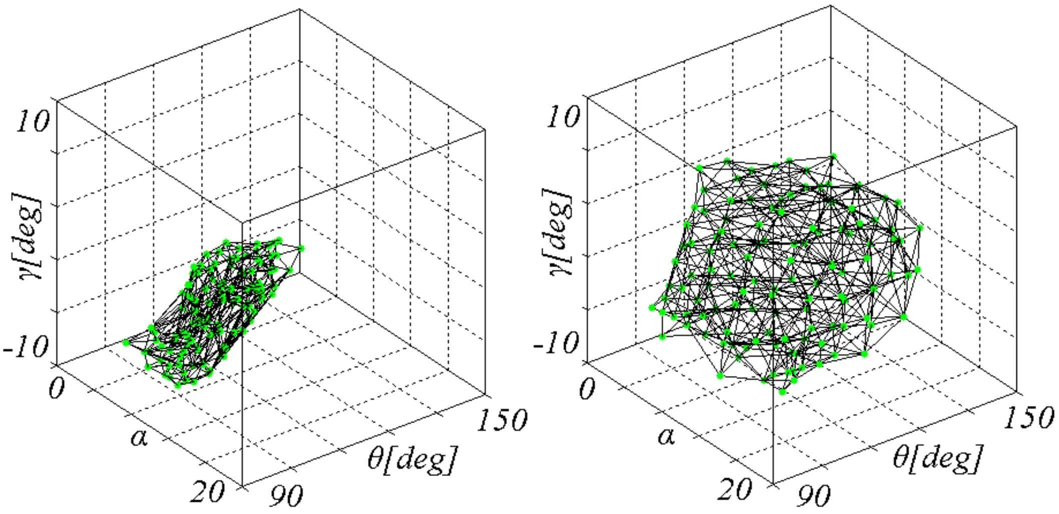
For any slope γ in the domain of viability (success map) there is a corresponding couple (α_{hip}, $\theta_{hip-max}$) that can be applied to the lower level of control to perform the walking, see Figure 3.18.



(a) Success map.

(b) Failure map.

Figure 3.14: Success and failure maps after learning on different terrain slopes with vigilance threshold $s_{vig} = 0.0$.



(a) Success map.

(b) Failure map.

Figure 3.15: Success and failure maps after learning on different terrain slopes with vigilance threshold $s_{vig} = -0.2$.

3.6 Discussion

The proposed algorithm can be regarded as a policy search method. Different searching methods has been proposed previously in reinforcement learning on autonomous robot controller [Grudic et al., 2003, Peters et al., 2003], see Section 2.4.2. Policy gradient method is one of the most famous, it was used widely in robotics and in walking controller [Endo et al., 2008, Li et al., 2011]. Policy Gradient Reinforcement Learning (PGRL) is an optimization technique that guarantees the convergence at least to a local optimum, unlike the other RL search methods. The convergence to the global optimum is not guaranteed because it depends on the initial condition.

Due to the randomly sampling before the decision phase and due to vigilance adaptation technique, the Success-Failure learning can guarantee the convergence to the successful clouds in the state space. Thanks to the evaluation phase, the decision phases, and the concept of vigilance, Success-Failure learning is a partially-random method (unlike Monte Carlo method, see Section 2.4.2), with properties of exploration and exploitation.

Evolutionary computation methods are also widely used in robotic for parameters optimization, see Section 2.4.3. The classical method of Evolutionary computation is the genetic algorithms (GAs) that generate solutions to optimization problems using techniques inspired by natural evolution. They are likely to converge toward a global optimum than PGRL techniques; furthermore it can solve problems with multiple solutions. However, it has limitations in robotics application where research is interested by the way to get the solution rather than the solution itself in order to build auto-adaptive and autonomous robots. GAs can provide only the solution according to the fitness function that must be well described. With Success-Failure learning, we are interested by the way to get the solution in order to build an auto-adaptive algorithm that can adjust the controller parameters in face of environments changes. Furthermore, Success-Failure learning is an online algorithm unlike most of the genetic algorithms.

3.7 Conclusion

In this chapter we presented a neuro-biologically inspired learning algorithm, it is considered as a policy searching method. The objectives of the mechanism were to learn from mistakes and to avoid making them again. This was done by building on experience of past mistakes and successes. We showed how these two experiences could build themselves through the stages of evaluation, decision and then trials. It can be said that the negative reward has importance as the positive. This mechanism was implemented on a planar biped; it allows the biped to learn walking without supervision. Thanks to switching between success map neurons that configure different slopes and walking velocity it added the property of adaptation even to changes of terrain slope.

The vigilance threshold that manages the switching between exploration and exploitation properties was selected by our experience. A model for an auto adjusted vigilance will be presented in Chapter 5.

This chapter discussed the walking task, where the feedback was even success or failure, which was used to learn the success map and the failure map. However, succeeded trials are not similar in term of efficiency. In Chapter 5 we will present the concept of adaptive reward that promise to evaluate the success.

Before improving the success-failure learning, we are looking to improve the low-level controller. The reflexive controller presented in this chapter can't generate motion patterns in the absence of sensory information, which is not the case in biological systems (e.g. cats ...). In the next chapter we introduce an efficient biologically-inspired locomotion controller that can show different motion patterns. Then, the benefits of success-failure learning will be presented to reduce the dimensionality based on patterns' energy. Such efficient low-level controller driven by success failure learning promise achieving locomotion tasks perfectly, especially when robustness is required for lower body tasks.

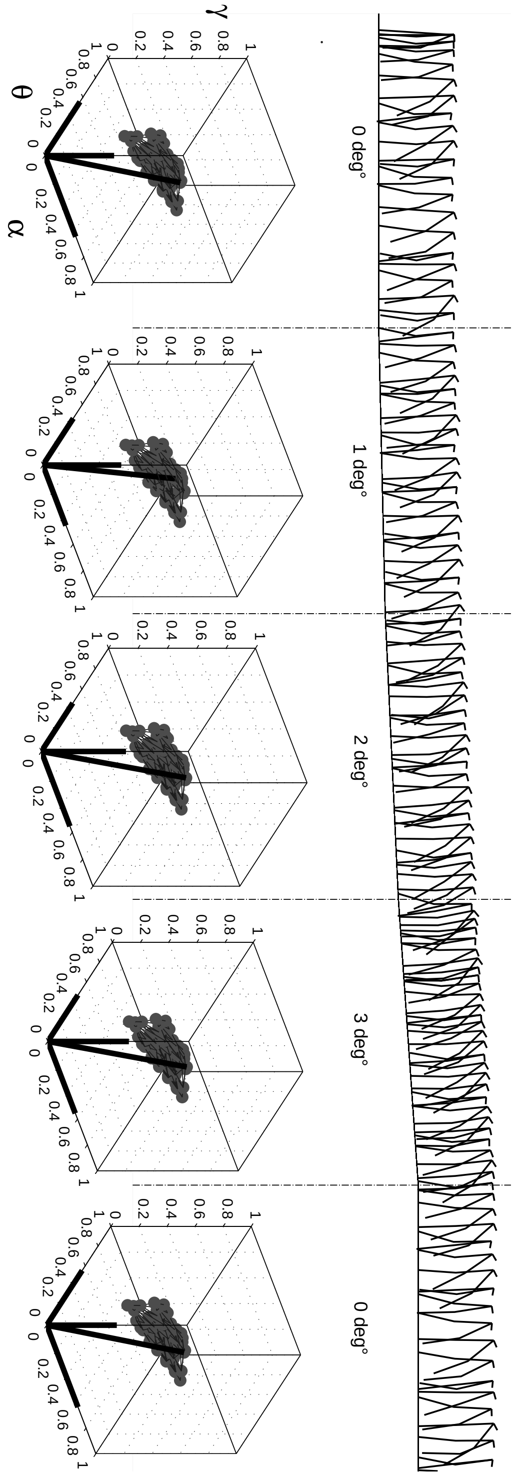


Figure 3.16: Simulation snapshots for walking on uphill terrain with corresponding neurons.

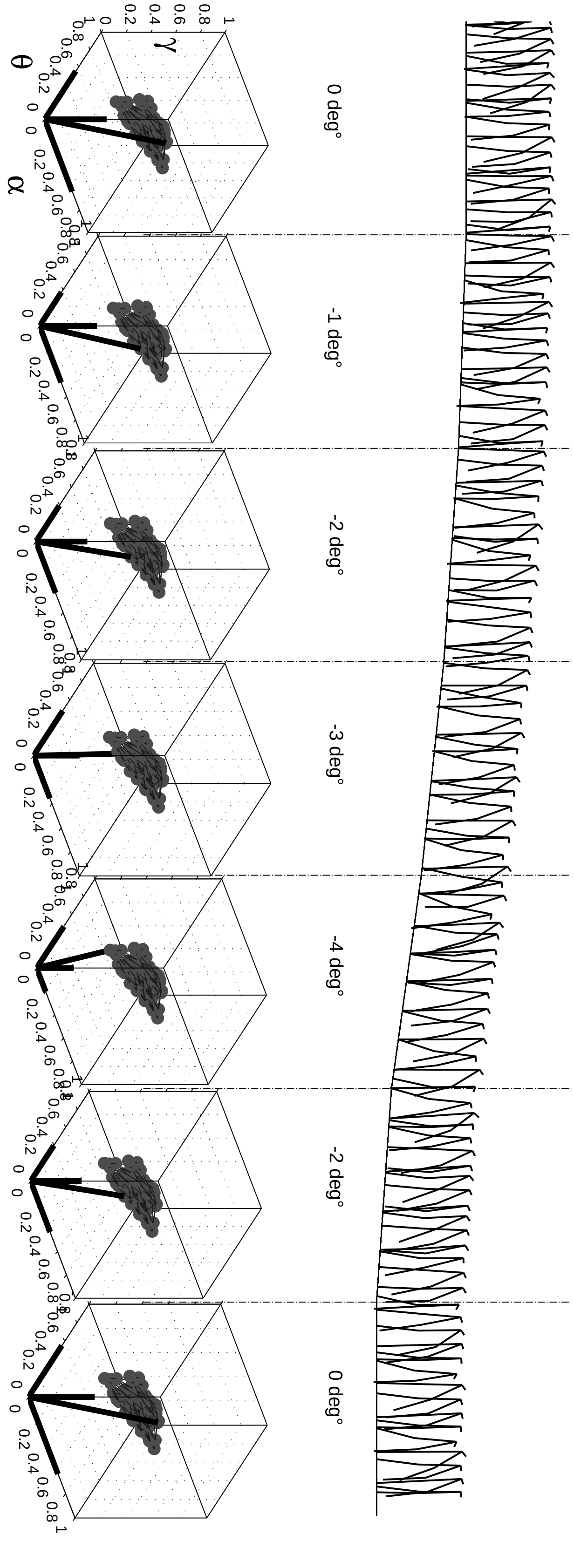
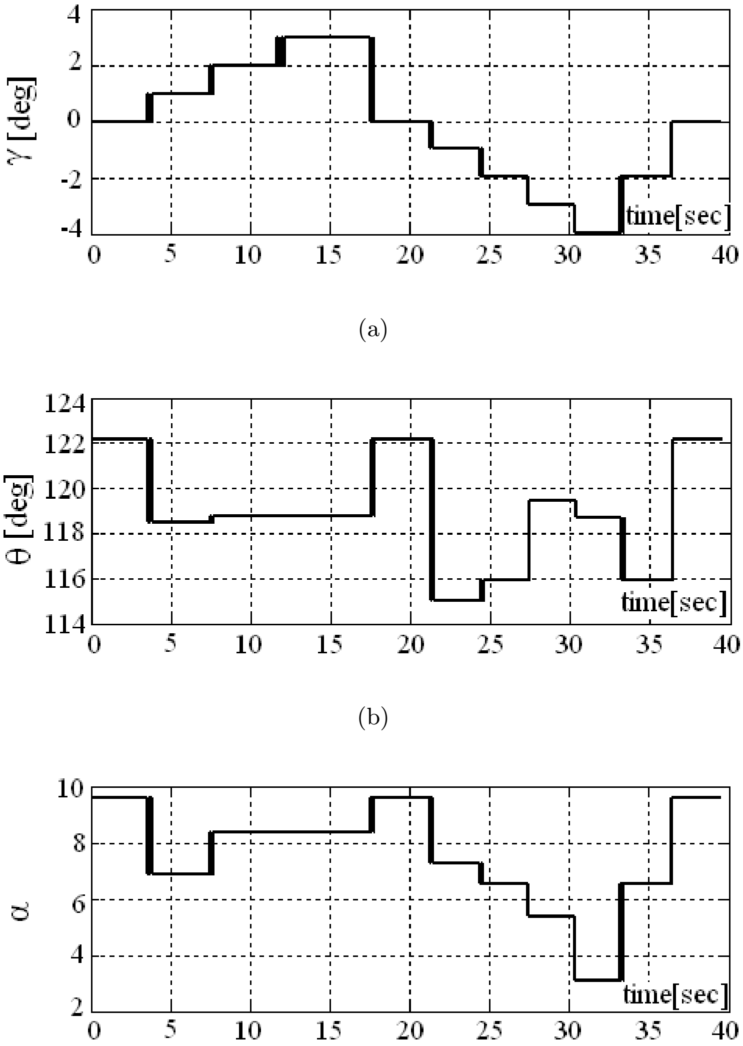


Figure 3.17: Simulation snapshots for walking on downhill terrain with corresponding neurons.



(a)

(b)

(c)

Figure 3.18: Switching between the neurons of success map during walking on irregular terrain. (a) represents the terrain slope, which is an input to the learning mechanism. (b) and (c) are the amplitude and the dynamics of the extensor sensor neuron, the outputs of the learning mechanism.

Chapter 4

Primitives Pattern Generation and Classification

Contents

In this chapter, we present an extended mathematical model of the Central Pattern Generator (CPG) in the spinal cord. The proposed CPG model is used as the underlying low-level controller of a humanoid robot to generate various walking patterns. Such a biological mechanism has shown to be highly robust in animal and in human locomotion. Our model is mainly supported by two neurophysiological studies. The first study identified a neural circuitry consisting of two-layered CPG, in which pattern formation and rhythm generation are produced at different levels. The second study focused on a specific neural model that can generate different patterns, including oscillation. This neural model was employed in the pattern generation layer of the CPG, which enables it to produce different motion patterns – periodic as well as aperiodic motions.

Motion patterns for the joint are classified into different classes according to a metrics, which reflects the kinetic energy of the joint. Due to the classification metrics, high-level control for action learning is introduced. For instance, an adaptive behavior of the rhythm generator neurons in the hip, the knee, and the ankle joints against external perturbation is shown to demonstrate the effectiveness of the proposed learning approach.

Due to pattern formation layer, the CPG is able to produce behaviors related to the dominating rhythm (extension/flexion) and explain, furthermore, the rhythm deletion without rhythm resetting behavior. The proposed multi-layered multi-pattern CPG model (MLMP-CPG) has been deployed on a 3D humanoid robot (NAO), while performing locomotion tasks. Simulations and further experimental results show high robustness of the walk under environment changes and even under a large range of disturbances.

4.1 Introduction

Biological studies suggest that animals' locomotion is mainly generated at the spinal cord level by neural circuitry called the central pattern generator (CPG) that may be affected by reflex circuits and adjustment signals from the brain [Orlovsky et al., 1999], [McCrea and Rybak, 2008], [Brown, 1911]. These studies were taken into account in the implementation of robots' locomotion algorithms, [Ijspeert, 2008], [Taga et al., 1991], [Kimura et al., 1999], [Endo et al., 2008], [Morimoto et al., 2008]. Biologically inspired walking mechanisms for legged robots do not require a perfect knowledge of the robots' kinematics and dynamics. Different models of neural oscillators were widely used to generate rhythmic motions [Matsuoka, 1985], [McMillen et al., 1999], [Ludovic et al., 2009], [Righetti et al., 2006], [Nakanishi et al., 2004]. The oscillatory pattern is generated by two mutually inhibiting neurons (e.g. Matsuoka [Matsuoka, 1985]). Rowat and Selverston [Rowat and Selverston, 1991] proposed a model of rhythmic neuron that can generate different types of patterns such as oscillatory ones. The different behaviors in the activity of these neurons can be used in robot's locomotion to achieve different tasks. However, complex tasks like walking, hopping, running, and obstacle avoidance, require correct synchronization and switching between the patterns [Ivanenko et al., 2007]. In the action learning approach, where learning always occurs in the space of parameters, there is a limitation to learn complex tasks, due to the dimension of this space which can drastically increase. This issue can be solved by looking for a new representation of patterns. Instead of learning in the space of parameters, learning can occur inside a new space called patterns' space (e.g. in the case of one dimensional patterns space, patterns will be represented only with one axis).

In this chapter, we produce a biological inspired neural controller for biped walking, based on two neurophysiological studies [Rybak et al., 2006, Rowat and Selverston, 1991], see Figure 4.1. The proposed multi-layered multi-pattern CPG model (MLMP-CPG) has been deployed on a 3D humanoid robot (NAO), while performing locomotion tasks. To deal with environment changes, the adaptation of the neurons behaviors will be introduced. Therefore, a new space for joints motion patterns will be proposed.

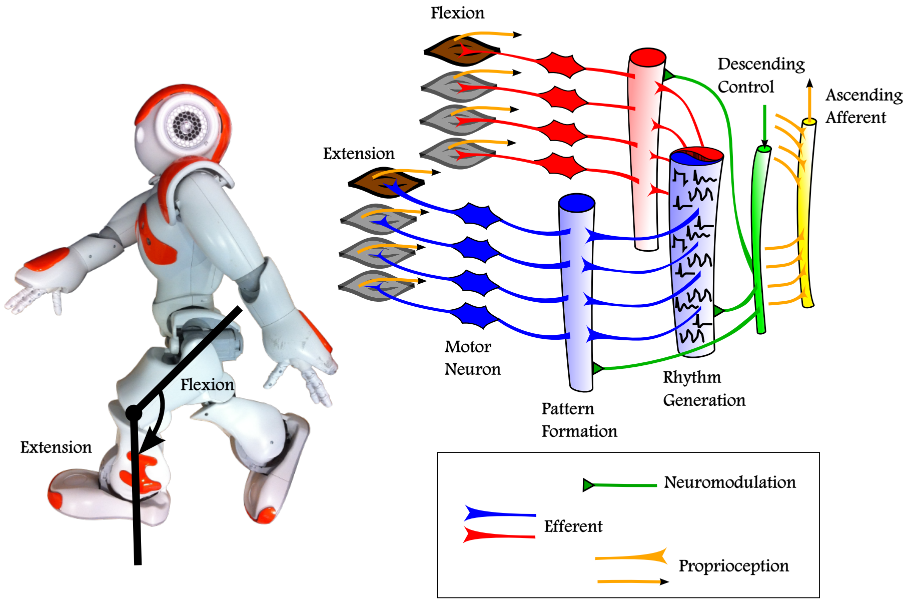


Figure 4.1: Conceptual overview of the multi-layered multi-pattern CPG for humanoid robot locomotion.

This chapter is organized as follows: Section 4.2 describes the architecture and the neural model of each layer of the CPG. The observed behaviors of the CPG at the joint level are presented in Section 4.3. Section 4.4 details the entire architecture for humanoid robot locomotion. This was first simulated with a planar biped, then it was validated on the Nao humanoid robot. The robustness of the proposed model for walking on sloped terrain and also in reaction against external perturbation force is also presented. A new representation of successful and failure walking patterns is proposed. This approach allows a high level control in the space of patterns instead of the space of parameters. Our learning scheme allows switching between bipedal patterns to achieve different locomotive tasks, and its effectiveness will be demonstrated. The stability of the central pattern generator has been studied using Poincaré maps.

4.2 Biological inspired CPG Model

Physiological studies suggest that rhythmic movements in animal's locomotion systems are produced by a neural network called CPG [Marder and Calabrese, 1996]. It can generate a locomotive rhythmic behavior with neither sensory nor central inputs [Kuo, 2002]. Sensory inputs shape the output of this locomotion system, and allow the animal to adapt its locomotion patterns to external or internal changes. Genetic studies on newborn rat and mice suggest that rhythmic limbs movements during locomotion are generated by neuronal networks located within the spinal cord [Kiehn and Butt, 2003]. Matsuoka and McMillen neural oscillators are widely used as mathematical models for non-linear oscillators [Matsuoka, 1985], [McMillen et al., 1999]. These half-center oscillators consist of two neurons that individually have no rhythmic behavior, but which produce rhythmic outputs when they are reciprocally coupled. This chapter describes another model of non-linear rhythm generator. This model is such that one neuron can generate not only oscillatory but also different motor patterns [Rowat and Selverston, 1991].

4.2.1 CPG Architecture

Similar to that in biological systems, the MLMP-CPG receives tonic drive from the high-level controller (for instance, supraspinal locomotion centers in mammalians) [Markin et al., 2010]. The mesencephalic locomotor region (MLR) is one of the locomotion control centers in the brainstem and was discovered in cats by Shik *et al.* [Shik et al., 1966]. The descending drive signal from MLR allows the CPG to generate basic locomotion behaviors by providing alternating activation of the corresponding motoneurons that drive joint actuators (e.g. extensor and flexor muscles). Figure 4.2 shows the proposed basic CPG wiring diagram for one robot joint after merging two neurophysiological studies [Rybak et al., 2006, Rowat and Selverston, 1991]. The architecture of the CPG is inspired from [Rybak et al., 2006] that separates the locomotion cycle timing and activation, while the neural model responsible for the generated behavior is proposed by [Rowat and Selverston, 1991].

This CPG is separated into three layers: Rhythm Generation neurons (RG), Pattern Formation neurons (PF) and Motor Neurons (MN). the activity of these neurons in the CPG is shaped by sensory neurons (feedback). The following subsections introduce each of these layers.

4.2.2 Model of Rhythmic Generator Neurons

In animal locomotion, the control of muscle-skeletal system is ensured in a hierarchical manner by a high-level nervous system and spinal cord nervous system (CPG) for rhythmic locomotion patterns. Several mathematical models have been proposed for rhythmic pattern generation (like, Matsuoka [Matsuoka, 1985], Taga [Taga et al., 1991], Rowat & Selverston [Rowat and Selverston, 1991]) and some of them were implemented in legged robots locomotion [Endo et al., 2004, Righetti and Ijspeert, 2006, Ijspeert et al., 2007, Endo et al., 2008].

Among many models of motor pattern generator, Matsuoka oscillator is widely used in robotics research. Matsuoka model is based on the mutual inhibition of two neurons with self-inhibition effect. Due to these connections, each neuron can produce rhythmic activity. Taga *et al.* used the Matsuoka oscillator like a neural rhythm generator to control the locomotion of a biped robot in a 2D simulated environment [Taga et al., 1991]. Their controller is based on the Matsuoka model as a unit oscillator which produces a torque to be applied on a specific joint. Matsuoka model has been widely used in legged robots locomotion to generate walking

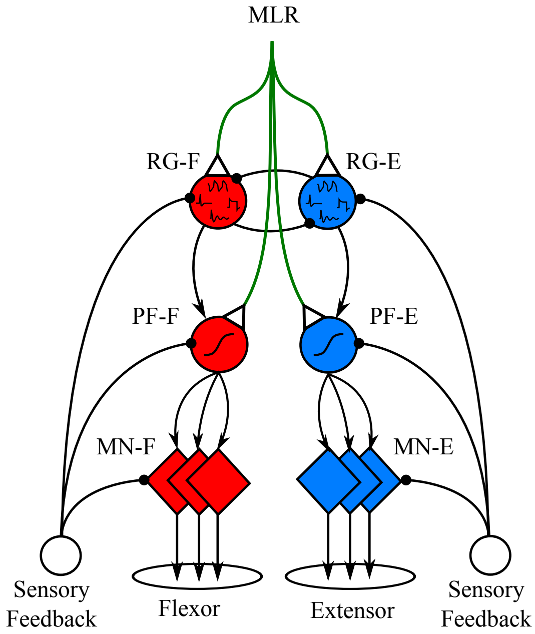


Figure 4.2: Model of one joint CPG controller with three layers: Rhythm Generator (RG), Pattern Formation (PF), and Motor Neuron (MN) layer.

patterns [Endo et al., 2004, Matsubara et al., 2006, Liu et al., 2008]. In Matsuoka and Taga's models the oscillatory behavior arises from the mutual connection between neurons, however, each neuron cannot produce oscillation activity without coupling [Endo et al., 2008]. Furthermore, the Matsuoka model is limited to show only oscillatory patterns [Liu et al., 2007]. This widely used oscillator cannot generate different activities rather than only oscillation, while complex behavioral motion requests a combination of different motor behaviors. Another neural model needs to be employed in robot locomotion in order to show complex motions while interacting with the surrounding world.

Rowat & Selverston proposed a neural model based on two cells with self-rhythmic generation ability [Rowat and Selverston, 1997]. Each cell independently generates its own pattern according to two parameters that are related to the membrane conductivity for fast and slow currents. All fast membrane currents are represented by a single fast current and all slow membrane currents are represented by a single slow one. The cell model is represented by two differential equations:

$$\tau_m . \frac{dV}{dt} = -(fast(V, \sigma_f) + q - i_{inj}) \tag{4.1}$$

$$\tau_s . \frac{dq}{dt} = -q + q_\infty(V) \tag{4.2}$$

$$\tau_m < \tau_s \tag{4.3}$$

Where V is the membrane potential and q is the lumped slow current. While the fast current is supposed to activate immediately, the membrane time constant τ_m is assumed to be significantly smaller than the slow current's time constant for activation τ_s. The ratio of τ_s to τ_m was fixed to 20 in [Rowat and Selverston, 1997], but when the ratio is as small as 1.5 most model patterns still arise. The injected current is i_{inj}. An idealized current-voltage curve for the lumped fast current is given by:

$$fast(V, \sigma_f) = V - A_f.tanh((\sigma_f / A_f)V) \tag{4.4}$$

The fast current represents the sum of a leak current and an inward Ca^{++}. The dimensionless shape parameter for the current-voltage curve is given by:

$$\sigma_f = \frac{g_{Ca}}{g_L} \tag{4.5}$$

Where g_L is a leak conductance and g_{Ca} is the calcium conductance. $q_\infty(V)$ is the steady state value of the lumped slow current, which is given by:

$$q_\infty(V) = \sigma_s(V - E_s) \tag{4.6}$$

$q_\infty(V)$ is linear in V with a reversal potential E_s. σ_s is the potassium conductance g_K normalized to g_L. σ_s is given by:

$$\sigma_s = \frac{g_K}{g_L} \tag{4.7}$$

This model can be extended to show two different conductivities for inward and outward; with conductance σ_{in} for inward slow current smaller than for outward slow current σ_{out}. The steady state value of the lumped slow current is given by:

$$q_\infty(V) = \begin{cases} \sigma_{in}(V - E_s) & if \quad V < E_s \\ \sigma_{out}(V - E_s) & if \quad V > E_s \end{cases} \tag{4.8}$$

q and i_{inj} have the dimension of an electrical potential. A true current is obtained by multiplying the model current by a leak conductance g_L. V, E_s, i_{inj}, and q are given in millivolts while τ_s and τ_f are expressed in milliseconds.

With different values for the cell parameters, different intrinsic patterns can be generated: quiescence (Q), almost an oscillator (A), endogenous oscillator (O), depolarization (D), hyperpolarization (H), and plateau (P), as shown in Figure 4.3.

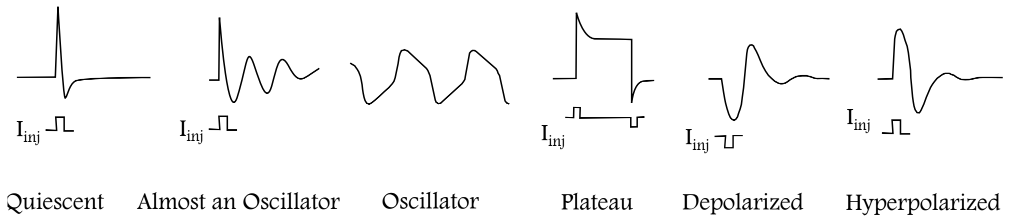


Figure 4.3: The six intrinsic patterns of Rowat and Selverston cell's model, [Rowat and Selverston, 1991]. These patterns are very similar to the four rhythms described in [Marder and Bucher, 2001]: Endogenous bursting, Postinhibitory rebound, Plateau potentials, Spike frequency adaptation.

Rowat & Selverston cell model was used in robotic locomotion in order to design neurocontrollers with genetic algorithms for various multi-legged robots [Hoinville, 2007]. This work showed that such a neural model is very well suited to generate adaptive rhythmic locomotion for multi-legged robots because it demonstrated properties of the neural plasticity through its parameters. This cell model has shown to be effective to control a simulated two-joint planar leg that slips on a rail in order to show the role of sensory feedback on a CPG model to improve the locomotion task [Amrollah and Henaff, 2010].

4.2.3 Model of Pattern Formation Neurons

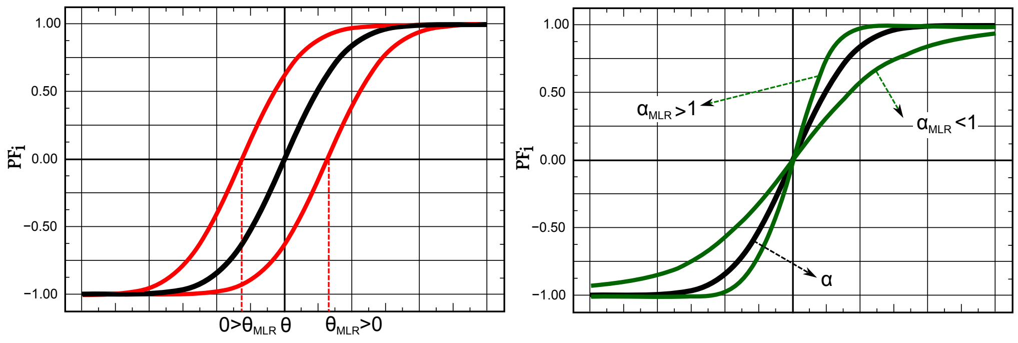
Neurons at this layer have inputs from rhythm generator layer and inputs from proprioception and exteroception. In neurophysiological studies, it has been shown that the pattern formation neurons have also a supraspinal drive that modulates the functionality of PF neurons not to change the activation rhythm, but rather to balance between flexion domination and extension domination [Rybak et al., 2006]. The supraspinal drive to pattern formation neurons ensures rhythm deletion of motor neurons activities without resetting the phase in the rhythm generation layer (e.g. RG neurons keep oscillation, while motor neurons are not active). Ryback *et al.* observe a deletion of the generated rhythm during animals locomotion, however, muscles are able to return to the previous rhythm, this is referred to as "rhythm deletion without phase resetting" [Rybak et al., 2006]. Although the output of motoneuron (extensor or flexor or both) was absent for a while, the original oscillation (rhythm) is preserved even after the deletion.

We propose a model for pattern formation neurons that take into account the biological inspiration related to rhythm domination and deletion and the brainstem descending control into pattern formation neurons [McCrea and Rybak, 2008]. The activation of pattern formation neurons in our model is calculated as follows:

$$PF_i = \frac{1}{1 + e^{\alpha.\alpha_{MLR}((\theta+\theta_{MLR})-I)}} \tag{4.9}$$

$$I = \frac{w_{rg2pf}.RG_i + \sum_{j=1}^{n} w_j.S_j}{n+1} \tag{4.10}$$

where PF_i is the activation value of the i^{th} pattern formation neuron, α is a positive constant that denotes the slope of the sigmoid function, θ is the center point of the curve that denotes the threshold of the neuron, I is the averaging input to pattern formation neuron, w_{rg2pf} is the weight of the synaptic connection between rhythm generation neurons and pattern formation neurons, RG_i is the activation of i^{th} rhythm generator neuron, S_j is the activation of the *proprioception* or *exteroception* neuron and w_j is the weight between this neuron and the pattern formation neuron. α_{MLR} is a constant that represents the descending control from the high level controller to modulate the activation of the neuron regarding the input range of pattern formation neuron. θ_{MLR} is the modulation of the threshold by the high level controller that drives the rhythm domination (extension/flexion). Figure 4.4 shows the activation function of pattern formation neuron with different descending control values.



(a)

(b)

Figure 4.4: Descending control of pattern formation neurons.

4.2.4 Model of Motor-Neurons

The activation of motor neurons is calculated as follows:

$$MN_i = \frac{1}{1 + e^{\alpha(\theta - I)}} \tag{4.11}$$

$$I = \frac{w_{pf2mn}.PF_i + \sum_{j=1}^{n} w_j.S_j}{n+1} \tag{4.12}$$

Where α and θ are the slope and the threshold on the sigmoid activation function. In this paper, they are fixed empirically to 5 and 0.5 respectively. I denotes the averaging input to motor-neurons. S_j is the activation of the related sensory-neuron and w_j is the weight between this neuron and the corresponding motor-neuron. w_{pf2mn} is the weight of the synaptic connection between pattern formation neurons and motor-neurons.

4.2.5 Model of Sensory Neurons

A static model of sensory neuron proposed by Ekeberg [Wadden and Ekeberg, 1998] is described in Equation (4.13). ρ_i is the activity of sensory neuron, α is a positive constant that denotes the dynamics of the neuron, θ is the amplitude and ϕ is the input on the neuron. ϕ can be an angular position, or a contact force [Geng et al., 2006].

$$\rho_i = \frac{1}{1 + e^{\alpha(\theta - \phi)}} \tag{4.13}$$

The extension and flexion sensory neurons in each joint (ES and FS) inhibit the corresponding motor neuron for this joint. This circuitry is referred to as articular reflex. ES and FS sensory neurons for each joint have similar threshold that was calculated as follows: $\theta = (\phi_{max} + \phi_{min})/2$, where ϕ represents the joint's angle. ES and FS sensory neurons have slopes α with different signs, one with positive and the other with negative slope. The values of α is selected in the way that ensures variation of at least 90% in the output of sensor neurons when the input (the joint's angle) changes between ϕ_{min} and ϕ_{max}.

4.3 CPG Behaviors

This CPG model has two control levels. The first one concerns the generated rhythm and the second level concerns the pattern shape. By controlling CPG on these levels it can generate different locomotion behaviors. We distinguish two categories of behaviors, the first is related to Pattern Generation layer while the second is related to Pattern Formation layer.

4.3.1 Pattern Generation

In the analytic study, after observing the phase diagram of a joint and changing the parameters σ_s and σ_f in the rhythm generators neurons, different motion behaviors were observed. Figure 4.5 shows the distribution of motion patterns in space of σ_s and σ_f.

The zone marked by (x) corresponds to plateau behavior, (*) corresponds to quiescence; (+) designates almost an oscillator, and the zone filled by (o) corresponds to oscillatory behaviors. Varying σ_s and σ_f in RG of a joint will change its motion pattern. The four detected basic motion patterns can lead the robot to achieve some complex tasks like walking, running, and jumping depending on the synaptic circuits between joint CPGs.

4.3.2 Pattern Formation

Patterns generated by RG neurons transit into motor neurons by the pattern formation layer where they are shaped according to external or internal variables. This

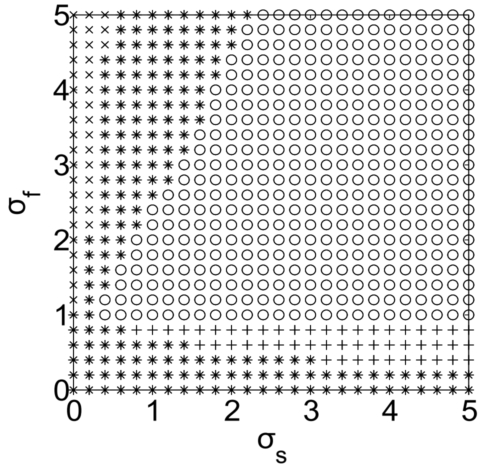


Figure 4.5: The different behaviors observed on the joint for the same injected current. (x): Plateau , (*): Quiescence; (+): Almost an oscillator, and (o): Oscillatory behavior. The voltage of the joint motor is calculated as in Equation (3.6).

control level explains non-resetting deletions behaviors that were observed in animals' locomotion [Lafreniere-Roula and McCrea, 2005] (deletion without resetting the rhythm). Furthermore, it also explains extensor and flexor domination. The role of the pattern formation layer is detailed in [McCrea and Rybak, 2008].

Figure 4.6 shows the effect of extensor and flexor domination on a generated oscillatory motion for one joint. The activation of the CPG neurons are illustrated for extensor-domination and flexor-domination with and without feedback.

Extensor-domination motion was achieved by a descending control $\theta_{MLR} = +0.5$, while flexor-domination was achieved by a descending control $\theta_{MLR} = -0.5$. The variation of θ_{MLR} changes the behavior of the CPG to show extensor or flexor domination behaviors, which bring certain robustness to deal with environmental changes.

4.4 Multi-Layered Multi-Pattern CPG for Humanoids

Most Central Pattern Generator models used in robot locomotion cannot explain the complex behaviors in human and animal locomotion. They are based on interconnected neurons that show oscillatory motion. Ijspeert has shown phase transition between walking activity to swimming activity of his salamander robot driven by a spinal cord model [Ijspeert et al., 2007]. However, the existing CPG models are unable to show complex behaviors that can mix both periodic and non-periodic motions.

This section introduces the complete architecture of the CPGs in order to generate diverse patterns for humanoid robot locomotion. It describes inter- and intra-limbs coordination precisely, and details the CPG circuitry for each joint. The role

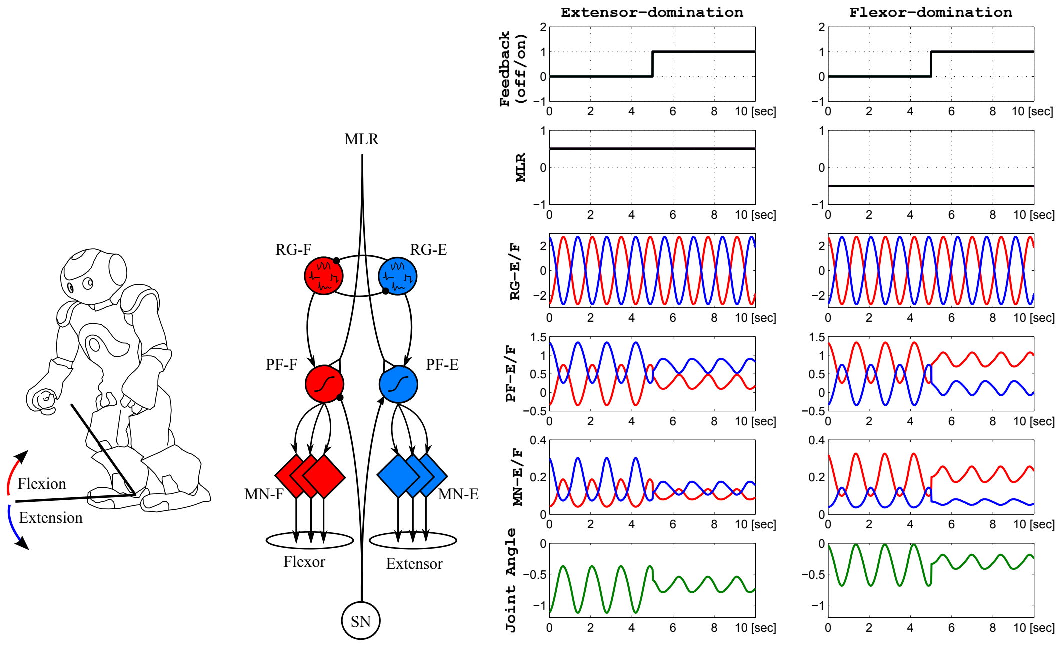


Figure 4.6: One-joint CPG with extensor/flexor dominated rhythm.

of phase resetting in the stability for robot walking on flat terrain is also presented.

Figure 4.1 illustrates the conceptual overview of the proposed multi-layered multi-pattern CPG for humanoid robot locomotion.

4.4.1 Motor Coordination

Complex locomotion movements in human and animals (walking, running, jumping ...) require combination and synchronization of body parts motion. In vertebrates, the cerebellum plays a major role in the accuracy of timing and magnitude of muscles activities during ongoing movement by using sensory information. Different studies show that it is involved in inter-limb coordination during locomotion [Bracewell et al., b 22]. However, the cerebellum is not responsible for movement initiation. The cerebellum has been suggested to play a role for motor timing [Purves et al., 2004]. It receives a copy of the executed motor program from the cortex and also works as a comparator. The cerebellum receives information from the muscles spindles, tendons, and joints. This information is used to measure the matching between the motor program imposed by the cortex and the executed motion on the muscles level. Therefore it is able to control the coordination of muscle activity and correct the motion.

Inspired by the cerebellum role in *inter-* and *intra-limbs* coordination in vertebrates, we suggest that the inter-joint coordination circuitry is task-related and has been defined by a descending motor program from high-level controller.

4.4.2 Control Architecture for a Planar Biped

To achieve a complex movement like walking, synchronization between joints is needed. The complex patterns like walking and running are always composed of synchronized basic patterns. The synchronization between patterns is ensured by coupling the CPGs for the joints. This section describes the neural controller of a 8-links simulated walker (see Figure 4.7).

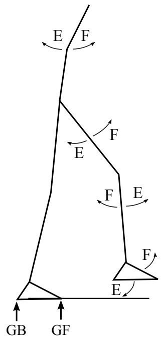


Figure 4.7: 2D simulated walker, simulation run in MATLAB environment. The simulated robot's mass is $22[kg]$. The body part lengths are: $10[cm]$ for the trunk, $16[cm]$ for the pelvis, $40[cm]$ for the tibia, $40[cm]$ for the thigh, and $20[cm]$ for the foot.

Fig.4.8 shows the proposed coupling circuits between the rhythm generator neurons for the hip, the knee, and the ankle joints of a simulated biped robot.

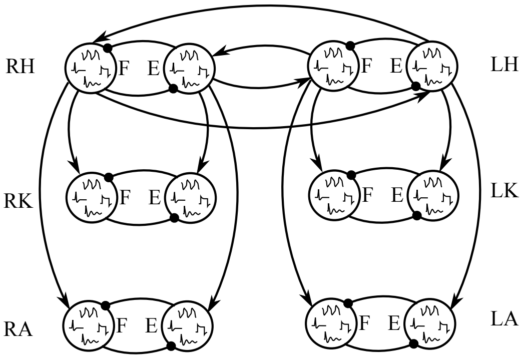


Figure 4.8: Coupling circuits between rhythm generators of the CPGs in hip, knee, and ankle joints. RH indicates right hip, RK is right knee, RA is right ankle. LH, LK, and LA indicate hip, knee, and ankle for left leg.

Each joint is driven by a simulated servo motor. With such simple coupling, the robot can carry out walking tasks from basic oscillatory patterns. With different coupling circuits, another task can be achieved. In some complex circuits, the robot can walk with different gaits. A desired task can be accomplished by defining basic patterns and special coupling circuit. The principle of our proposed circuit for walking is described by the activity between the CPGs which is regulated by excitatory synaptic connections. For inter-limb circuitry, the rhythm generator neuron extensor in the left hip (RG-E-hipL) excites the rhythm generator neuron flexor in the right hip (RG-F-hipR). The rhythm generator neuron flexor in the left hip (RG-F-hipL) excites the rhythm generator neuron extensor in the right hip (RG-E-hipR). The same synaptic excitation is proposed from the right hip to the left hip. For one leg, the rhythm generator extensor neuron in the hip (RG-E-hip) excites the rhythm generator extensor neuron in the knee (RG-E-knee) and the rhythm generator extensor neuron in the ankle (RG-E-ankle) of the same leg. The rhythm generator flexor neuron in the hip joint (RG-F-hip) excites the rhythm generator flexor neuron in the knee one (RG-F-knee) and the rhythm generator flexor neuron in the ankle joint (RG-F-ankle) of the same leg.

Figure 4.9 shows the Central Pattern Generator for the knee joint. RG-F, PF-F, and MN-F are rhythm the generator neuron, the pattern formation neuron, and the motor neuron for flexion. RG-E, PF-E, and MN-E are similar neurons for the extension side [McCrea and Rybak, 2008]. FS and ES are flexion and extension sensor neurons respectively from corresponding joints. AS is a hip extension sensor neuron for extension reflex [Geng et al., 2006].

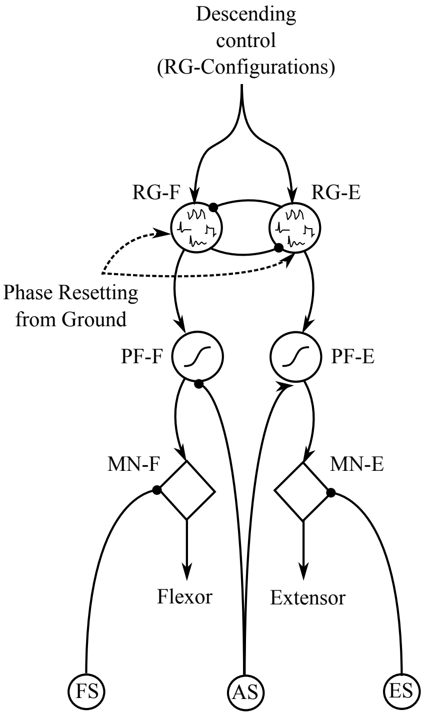


Figure 4.9: Central Pattern Generator for knee joint.

Figure 4.10 shows the Central Pattern Generator for the ankle joint. FB and FF are neurons that represent the risk of falling backward or forward according to the difference between the position of the Center of Mass projected on the ground and the Centre of Pressure. GB and GF represent the forces of contact for the corresponding leg at the back and at the front of the foot (see Figure 4.7).

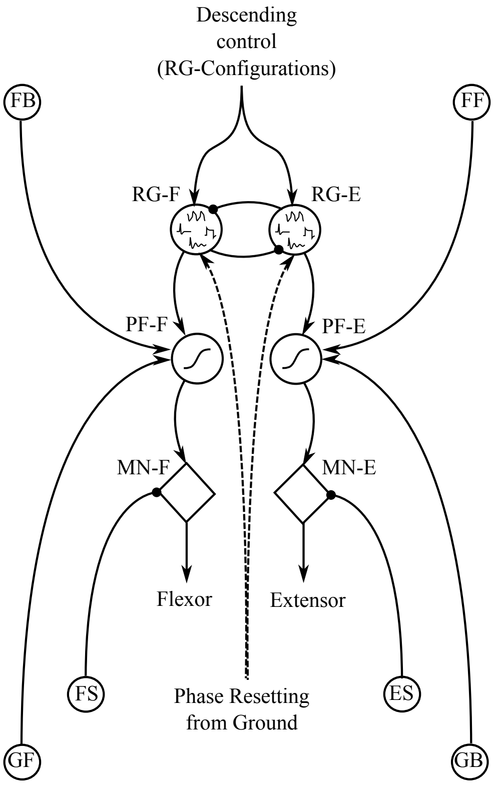


Figure 4.10: Central Pattern Generator for ankle joint.

Figure 4.11 shows the Central Pattern Generator for the hip joint.

The control circuit for the trunk joint is shown in Figure 4.12. The objective of this controller is to keep pelvis link close with the vertical. BS and BF are the sensor neurons that represent the angular position of the pelvis with the vertical direction, one neuron in the back and another in the front.

As described before, the locomotion comes from the interaction between CPG, sensory feedback, and descending control. Sensory information is used to shape the motion, to deal with some disturbances, and to achieve balance control [Taga, 1998]. Thanks to the interaction with sensory feedback, the robot can walk without a perfect knowledge of its dynamics.

The extension and flexion sensory neurons in each joint inhibit the corresponding motor neuron for this joint. This circuitry is referred to as articular reflex. Balance control is achieved by the difference between the center of pressure and the projection of the center of mass. In our model, the parameter of equilibrium is used as input of two neurons: falling forward and falling backward neurons. The activities of both neurons are injected in the pattern formation layer at the ankle CPG. If the robot has the risk to fall forward, the corresponding neuron becomes active to excite the pattern formation neuron extensor for the ankle of the stance leg. The flexor pattern formation neuron will be excited if the falling backward neuron becomes active. Now that the control architecture has been described and the model of rhythmic neurons

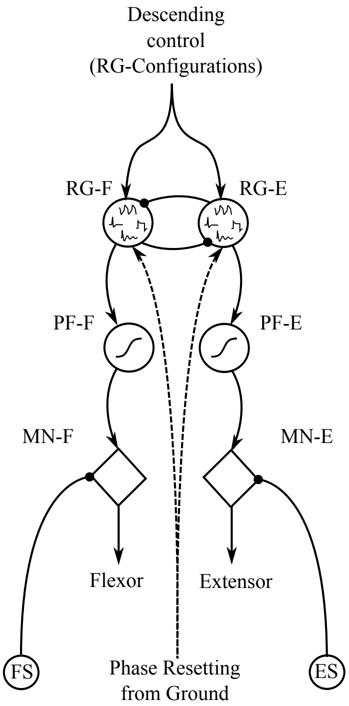

Descending control (RG-Configurations)
RG-F
RG-E
PF-F
PF-E
MN-F
MN-E
Flexor
Extensor
FS
ES
Phase Resetting from Ground

Figure 4.11: Central Pattern Generator for hip joint.

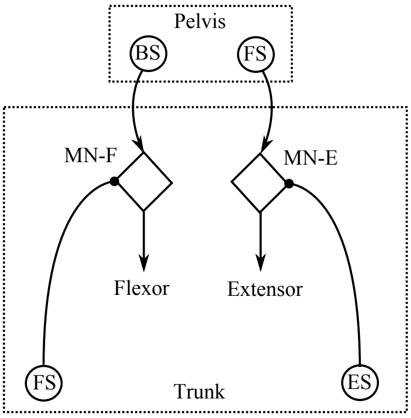


Figure 4.12: The control circuit for trunk joint.

has been determined, it is time to show how the simulated biped is learning to walk on a flat terrain. As the desired task is walking, and given that the coupling circuit is already defined, the biped will learn basic patterns, in space of σ_s and σ_f, that lead to successful walking.

4.4.3 Pattern Representation

The objectives of the learning mechanism is to detect in the space of σ_s and σ_f the basic patterns which lead to successful walk. Our previous work in experience-based learning mechanism with the vigilance concept has been used here to detect successful and failure walking patterns (see Chapter 3 for more details). Walking trial occurs inside a time window of ten seconds. Successful walking is defined when the simulated biped did not fall during the time window and achieved two steps at least.

This mechanism is composed of two phases, the evaluation phase, and the decision phase. It has been presented before in Chapter 3.

In the evaluation phase, two independent neural networks based on well-known Self Organizing Maps, proposed by Kohonen [Kohonen, 1995], are used to represent the knowledge in success and in failure. Success map learns in case of success trials, and failure map learns in case of failure ones. During learning, the two maps will be self-organized in the space of parameters that will be therefore divided into three zones: a zone of success represented by the success map, a zone of failure represented by the failure map, and a zone of conflict that corresponds to the interference between the two maps. The evaluation of any vector $\overrightarrow{v}$ from the space Ω belonging to success or failure is defined by the distance between $\overrightarrow{v}$ and each map. The distance of a vector with a map is the distance between this vector and the closest neuron in the state space (the winner neuron). For each $\overrightarrow{v}$, two distances therefore exist: one to success map called d_s, and another to failure map called d_f. In the decision phase, the comparison between the distance with success map d_s and the one with failure map d_f leads to an expected result in the case where the vector $\overrightarrow{v}$ is applied on the controller (trial). According to the expected result, if it may lead to failure, then an Early Warning Signal (EWS) becomes active to avoid the trial, and the decision will be "no-go". When EWS is inactive, the decision called "go"is taken. The decision mechanism is affected by the threshold of vigilance s_{vig}, which represents the tolerance to risk. The vigilance is related to human learning approaches and decision making [Ahn and Picard, 2005].

In order to increase the reflectivity of the vigilance threshold model proposed in our previous work, a modulation of the above mentioned threshold s_{vig} is introduced. This leads to get different values of it for each trial. Hence, this model increases the learning mechanism efficiency by extending the learning process to sectors of space of parameters. As an important issue, the risk behavior will change from cautious at the beginning of learning to be risky at the end. An example of vigilance threshold modulation is given as follows (see also Figure 4.13(c)):

$$y_1 \leq s_{vig} \leq y_2 \qquad \begin{cases} y_1 = a_1 - b_1 * log((x+c_1)^2) \\ y_2 = a_2 - b_2 * log((x+c_2)^2) \end{cases} \tag{4.14}$$

The coefficients values are $(a_1 = 0.9, a_2 = 1.47, b_1 = b_2 = 0.15, c_1 = c_2 = 20)$. They were selected after several attempts. y_1 and y_2 chosen curves ensure smooth change between the cautions and adventurous behaviors above mentioned behaviors. Walking patterns are represented by the success map. Falling patterns are represented by the failure map. With such learning mechanism, the learned failure map is as important as the learned success map, since patterns stored in the failure map can be used in an adaptation approach where walking patterns are limited (ex: in case of external disturbance). Fig.4.13 shows success and failure maps after learning 200 trials based on the new model of the vigilance threshold. The state space is normalized between 0 and 1 and each map has 25 neurons. Weights of neuron (w_1,w_2) denote the parameters of the rhythmic neuron ($w_1 = \sigma_s$, $w_2 = \sigma_f$). Therefore, there are 25 different configurations in each map that match 25 successful walking gaits stored in success map, and 25 unsuccessful walking patterns stored in failure map. Because of the topological properties of the Self Organizing Maps, three neurons in the failure map are situated in the success zone and show oscillatory behaviors ((0.39, 0.57),(0.46, 0.33),(0.17, 0.23)), see Figure 4.13(a). As these neurons did not represent any failure pattern, they are eliminated from the failure map.

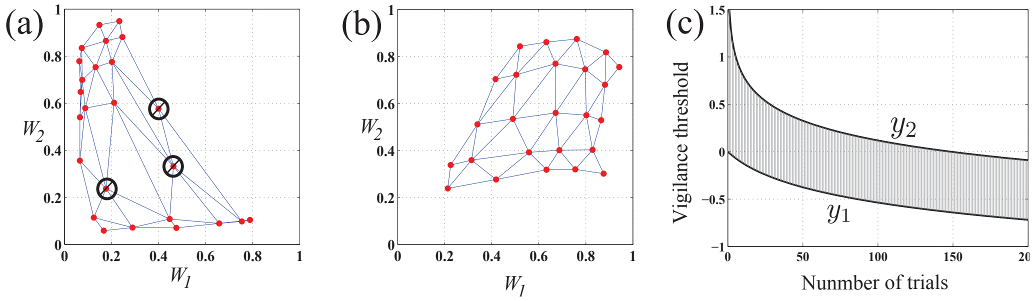


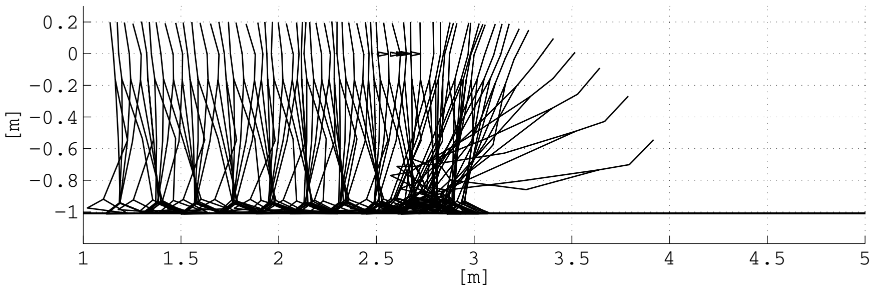
Figure 4.13: Success and failure maps after learning walk on flat terrain. (a)Failure map after learning unsuccessful walking patterns. Three neurons were eliminated from the map, because they did not represent any input vector, (b)Success map after learning walking patterns.(c) New vigilance Model related to learning iterations, $y_1 \leq s_{vig} \leq y_2$. The risk behavior will change from cautious at the beginning of learning to adventurous at the end.

4.4.4 Controller Robustness Against Perturbation

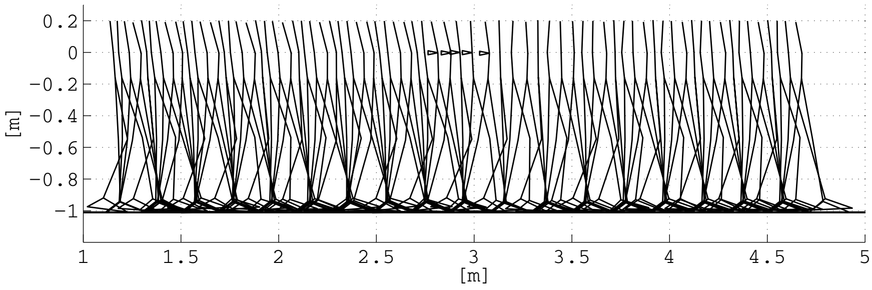
In this section we show the robustness of the neural controller against disturbance in two different ways. The first test of robustness was achieved by the introduction of phase resetting yield to disturbances. The second way to deal against disturbance is done by the introduction of a patterns switching mechanism that allows behavior adaptation to enable the handling of greater disturbing forces.

4.4.4.1 Phase Resetting

In the phase resetting technique, the rhythmic neurons were driven by ground contact force sensors. This technique brings robot dynamics closer to controller dynamics. Figure 4.14 shows the role of the rhythm generator phase resetting subjected to a disturbing force of $10N$ applied on the back of the simulated walker during a whole step. In the first case, walking is achieved without phase resetting. In the next case, walking is achieved with phase resetting of the rhythm generator neurons; this gives more robustness to the walking against the disturbing force. Of course, with a larger force, this technique cannot guarantee to avoid the biped robot from falling.



(a)



(b)

Figure 4.14: Phase resetting role in robustness. (a) shows walking with oscillatory patterns without phase resetting, the simulated walker falls after a disturbing force applied on the back during whole step. (b) shows walking and resistance to disturbances with phase resetting by ground force sensors.

4.4.4.2 Pattern Classification for Adaptive Behavior

As shown in the previous section, the walking task was achieved in the success map zone for the proposed coupling circuits. Because of the synaptic connection between rhythmic generator neurons for all joints, patterns cannot be independent. Then, the same pattern in all joints exists whenever the coupling circuitry is active. To have different patterns on different joints at the same time, the synaptic connection between the CPGs must be inhibited. By having independent patterns in the hip, the knee and the ankle joints, the biped can achieve some complex behaviors. This section introduces another technique for robust reaction to an external disturbing force.

As switching between success-map neurons during walking will change the walking pattern and therefore the walking gait, it can also be interesting to switch between these neurons against external disturbance. The limitation of this algorithm will appear for a large disturbing force. With a large disturbing force, staying in walking cycle inside success map is limited, and this cannot avoid falling. This can be solved by switching toward failure patterns stored in failure map neurons. However, inhibiting the synaptic connection between CPGs is necessary to get different patterns in different joints.

In such case, the space of parameters will be augmented, with different pair (σ_f, σ_s) for each joint. It increases from 2 dimensions in case of the existence of coupling circuitry to 12 dimensions in case of independent patterns. To reduce the dimensions, we propose to represent all the patterns of a joint on one axis only. This reduces the dimension by two, and facilitates classification and visualization of high-dimensional data. To do so, a metrics $\mathcal{E}$ which reflects the kinetic energy of one joint is introduced (eq. 4.15). Based on this metrics, an energy based classification of the patterns can be carried out.

$$\mathcal{E} = \int_{t_0}^{t_f} \dot{\theta}^2 \, dt \tag{4.15}$$

Figure 4.15 shows the logarithmic scale of the energy-based metrics for all the motion patterns of Figure 4.2. Figure 4.16 shows the logarithmic scale of the energy based metric of all neurons of failure and success maps given in Figure 4.13. The first 25 neurons belong to the failure map, and the last 25 neurons belong to the success map. The different behaviors are separated according to the energy-based metrics of motion patterns. Two neurons with Plateau have the lowest values for the energy-based metrics, then the 16 neurons with Quiescent behaviors, then the four neurons with Almost an oscillator, then all the neurons of success map according to the Oscillation frequency.

Patterns can be classified on a new axis according to the logarithmic scale of the energy based metric. As shown in Figure 4.16 patterns can be positioned on this axis in the following order: Plateau, Quiescent, Almost an oscillator, and Oscillatory patterns from low to high oscillation frequency. All neurons in success and failure maps can be placed on the new axis according to their rhythm. Therefore, the two dimensional space (σ_s, σ_f) can be represented in only one dimension axis. One axis is obviously needed for each joint. In the first step of the study, only synapses between CPGs of the hip and the knee joints are inhibited, while keeping the connection between CPGs of the ankle and the hip joints. Figure 4.17 shows the two dimensional space of patterns for the hip and the knee joints.

The walking zone in Figure 4.17(b) corresponds to oscillatory patterns in the

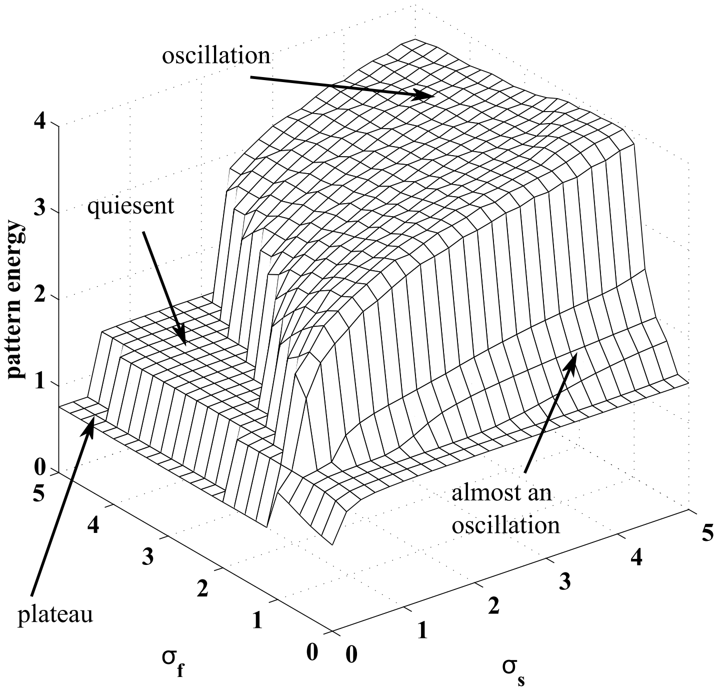


Figure 4.15: The energy-based metrics patterns for the space of σ_s and σ_f.

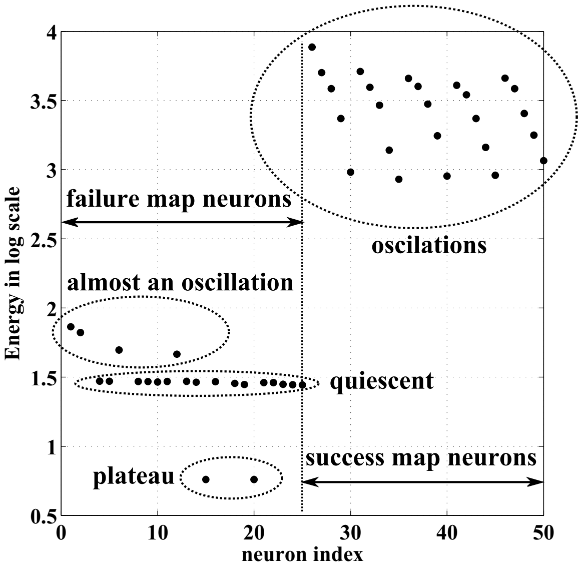
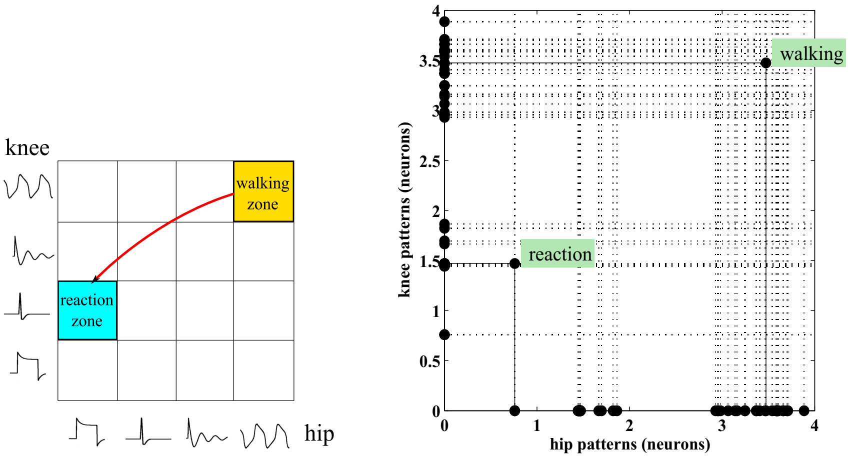


Figure 4.16: The energy-based metrics patterns for success and failure neurons represented on the horizontal axis. Neurons of success map represent oscillatory patterns with different frequency. Each neuron represents a pattern, but neurons are separated into four classes of patterns according to the energy-based metrics.



(a)

(b)

Figure 4.17: The space of patterns is for hip and knee joints, with an example of switching against disturbance. (a) Patterns switch from walking by oscillatory patterns to quiescent pattern for the knee and plateau for the hip. (b) Neurons switch from walking zone to other neurons that represent quiescent pattern for the knee and plateau for the hip. Each neuron represents one pattern.

hip and the knee joints. In case of external disturbing force, pattern manipulation is necessary to avoid falling. The figure shows the group of patterns in the hip and the knee joints for which the robot can react against the disturbance. An example for walking and reaction phases is shown in Figure 4.18. First, it presents the normal walking on a flat terrain without any disturbance. Next, it illustrates the fall consecutive to an external disturbing force of $45N$ applied on the back of the robot (the simulated robot mass is about 22 kg and the walking speed is almost $0.2m/s$). Figure 4.18(c) shows how the biped robot reacts correctly against the external force by adapting the behavior of the rhythm generators neurons.

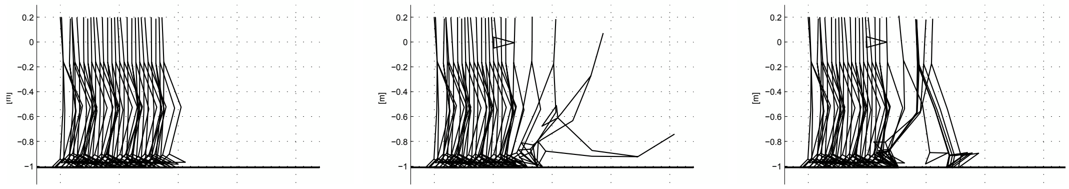


Figure 4.18: Effects of adaptation mechanism on the biped to avoid falling. Walking without disturbance. Falling due to the disturbance. Successful walking with adaptation to the disturbance.

4.4.5 Experiment on Humanoid Robot: Stability and Robustness

Figure 5.7 shows the proposed inter-joint coordination circuitry between the CPG of each joint of the NAO humanoid robot. This coordination occurs on the layer of rhythm generation neurons.

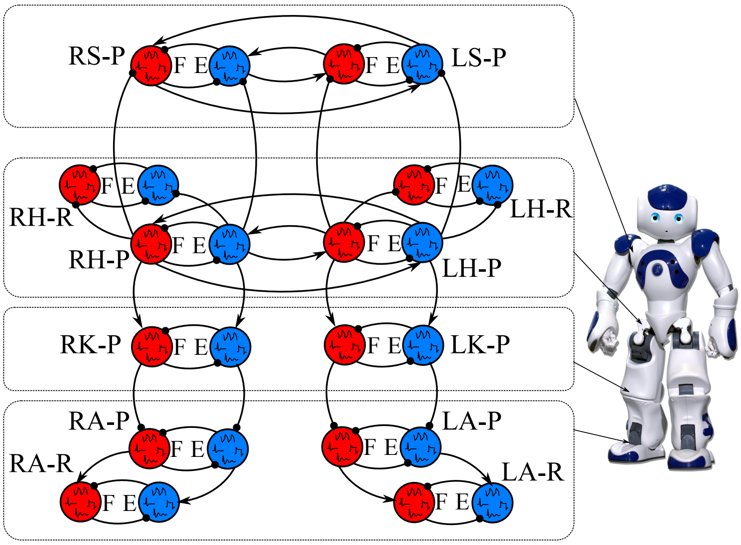


Figure 4.19: Coupling circuitry between rhythm generator neurons. 'F': flexion neuron, 'E': extension neuron. RS-P and LS-P are right and left pitch shoulder rhythmic neurons. LH-P and LH-R are pitch and roll rhythmic neurons for the left hip. The other layers of each CPG are hidden for better readability of the figure.

The principle of our proposed circuit for walking is described by the activity between the CPGs which is regulated by excitatory synaptic connections. For inter-limb circuitry, the rhythm generator neuron extensor in the left hip pitch (E-LH-P) excites the rhythm generator neuron flexor in the right hip pitch (F-RH-P) and inhibits the rhythm generator neuron extensor in the left shoulder pitch (E-LS-P). The rhythm generator neuron flexor in the left hip (F-LH-P) excites the rhythm generator neuron extensor in the right hip (E-RH-P) and inhibits the rhythm generator neuron flexor in the left shoulder pitch (F-LS-P). The same synaptic excitation is proposed from the right hip to the left hip. For intra-limb circuitry, the rhythm generator extensor neuron in the hip pitch (E-H-P) excites the rhythm generator extensor neuron in the knee pitch (E-K-P) and inhibits the rhythm generator extensor neuron in the hip roll (E-H-R) of the same leg. With such simple coupling, the robot can carry out walking task from basic oscillatory patterns. However, different coupling circuits can lead the robot to achieve different tasks. A desired task can be accomplished by defining basic patterns and the related coupling circuit.

One of the important phases of the design of a CPG circuitry is the connection with feedback. Proprioception and exteroception feed the CPG neurons at different layers: pattern generation neurons, pattern formation neurons, motoneurons, and interneurons [Rybak et al., 2006]. Particular joint proprioception has effects on the corresponding motor neurons of the same joint (e.g. the extensor sensor neuron inhibits the extensor motor neuron of the same joint), and other joint proprioception feeds motor neurons of other joints (e.g. AS sensor neuron in Table 4.2 represents the effect of hip joint angle on the knee motor neuron) [Manoonpong et al., 2007]. This section details the connectivity inside the CPG for each following joint of the robot: shoulder-pitch, hip-pitch, hip-roll, knee-pitch, ankle-pitch, and ankle-roll.

The weights of the synaptic connections of the CPG of hip joint pitch, hip-roll, shoulder-pitch, and ankle-roll are shown in Table 4.1. ES and FS are the extension and flexion sensory neurons for the corresponding joint. The variation of the descending control σ_s and σ_s allows the CPG to produce different motion patterns (Plateau, Quiescent, Almost an Oscillator, and Oscillators with different frequencies), while the variation of the descending control α and θ allow the CPG to shape the generated patterns.

The weights of the synaptic connections of the CPG of knee joint pitch are shown in Table 4.2. The connection of AS sensor neuron with knee motor neurons represents the stretch reflex, where the hip flexion exhibits the knee extension motor neuron and inhibits the knee flexion motor neuron. In this work AS sensory neuron is represented by FS sensory neuron for the hip joint pitch.

The weights of the synaptic connections of the CPG of ankle joint pitch are shown in Table 4.3. The exteroception GB and GF represent the effect of the ground contact sensor on the ankle joint. These neurons are tuned to match less than 0.1 (10%) of the Sigmoid activation in case of foot swing phase, while they match 0.9 (90%) of the Sigmoid activation in case of foot stance phase.

The exteroception FB (fall backward) and FF (fall forward) represent the effect of the torso angle with the vertical direction on the ankle joint. When the torso bends forward (walking direction) and increases the angle with the vertical, FF sensor neuron becomes more active. When the torso bends backward and increases the angle with the vertical, BF sensor neuron becomes more active. These two neurons project to pattern formation neurons of ankle-joint CPG.

Table 4.1: Synaptic Connections of the CPG for hip-pitch, hip-roll, shoulder-pitch, ankle-roll.

Source	*Target neuron*					
	RG-E	RG-F	PF-E	PF-E	MN-E	MN-F
σ_s	[0,5]	[0,5]	–	–	–	–
σ_f	[0,5]	[0,5]	–	–	–	–
α	–	–	]0,10]	]0,10]	–	–
θ	–	–	[-0.5,0.5]	[-0.5,0.5]	–	–
RG-E	–	-1	1	–	–	–
RG-F	-1	–	–	1	–	–
PF-E	–	–	–	–	1	–
PF-F	–	–	–	–	–	1
ES	–	–	–	–	-0.4	–
FS	–	–	–	–	–	-0.4

Table 4.2: Synaptic Connections for Knee-Pitch CPG.

Source	*Target neuron*					
	RG-E	RG-F	PF-E	PF-E	MN-E	MN-F
σ_s	[0,5]	[0,5]	–	–	–	–
σ_f	[0,5]	[0,5]	–	–	–	–
α	–	–	]0,10]	]0,10]	–	–
θ	–	–	[-0.5,0.5]	[-0.5,0.5]	–	–
RG-E	–	-1	1	–	–	–
RG-F	-1	–	–	1	–	–
PF-E	–	–	–	–	1	–
PF-F	–	–	–	–	–	1
ES	–	–	–	–	-0.4	–
FS	–	–	–	–	–	-0.4
AS	–	–	–	–	0.9	-0.9

Table 4.3: Synaptic Connections for Ankle-Pitch CPG

Source	*Target neuron*					
	RG-E	RG-F	PF-E	PF-E	MN-E	MN-F
σ_s	[0,5]	[0,5]	–	–	–	–
σ_f	[0,5]	[0,5]	–	–	–	–
α	–	–	]0,10]	]0,10]	–	–
θ	–	–	[-0.5,0.5]	[-0.5,0.5]	–	–
RG-E	–	-1	1	–	–	–
RG-F	-1	–	–	1	–	–
PF-E	–	–	–	–	1	–
PF-F	–	–	–	–	–	1
ES	–	–	–	–	-0.4	–
FS	–	–	–	–	–	-0.4
GB	–	–	-0.1	0.1	–	–
GF	–	–	0.1	-0.1	–	–
FB	–	–	-0.1	0.1	–	–
FF	–	–	0.1	-0.1	–	–

4.4.5.1 Stability Analysis

To analyze the stability of the walk of the NAO robot regarding the dynamic interaction between the robot and the ground, we studied the phase diagram of the robot torso inclination with the vertical direction during a long walking period (250[*sec*], 400 steps) (see Figure 4.20).

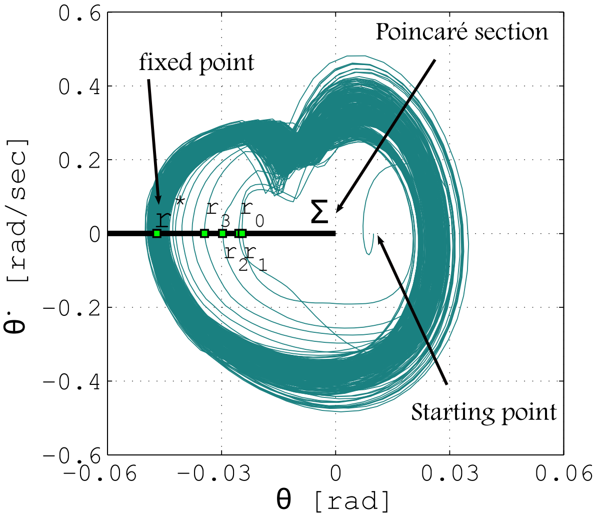


Figure 4.20: Phase diagram of NAO robot torso inclination with the Vertical direction and stability analysis of the cycle based on Poincaré map.

Starting from initial position, this diagram shows the convergence of the walking cycle into a limit cycle. We are using Poincaré maps to study the swirling flows near the periodic orbit. We define $\sum$ as a transversal section on the flow in one direction with null velocity. The Poincaré map P is a mapping from $\sum$ to itself. Figure 4.21 shows first returns (r_0, r_1, r_2, r_3) approaching the limit cycle. A fixed point occurs at $r^* = 0.046$ where the cobweb diagram for the sequence r_n intersects the 45-degree diagonal line. The cobweb shows that the fixed point r^* is globally stable.

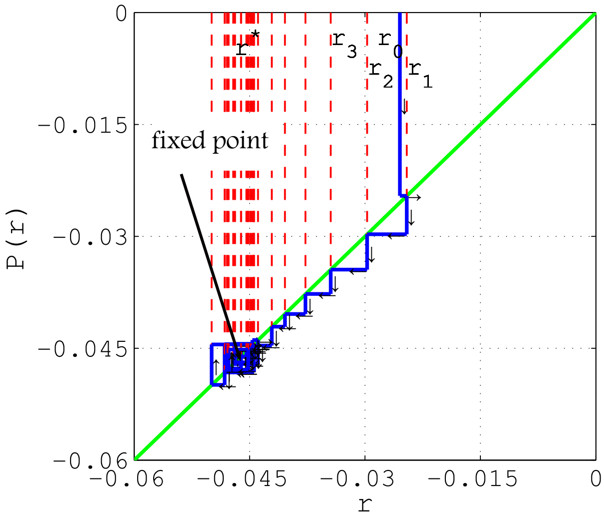


Figure 4.21: Cobweb diagram for the sequence r_n given by Figure 4.20.

4.4.5.2 Walking on Sloped Terrain

As in walking on flat terrain, periodic patterns are also employed for walking on sloped terrain. However, adjustments are required to guarantee stable switching from flat to sloped terrain. This can be done by changing the frequency and the amplitude of the periodic patterns and adjust the center of oscillation of each joint. The descending control from the high level into the multi-layered multi-pattern CPG allows adjusting these parameters by learning the appropriate behavior for each environmental state.

Figure 4.22 shows snapshots of NAO robot walking on 11 ° sloped up terrain. The amplitude of the oscillation is adjusted by modulating the slope of the sigmoid function in pattern formation neurons ($\alpha_{MLR} = -0.2$). The center of oscillation in each joint is adjusted by modulating the center point of the curve that denotes the threshold of the pattern formation neuron ($\theta_{MLR} = 0.15$). The frequency of oscillatory patterns is tuned by the descending control for rhythm generator neurons σ_s and σ_f.

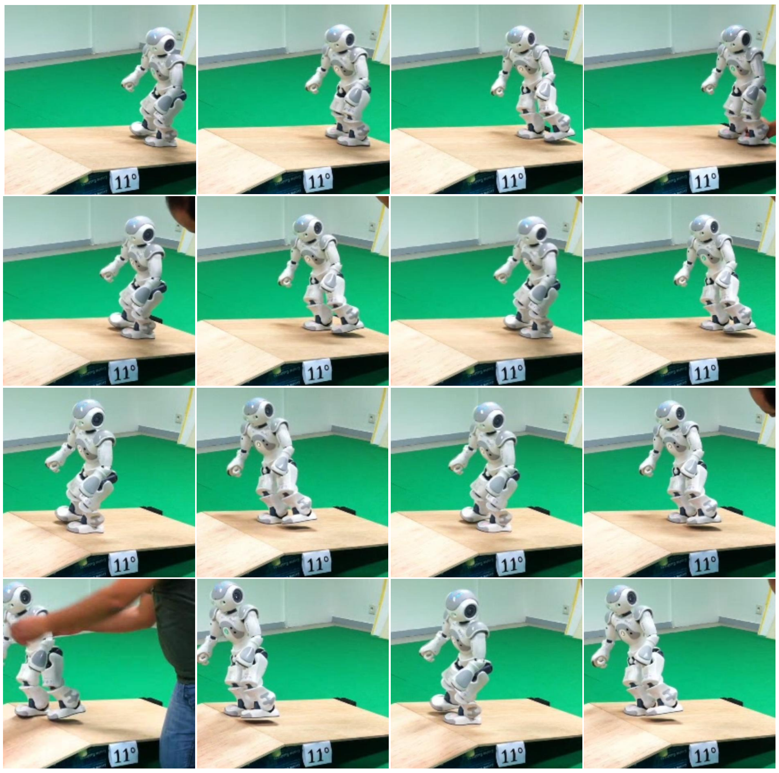


Figure 4.22: Nao robot while walking uphill (a video is available on: `http://web.ics.ei.tum.de/~nassour/naowalkinguphill.mp4`.).

4.4.5.3 Reaction Against Disturbance During Walking

In order to allow high-level control during goal-directed actions, the technique for dimensionality reduction proposed previously was employed to establish a space of patterns based on their energy at rhythm generation layer. Complex locomotion behaviors can then be represented in this pattern space, and switching between behaviors can subsequently occur in a simple manner. Figure 4.23 shows the pattern space for each joint of the NAO humanoid robot while switching from oscillatory patterns to non-oscillatory patterns in order to react against external disturbing force applied at the $9th$ second during $t = 0.1[sec]$. This study was carried out in the Webots simulator. The disturbance comes from the collision with a ball that pushes the robot (robot mass is $4.3[Kg]$) from the back in the walking direction with a force of $34[N]$.

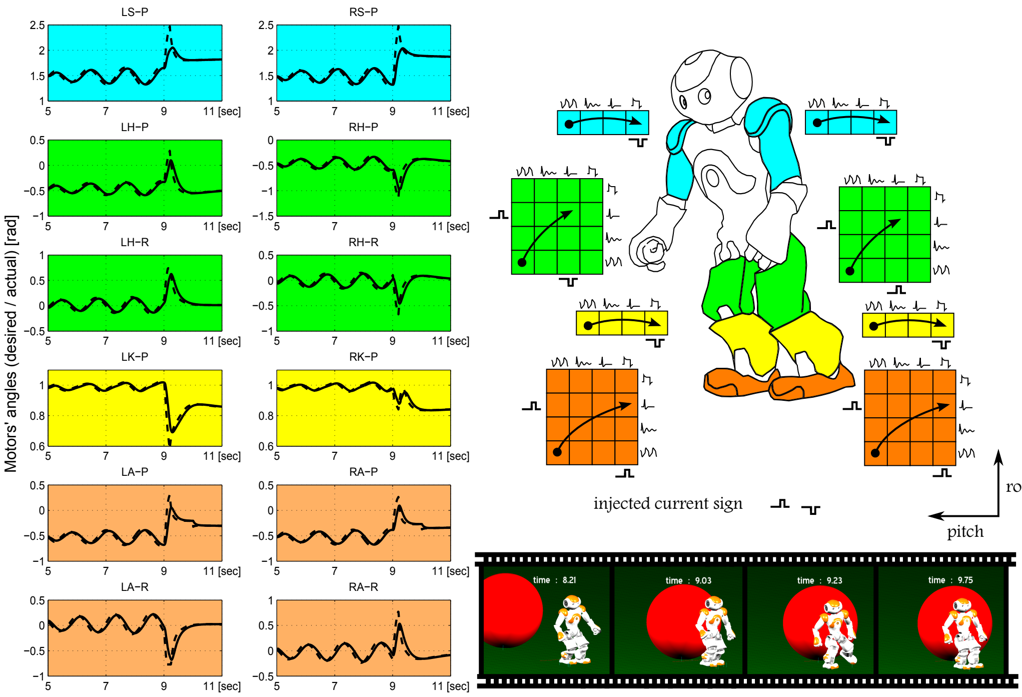


Figure 4.23: NAO humanoid robot in simulation reacts against external disturbance by switching into another motor program on the pattern generation layer (a video is available on: http://web.ics.ei.tum.de/~nassour/naoballreactionsim.avi).

On the right-hand side, the figure illustrates the switching for each joint in the space of patterns. On the left-hand side, the figure illustrates the switching in each joint with time. The direction of the switching in each joint is related to the direction of the injected current in the rhythm generator neuron. The robot was in walking behavior before the disturbance. Once the robot is subjected to the disturbance, switching occurs into a designed behavior. The two arms will move together to the back side by switching into plateau pattern in each shoulder. Hip joints switch to quiescent patterns with opposite direction for pitch joints and in the same direction for roll joints. Knee joints switch to plateau pattern. Ankle joints pitch switch to plateau pattern, while ankle joints roll switch to quiescent patterns.

Different motor programs can be introduced into the robot to show complex behaviors in the presence of environmental changes. Robots can acquire such skills by learning (e.g. self-exploratory learning, learning by observation ...). The ability to learn and acquire skills is related to the ability to represent these skills in memory (high-level issue), and the ability to generate the related motion patterns (low-level issue).

4.5 Conclusion

In this chapter, a new Central Pattern Generator model, named multi-layered multi-pattern CPG (MLMP-CPG) that is able to generate a diverse range of motion patterns was presented. This new model follows two neurophysiological studies, and the MLMP-CPG is based on three layers: 1) rhythm generator; 2) pattern formation; and 3) motoneuron layer. At each layer, exteroceptive or proprioceptive afferent feedbacks can affect the shape or the frequency of the generated patterns, especially during phase resetting and phase shifting. The global circuitry based on this CPG is validated in the walking control of a humanoid robot. In addition, a technique for dimensionality reduction depending on the energy-based metrics patterns was proposed in order to represent different motion patterns by only one parameter. A patterns space was introduced to deal with these patterns. The switching between patterns was simplified since it occurs in the patterns space. This switching allows fast changing in the behavior as a reaction for sudden disturbances. By employing this technique for dimensionality reduction, learning to switch between patterns can occur in the pattern space instead of inside the parameter space. Hence biologically inspired mechanism for action selection can be investigated.

Simulations as well as real-world experiments were carried out on a NAO humanoid robot. A Poincaré stability analysis showed that the walking was stable and the interaction with the environment flowed near a periodic orbit. Results showed that this neural circuitry is able to produce a 3D walking gait that stays stable even when the slope of the ground changes and that can shift to another gait when sudden external force pushes the robot. Unlike previously proposed CPG models, the multi-layered multi-pattern CPG is able to show complex behaviors that combine both periodic and non-periodic motion patterns within a single control framework. Such behavioral motions are essential for adaptive robot locomotion.

Chapter 5

Qualitative Adaptive Reward Learning (QARL) with Success Failure Maps

Contents

In the human brain, rewards are encoded in a flexible and adaptive way after each novel stimulus. Neurons of the orbitofrontal cortex are the key reward structure of the brain. Neurobiological studies show that the anterior cingulate cortex of the brain is primarily responsible for avoiding repeated mistakes. According to vigilance threshold, which denotes the tolerance to risks, we can differentiate between a learning mechanism that takes risks and one that averts risks. The tolerance to risk plays an important role in such a learning mechanism. Results have shown the differences in learning capacity between risk- taking and risk-avert behaviors. These neurological properties provide promising inspirations for robot learning based on rewards. In this chapter, we propose a learning mechanism that is able to learn from negative and positive feedback with reward coding adaptively. It is composed of two phases: evaluation and decision making. In the evaluation phase, we use a

Kohonen self-organizing map technique to represent success and failure. Decision making is based on an early warning mechanism that enables avoiding repeating past mistakes. The behavior to risk is modulated in order to gain experiences for success and for failure. Success map is learned with adaptive reward that qualifies the learned task in order to optimize the efficiency. Our approach is presented with an implementation on the NAO humanoid robot, controlled by a biologically inspired neural controller based on a central pattern generator. The learning system adapts the oscillation frequency and the motor neuron gain in pitch and roll in order to walk on flat and sloped terrain, and to switch between them.

5.1 Introduction

In this chapter, we bring forward an approach to better match biological models of brain-like mechanisms in learning tasks. The key point presented in this current work is the careful combination of two usually isolated studies of two distinct brain regions, namely, “Anterior Cingulate Cortex (ACC)” and “orbitofrontal cortex (OFC)”. We draw upon these studies in coming up with a functional and practical computational model that has been applied to a physical humanoid robot. Figure 5.1 provides a conceptual overview of this work. We have addressed the development of a learning mechanism based on the well-known self-organizing maps (SOMs). Walking has been used as an example task, which follows on our previous neuronal-based studies on the “Central Pattern Generator (CPG)” of the spinal cord for patterns generation for walking.

The adaptation property of the brain even with limited dynamic coding range enables efficient processing of different physical events like locomotion[Fairhall and Bialek, 2002]. The brain’s reward system discriminates a diversity of possible rewards, which can ensure best conditions for survival. The orbitofrontal cortex is related to reward dealing in the brain. Damages to the OFC have shown abnormal responses to changes to reward contingencies [Iversen and Mishkin, 1970]. Due to the sensitivity of neurons of this cortex to the types and the amount of rewards, OFC can be said to encode reward features into a scalar value [Thorpe et al., 1983]. Physiological studies demonstrated the adaptivity of the OFC in coding the reward according to the available rewards that changed in every block of trials [Tremblay and Schultz, 1999]. They show how the coding of reward in this cortex can be affected by the changes in reward distribution [Kobayashi et al., 2010]. This supports the concept that the OFC adjusts rewards information in flexible and adaptive manner after each new stimuli [Tremblay and Schultz, 2000].

Neurocognitive studies have identified an early warning system in the human brain that can avoid making past mistakes. They have shown how the brain remembers details about past dangers [Singer et al., 2004]. The ACC is activated during high-risk decision [Cohen et al., 2005], and also after making mistakes [Brown and Braver, 2008]. ACC responds to the sources of errors and to the earliest

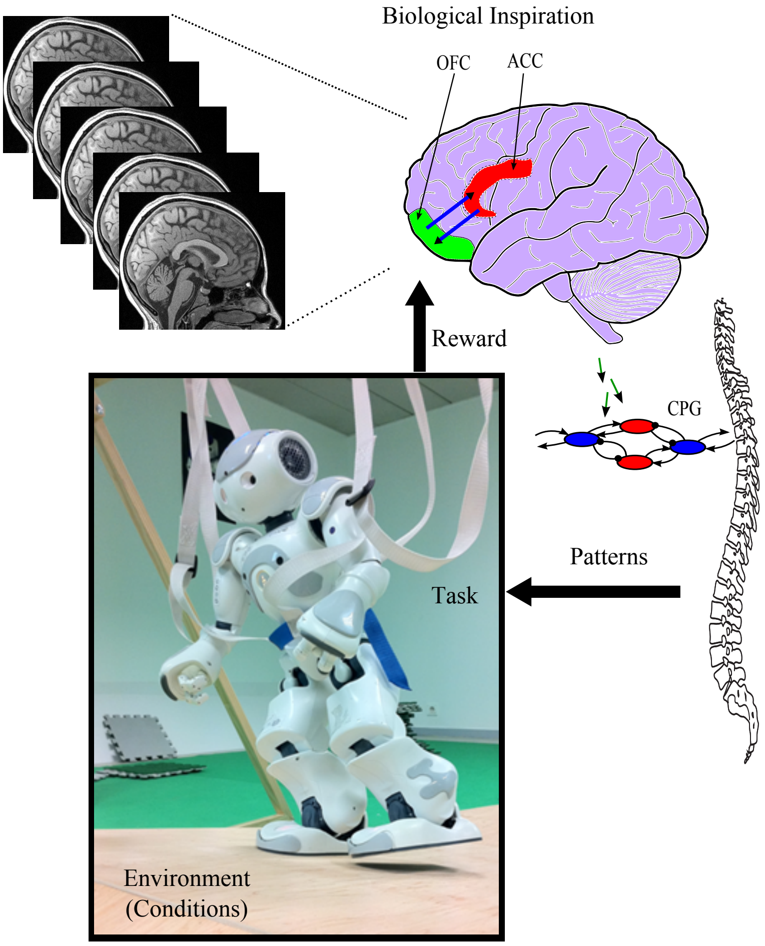


Figure 5.1: The conceptual overview of our work. ("ACC" is the Anterior Cingulate Cortex; "OFC" is the OrbitoFrontal Cortex; "CPG" is the Central Pattern Generator.)

sources of error information available (before making mistakes) [Mars et al., 2005], therefore, it acts as an early warning system that adjusts the behavior to avoid bad situations. Thus, ACC helps human to avoid repeating mistakes because it learns to identify previously occurred mistakes. Furthermore, ACC predicts error for non-visited situations.

It has been shown that the decision of taking risk was accomplished by activities in ACC and OFC [Cohen et al., 2005]. Activity increase with failure likelihood and also reward action likelihood. The fusion of the functionalities of these two cortex areas in one mechanism give raise to get a task learning system that could predict risky cases and avoid danger (e.g. a learning to walk task).

Computational models of learning systems such as techniques based on the associative memory like the CMAC neural networks that rely on offline trajectory generation. It first learn the joints trajectory, and then generate the learnt trajectory [Sabourin et al., 2006]. They assume that the models of the robot and the environment are available; therefore a stable walking pattern can be generated offline.

On the contrary, Reinforcement Learning techniques aim to adjust the physical actions and motor skills. It allows to the automatic determination of the ideal behavior within a specific context, in order to maximize performance. Simple reward feedback is required for the agent to learn its behavior [Sutton and Barto, 1998]. Robot bipedal locomotion research such as those by Morimoto *et. al.* [Morimoto et al., 2005], have improved biped walking controller using an approximated Poincaré map based on reinforcement learning . Their model controls the action between each two single support states for 2-D five-link biped robot with U-shaped foot. Another study used CMAC as a multivariable function to approximate the Q-factor in the Q-learning to learn the foot placement for the front leg in order to walk with a constant velocity [Chew and Pratt, 2002]. Reinforcement learning is used also as a subcontrol routine to compensate dynamic reactions of the ground around the ZMP [Katić and Vukobratović, 2004].

The main difference between our proposed works and above-mentioned works consist is the fact that we generalize learning of a task over varying conditions. Our method is motivated by the functions of ACC and OFC, which build on past experiences without requiring a predefined model of the environment. We propose a technique that works by learning an action-value function to follow a fixed policy by optimizing the energy of the task that keeps record of both positive and more importantly negative action consequences.

In this way, we aim to produce an early warning mechanism that can help to avoid repeating past errors in the generation of walking patterns of a humanoid robot. It is necessary for such a mechanism to experience mistakes, as well as experience of success, in order to evaluate new situations before taking any decision and performing the next action.

The notion of reward adaptation is introduced in order to qualify the walking task in term of energy. The notion of adaptive vigilance threshold is also introduced; the tolerance to risk is modulated to assure having same experience for success as for failure, which makes the system converge. Selection with a qualitative adaptive reward allows not only to determine the state space of parameters in the zone of success but also to optimize the learned task. It is used to adapt the intrinsic parameters of a low level controller based on a CPG for walking on flat and sloped terrains. Experimental validation was conducted on a NAO humanoid robot [Gouaillier et al., 2009].

The motivation of our works is to put forward better models based on biologically plausible mechanisms [Cheng et al., 2007]. In the current work, we highlight the importance of the different brain mechanisms and how they have been able to influence the development of real robotic control. To further carry this work forward, we have to match the functions of the mechanisms to additional brain studies [e.g., functional nuclear magnetic resonance (FMRI) studies].

This chapter is structured as follows. Sections 5.2 and 5.3 explain the neurobiological motivation and the inspiration of our improvement in success-failure learning. Section 5.4 presents the improvement of our learning mechanism in details. The interest of adapting vigilance is presented, then the concept of qualitative adaptive reward is described. Section 5.5 describes a biologically-inspired neural controller for locomotion based on CPG. Three intrinsic parameters of this low level controller are studied by the proposed learning mechanism. In Section 5.6, we apply the proposed methods on a robot in order to enable it to learn to walk on flat terrain. Learning to walk on sloped terrain is presented in the Section 5.7, which, we focus on the switching between different sloped terrains based on past experiences and sensory feedback.

5.2 Relative Reward Preference in Primate Cortex

The primate orbitofrontal cortex is involved in the motivational control of goal-directed behavior [Tremblay and Schultz, 1999]. It has an essential role in controlling and correcting reward-related and punishment-related behavior [Rolls, 1996]. Neurons of OFC are involved in the processing of motivational values of voluntary action rewards. [Tremblay and Schultz, 1999] shows that OFC neurons increase their activities during the expectation of reward and after receiving it. Authors explore the motivational properties in the macaque OFC neurons through a spatial delayed-response task where an initial instruction screen image indicate the left or right target of movement and the liquid or food reward that will be delivered to the monkey at the end of a trial. After a short delay, an image for two squares will appear as trigger that motive the monkey to move its hand from an initial position into the left or right target lever that was announced by the instruction. After a short delay, the correct action will be rewarded with a drop of liquid or piece of food. Figure 5.2 shows the framework of a spatial delayed response task.

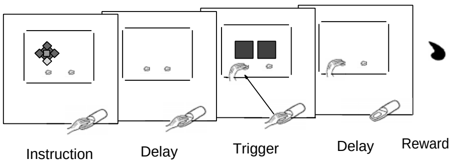


Figure 5.2: The framework of a spatial delayed response task for macaque monkeys. Extracted from [Tremblay and Schultz, 1999].)

According to [Tremblay and Schultz, 1999], subjects select more frequently rewards when they have to choose between different rewards at the same time. However, the frequent rewards can be ignored when more delectable rewards become available. It seems that motivational values are not fixed to defined rewards, unlike physical properties.

When two instruction images were presented instead of one, each of them are associated with different reward value, monkeys showed clear choice between rewards for each comparison. Figure 5.3 shows the OFC neural coding of relative reward preference in monkeys. Three food rewards are used (A, B, C). Two rewards were presented together at each trial (A and B, B and C, and A and C). The two rewards were alternated randomly in each trial block. All combination of rewards were tested. In 90 – 100% of trials blocks, monkeys chose the higher reward (A over B, B over C, and A over C). The neurons were considerably more activated before the preferred reward A when it presented with the non-preferred reward B. Same neurons were activated significantly for a non-preferred reward B when presented with a non-preferred reward C in the same trial block, see Figure 5.3. Therefore, the orbitofrontal neurons activate in function of more preferred rewards relatively. As a result, the reward discrimination in some OFC neurons is based on the relative preference rather than the physical properties [Tremblay and Schultz, 1999], this is what we call a qualitative adaptive reward coding *QARL* in the next section.

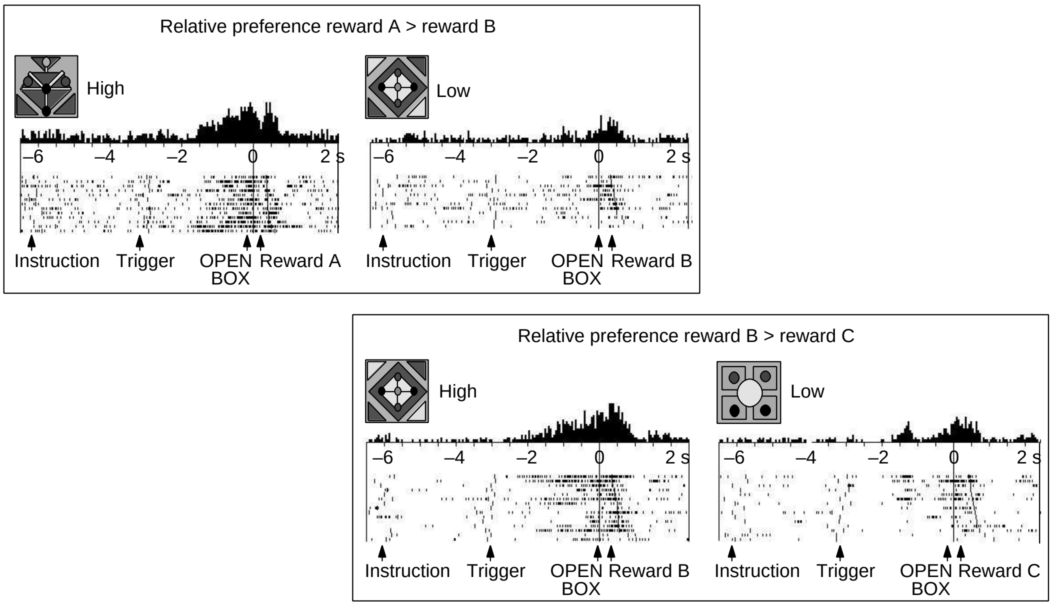


Figure 5.3: The relative reward preference coding in monkeys' orbitofrontal neurons for two trials blocks. A is raisin, B is apple, C is cereal. Each reward was predicted by a specific instruction image.Extracted from [Tremblay and Schultz, 1999].)

5.3 OFC-ACC Connectivity During Decision-Making

Brain regions involved in decision making have been studied widely [Cohen et al., 2005, Bicho et al., 2011, Doya, 2008]. The challenge was not only to detecting the brain regions that exhibit significantly during such mechanism, but also to understand how different brain regions interact between each other. [Cohen et al., 2005] designed a FMRI study that separates experimentally the neural activity related high-risk and low-risk choosing from other processes such as reward anticipation and evaluation during the general framework of decision making. They showed that choosing high-risk over low-risk was related with increased activity in both ACC and OFC. It seems that OFC carries on reward associations for stimulus [Rolls, 2000], and that ACC contains mechanisms that control the selection of appropriate behaviors [Van Veen et al., 2001]. According to [Cohen et al., 2005] no ACC activities were observed during low risk decision, while both ACC and OFC show a high activation when subjects made high-risk, as in Figure 5.4. However, this study was not able to distinguish whether ACC activation are related with small chance of a large reward or large chance of a failure.

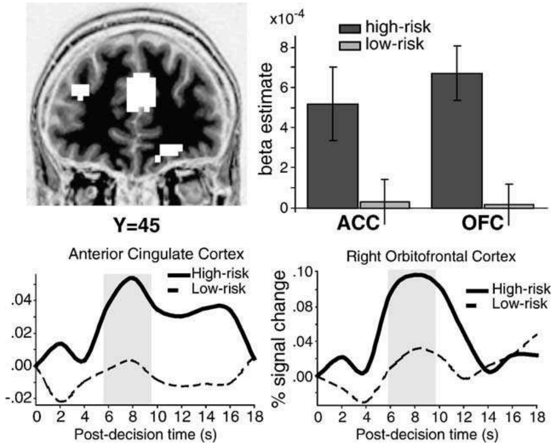


Figure 5.4: Brain regions exhibition during high-risk and low-risk decisions. ACC and OFC activation related decision time. Extracted from [Cohen et al., 2005].)

ACC and OFC exhibited similar patterns for activation and time courses and distinct patterns of functional connectivity. This suggests that they may play different and complementary roles in decision making [Cohen et al., 2005].

Based on the previous studies, we introduce the qualitative adaptive reward concept that works with success-failure learning to learn and to evaluate humanoid robots tasks and to optimize the performance.

5.4 Improved Success-Failure Learning

The success-failure learning presented in Chapter 3 represents the experiences for the success and for the failure by two self-organizing maps in order to learn a task. However, the selected vigilance threshold influences the convergence of both maps. It must be initiated in the way that guarantees the convergence of the maps.

Furthermore, the success map learns all successful trials with same importance, without taking into account that some trials can be achieved in a better way than other trials.

This section shows how to improve the learning process regarding the vigilance threshold and also regarding the learned trials with taking the performance into account. The first improvement consists in adjusting the vigilance in adaptive way, the second improvement is to control the learning process for successful trials with taking into account the rewards of trials in adaptive way which considers that learning starts from scratch where no prior information are available about the reward.

5.4.1 Vigilance Adaptation

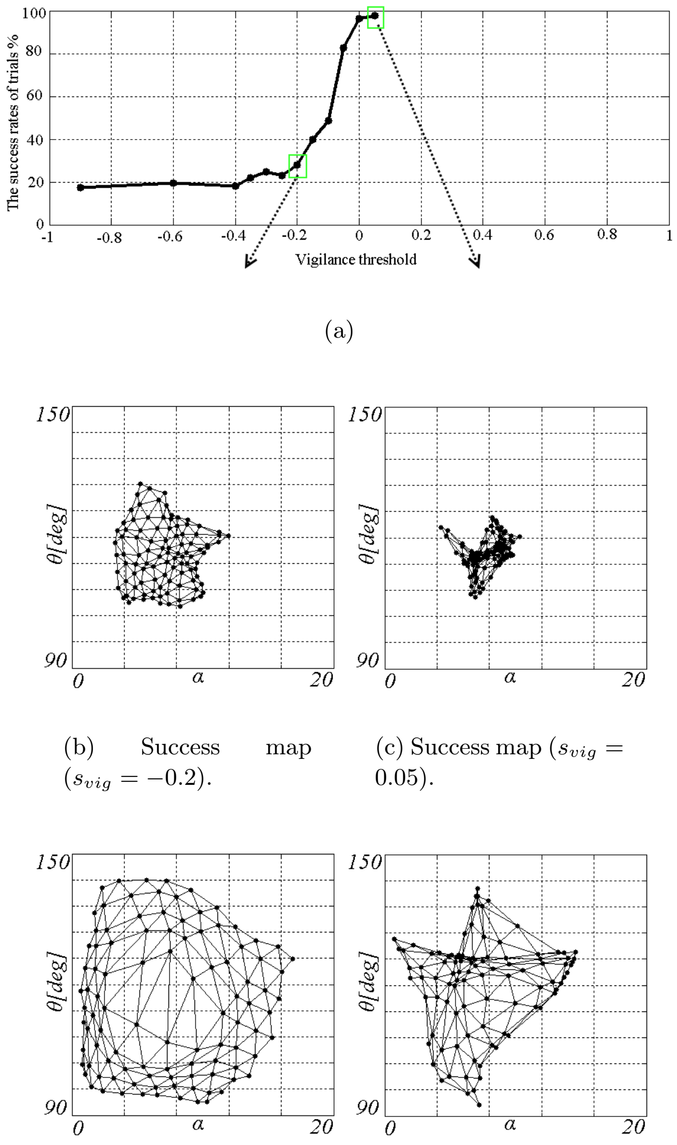
According to the vigilance threshold, the system can be risky or cautious during learning. Figure 5.5(a) shows succeeded trials ratio for learning stages with different vigilance thresholds [Nassour et al., 2009]. In our previous work [Nassour et al., 2009], learning occurred in two dimensional parameters space of a sensorimotor walking controller [Geng et al., 2006]. The first α denotes the dynamics of rhythmic movement in the hip joint (dynamics of extensor sensor neuron), while the second θ represents the amplitude of this movement (amplitude in the activity of extensor sensor neurons.

Note that for a vigilance threshold $s_{vig} = 0.05$, and after 500 trials, there is 98% of success and only 2% of failure. As a result, only the success map converges. The area occupied by the success map with cautious behavior will be much smaller than the area occupied by the success map with more risky behavior in term of vigilance threshold, see Figure 5.5(c) and Figure 5.5(b). On the other hand, with a vigilance threshold $s_{vig} = 0.05$ and the system avoiding risk, the failure map was not able to self-organized in parameters space Ω, largely due to the lack in the number of failed trials, as input vectors were not sufficient for learning, Figure 5.5(e). On the contrary, Figure 5.5(d), where taking risks, the rate of failure is more than 70%. With smaller vigilance threshold the system takes risks considerably, and the decision mechanism tends to accept all proposed vectors from Ω to be tested on the robot. Otherwise, no more selection occurs on the proposed pattern, which justifies the saturation on the left side in Figure 5.5(a).

Therefore, it is important to modulate the vigilance threshold to ensure success and failure maps learn together, converge, and avoid the saturation areas in Figure 5.5(a). For instance, the number of succeeded and failed trials can be used to influence *risk-taking* and *risk-avoiding* behaviors. Increasing the current vigilance threshold if the number of failed trials is greater than the number of succeeded trials will lead the system to risk-avoiding behavior. Decreasing that threshold if the number of failed trials is smaller than that for succeeded trials will lead to risk taking behavior.

5.4.2 Qualitative Adaptive Reward Learning (QARL)

In the proposed success-failure learning, the success map learns all succeeded trials with the same importance. However, succeeded trials can be qualified differently according to a desirable criterion. The objective is to influence learning by trials quality.This can be done by introducing the quality of trial as a weighted reward into the map. Each trial will have its own weighted reward representing the objective



(a)

(b) Success map ($s_{vig} = -0.2$).

(c) Success map ($s_{vig} = 0.05$).

(d) Failure map ($s_{vig} = -0.2$).

(e) Failure map ($s_{vig} = 0.05$).

Figure 5.5: Success and failure maps after learning on flat terrain with vigilance threshold $s_{vig} = 0.05$ (right), $s_{vig} = -0.2$ (left). (a): Rate of succeeded trials as a function of vigilance threshold.

criterion to be optimized. During each learning step, neurons will get closer to trials with high rewards rather than to trials with low rewards. After enough number of trials, success map will move into a spatial area associated with the highest rewards. The quality of a trial $\eta(k)$ is expressed as a number ranging from η_{min} to

η_{max}. However, this range cannot be determined at the beginning of learning. This is because no previous experience, neither for success nor for failure is available at the beginning.

Most reinforcement learning based robotic walking studies uses predefined constant to determine the maximum and the minimum reward or to determine the multiplier factors [Li et al., 2011, Endo et al., 2008]. In their definition of the reward function, maximum and minimum values are used to normalize the rewards [Li et al., 2011]. These parameters represent the minimum and maximum score for walking speed and for the zero moment point, which cannot be estimated without extensive experimentations on the robot [Li et al., 2011]. One of the challenges is to adjust these parameters automatically and adapt them by learning.

Therefore, adaptation is needed to re-determine the range limits η_{min} to η_{max} after each trial.

Let us denote input data by a n-dimensional vector $v(k) = [\zeta_1(k), \zeta_2(k), ..., \zeta_n(k)]$. Where k is the index of input data in a trials sequence. Let weights vector for the ith neuron in the map be $w_i(k) = [\mu_{i1}(k), \mu_{i2}(k), ..., \mu_{in}(k)]$, where k denotes the index in the sequence in which the neurons are generated. The updated weights vector $w_i(k+1)$ is calculated as

$$w_i(k+1) = w_i(k) + \gamma(k).h_{ci}(k).\rho(k).[v(k) - w_i(k)] \tag{5.1}$$

Where $\gamma(k)$ is the learning rate which is a scalar factor that defines the size of the correction. Its value decreases with the step index k. The index i refers to the neuron under processing, and c is the index of the neuron winner (that has the smallest distance from input vector $v(k)$). The factor $h_{ci}(k)$ is the neighborhood function. It is equal to 1 when $i = c$ and its value decreases when the distance between the neuron w_i and w_c increases. (e.g. one choice for a neighborhood function is to use a Gaussian kernel around the winning neuron). The factor $\rho(k)$ denotes the qualitative adaptive reward of $v(k)$ which is computed iteratively as

$$\rho(k) = \begin{cases} \rho_{max} & k = 0 \\ \frac{\rho_{max} - \rho_{min}}{\eta_{max} - \eta_{min}}(\eta(k) - \eta_{min}) + \rho_{min} & k > 0 \end{cases} \tag{5.2}$$

where:

$$\begin{cases} \eta(k) = F(v(k)) \\ \eta_{max} = max(\eta(k = 0, ..., K)) \\ \eta_{min} = min(\eta(k = 0, ..., K)) \end{cases} \tag{5.3}$$

The function F allows to obtain the criterion $\eta(k)$ for the trial that corresponds to $v(k)$. For instance, for a bowling robotic arm, $\eta(k)$ can denote the efficiency of the throw by combining the obtained result and the energy spent by the actuators. K is the index of the current trial. Maximal and minimal rewards ρ_{min} and ρ_{max} are predefined from trainer.

When the success map learns after the first succeeded trial, the reward will be maximal. After the second succeeded trial, the trial with highest quality matches the maximal reward, and the trial with the lowest quality matches the minimal reward. A scaling between maximal and minimal reward will occur for any new succeeded trial. A trial that matched a high reward in the start of learning phase may match a low reward at the end of learning.

By introducing the concept of *QARL* it will be possible to scale the quality of a trial according to the quality in previous experiences even with starting from scratch. After learning, the optimal parameter is presented by the success map neuron that is closed to the trial with maximum reward in training set. The general diagram of the proposed technique is presented in Figure 5.6.

The self-organizing map has been employed as a clustering technique because it guaranteed a safely switching between two different behaviors, e.g. some neurons can match high efficient walking patterns, while others can match patterns with high walking velocity. Intermediary neurons assure this switching. This is also the interest in having not only one solution for the walking problem.

The proposed algorithm can be regarded as a policy search method. Different search methods have been proposed previously for reinforcement learning on autonomous robot controller [Grudic et al., 2003, Peters et al., 2003]. Policy gradient method is one of the most accepted approach, it was used widely in robotics and in walking controller [Li et al., 2011, Endo et al., 2008]. Policy Gradient Reinforcement Learning (PGRL) is an optimization technique that guarantees the convergence to at least a local optimum, unlike the other RL search methods. The convergence of a global optimum cannot be guaranteed unless the correct initial condition - this limit flexibility of this method as such a dependency cannot be established easily.

Due to the random sampling before the decision phase and due to the vigilance adaptation technique, "QARL" can guarantee the convergence to all successful clusters in the state space. In addition, the use of self-organizing maps helps to represent the successful clusters even they are separated in the state space.

Section 2.4.3 shows the role of the evolutionary computation methods for parameters optimization in robotics. GAs can provide only the solution according to the fitness function that must be well described beforehand. In "QARL", we are interested by the way to the solution in order to build an auto-adaptive algorithm that can adjust the controller parameters in dealing with environmental changes.

As it is based on learning from success and from failure trials, the proposed method (QARL) can be considered as a RL method. In other RL methods both of negative and positive rewards can be used and the difference of the efficiency among the successful cases can also be considered. However, there is the essential difference between our proposed method and other RL methods such as a kind of multi-armed bandit problems. In multi-armed bandit problem (MAB), an arm can lead to success with some trials and to failure with others trials. MAB is based on the success probability in the building of its' prior tree. In QARL, learning and sampling occurs in continuous space, therefore the number of samples for trails is

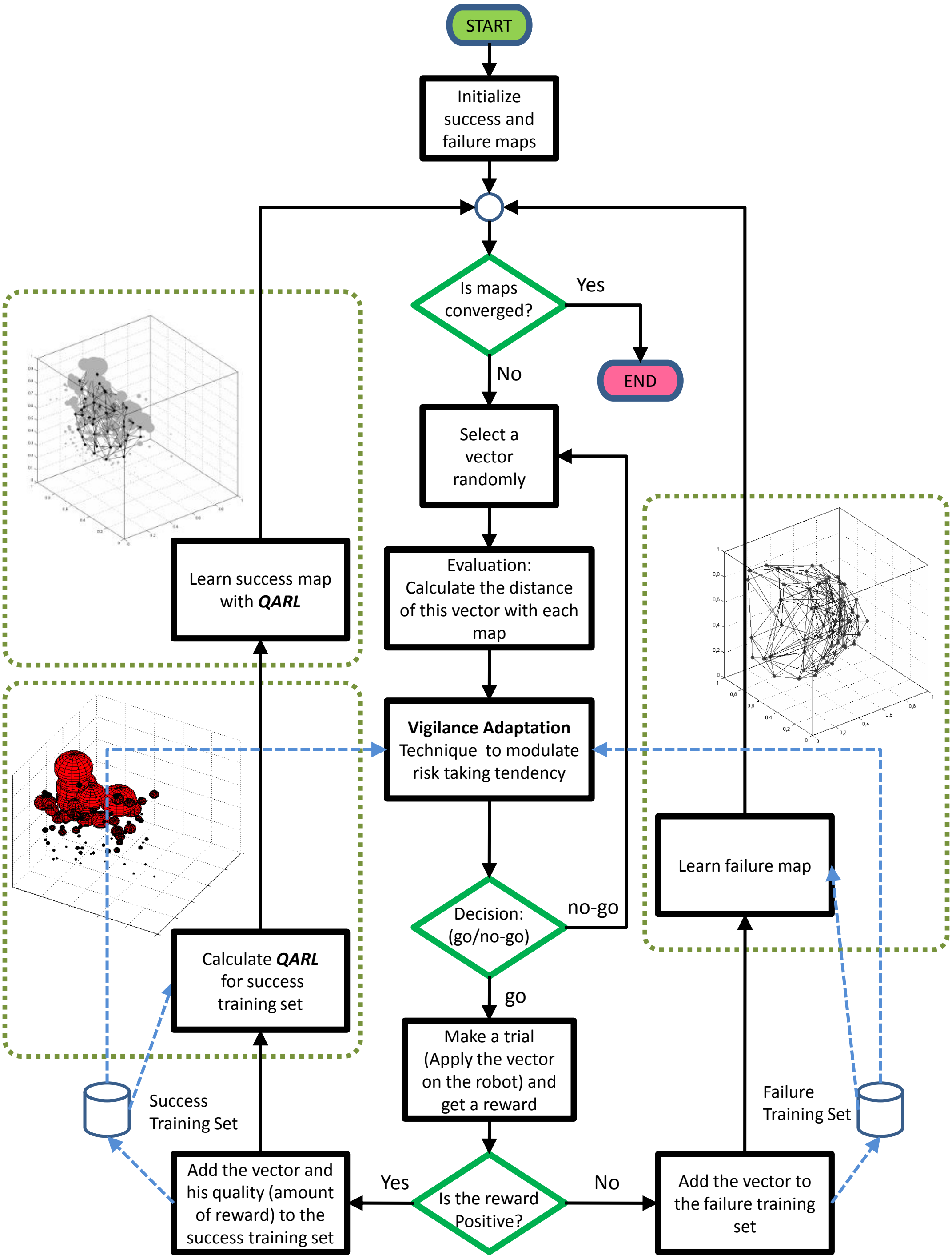

START
Initialize success and failure maps
Is maps converged?
Yes
END
No
Select a vector randomly
Evaluation: Calculate the distance of this vector with each map
Vigilance Adaptation Technique to modulate risk taking tendency
Decision: (go/no-go)
no-go
go
Make a trial (Apply the vector on the robot) and get a reward
Is the reward Positive?
Yes
No
Add the vector and his quality (amount of reward) to the success training set
Add the vector to the failure training set
Success Training Set
Failure Training Set
Calculate QARL for success training set
Learn success map with QARL
Learn failure map

Figure 5.6: Flow diagram for success-failure learning with vigilance adaptation concept and qualitative adaptive reward.

unlimited (not only multi-arms). A sample that had led to success or to failure will never change to opposite if it was selected to be tested again, which is not the case in MAB.

We applied the concept of qualitative adaptive reward with success-failure learning to humanoid robot, the humanoid NAO robotic platform is used in our experiments. Based to *QARL*, the robot learns to walk on flat terrain and constructs its experience for success and for failure. Then, learning to walk on sloped terrain will be presented and the robot will construct its experiences in walking on sloped terrain. The objective is to achieve success-failure learning in a space of intrinsic parameters of a low level controller for locomotion.

5.5 Bio-Inspired Neural Control for Locomotion

Biological evidences suggest that locomotion is mainly generated at the spinal cord, by a combination of a central pattern generator (CPG) and reflexes receiving adjustment signals from the brain particularly, from the cerebrum and the cerebellum [Orlovsky et al., 1999], [McCrea and Rybak, 2008], [Brown, 1911]. Locomotion is the result of dynamic interaction between the central pattern generator (CPG) and the connected feedback mechanisms. It has been shown that the central pattern generator is able to generate basic locomotor patterns according to the descending pathways that can control the locomotion tasks [Rossignol et al., 2006]. The feedback which dynamically adapts the locomotor pattern to the environment originates from muscles and skin afferents, as well as, from the basic senses (vision, audition, vestibular).

The Central Pattern Generator (CPG) is a neural mechanism that can produce rhythmic patterned outputs without rhythmic sensory or central inputs [Pinto and Golubitsky, 2006][Hooper, 2000]. It can generate periodic motor commands for rhythmic movements such as locomotion [Kuo, 2002]. Studies also showed that the CPG are localized in the lower thoracic and lumbar regions of the spinal cord [Kiehn and Butt, 2003].

These aforementioned studies have been taken into account in the design of robot's locomotion gait in order to realize such mechanisms for robust locomotion, especially on legged robots [Kimura et al., 1999, Taga, 2006, Ijspeert, 2008, Endo et al., 2008, Morimoto et al., 2008]. Different models of neural oscillators are widely used to generate rhythmic motion [Matsuoka, 1985, McMillen et al., 1999, Nakanishi et al., 2004, Righetti et al., 2006, Ludovic et al., 2009]. Such oscillations generated by two mutually inhibiting neurons are described by a set of differential equations (e.g. a Matsuoka Oscillator [Matsuoka, 1985]). Whereas Rowat and Selverston [Rowat and Selverston, 1991] model of rhythmic neuron can generate different types of patterns, not only oscillatory ones. The membrane currents of the neuron in this model are separated into two classes, fast and slow, in accordance with their time responses. Our study is based on the neural model proposed by [Rowat and Selverston, 1991] for modeling the rhythm generator that was detailed in Section 2.3.2 and described by Equations 2.20 and 2.21.

The ratio of τ_s to τ_m is about 20 as in [Rowat and Selverston, 1991]. In this study $\tau_m = 0.05$, and $\tau_s = 1$ for all rhythmic neurons.

With different values of the modeling parameters, different intrinsic behaviors can be achieved: quiescence, almost an oscillator, endogenous oscillator, depolarisation, hyperpolarisation, and plateau. Figure 4.2 shows the wiring diagram for one robot's joint. In this chapter, as we are interested in bipedal walking, which is periodic, only oscillatory pattern will be used, but different behaviors in the activity of these neurons can be used in robot's locomotion to achieve different locomotion tasks.

Walking gaits can be composed from basic synchronized patterns. The synchronization between patterns is ensured by coupling the joints' CPGs. Figure 5.7 shows the proposed coupling circuits between the rhythm generator neurons for the hip pitch and roll, the knee pitch, and the ankle pitch and roll, and the shoulder pitch joints of a NAO humanoid robot. With such simple coupling, the robot can carry out walking task from basic oscillatory patterns. With different coupling circuits, another task can be achieved.

The principle of our proposed circuit for walking is described by the activity between the CPGs, which is regulated by excitatory synaptic connections (see Figure 5.7). For example, the rhythm generator neuron extensor in the left hip pitch (LH-P E) excites the rhythm generator neuron flexor in the right hip pitch (RH-P F), inhibits the rhythm generator neuron extensor in the left hip roll (LH-R E) and the rhythm generator neuron extensor in the left shoulder pitch (LS-P E).

5.6 Learning to Walk

In this section, we apply the architecture proposed in the previous sections in order to learn efficiency walking for a bipedal humanoid robot, NAO. Figure 5.8 shows the neural model for the success-failure learning and the central pattern generator layers.

5.6.1 Walking Efficiency

We used success-failure learning with *QARL* to learn in a space of intrinsic parameters of the CPG controller (motor neuron gain in pitch, motor neuron gain in roll, and the dynamic of rhythmic generator neurons represented by σ_s). The optimization of walking efficiency was studied in term of energy as in [Abernethy, 2005].

Most of biomechanics studies on human movement focus on the efficiency of movement [Abernethy, 2005]. During flexion and extension of the joints, muscles release and absorb mechanical energy. When a muscle is exerting an active force and being lengthened by external forces at the same time, the mechanical energy is absorbed, and muscle is said to do negative work. It is said to do positive work, when the muscle is shortening as it develops a force. The efficiency with which a muscle operates is defined in [Abernethy, 2005] by

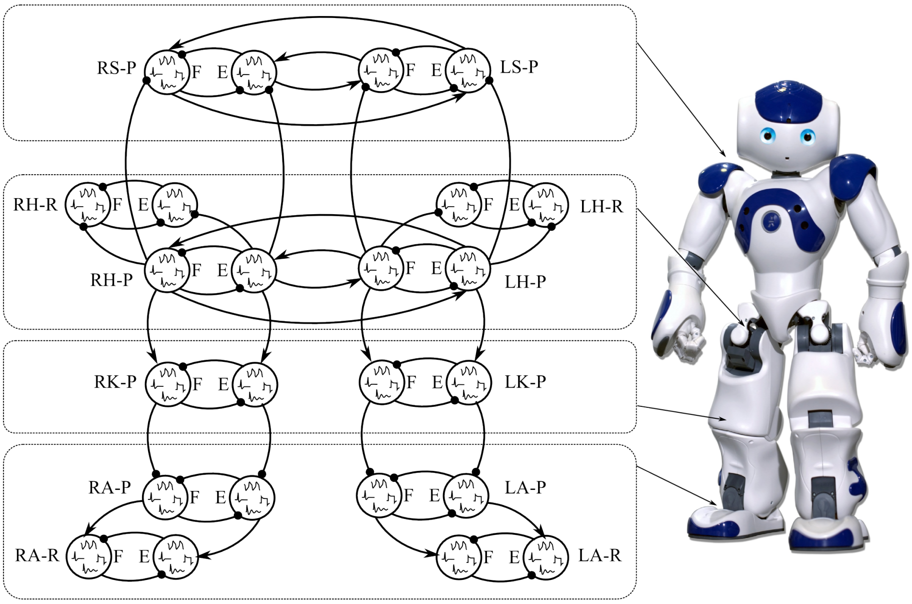


Figure 5.7: Coupling circuitry between rhythm generator neurons. 'F' for flexion neuron , 'E' for extension neuron. RS-P and LS-P are right and left pitch shoulder rhythmic neurons . LH-P and LH-R are pitch and roll rhythmic neurons for the left hip.

$$efficiency = \frac{mechanical\ work\ done}{metabolic\ energy\ consumed} \tag{5.4}$$

The mechanical work done on the muscle is considered as negative, while that done by the muscle is positive. The metabolic energy consumed by a muscle is generally defined as the entirety of its chemical processes [Guyton and Hall, 2006]. This study is also generalized from a muscle to whole body movements like walking, and running [Berryman et al., 2011, Margaria, 1976].

Inspired by biomechanical studies, the efficiency of walking for a humanoid robot can be described in a similar fashion.

In this case, *the mechanical work done* is the robot displacement energy in walking while *the metabolic energy consumed* can be represented by the actuators consumed energy as in Equation 5.4

5.6.2 QARL in Humanoid Walking

As our objective is to simultaneously learn and optimize walking, the robot learns to walk for a 1.5$[m]$ trajectory with start and end lines. In case of succeeded trials, the trainer sends a reward signal to the robot by caressing the head equipped with electrostatic sensors. Electric power is calculated at each instant as

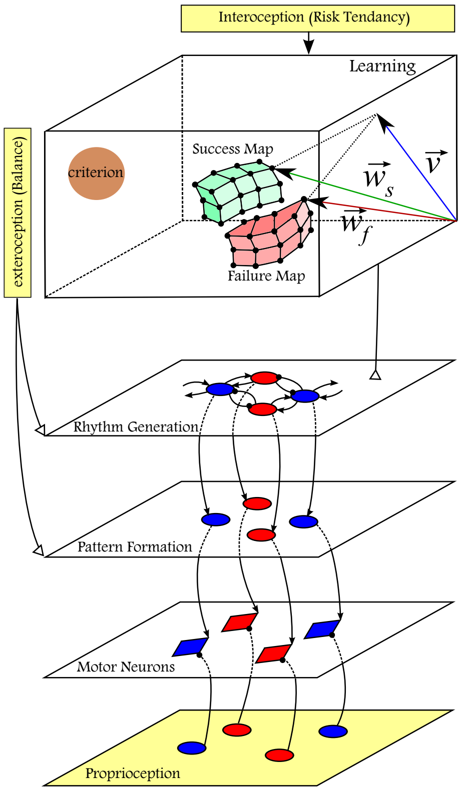


Figure 5.8: Flow diagram for success-failure learning with the central pattern generator layers.

$$P(t) = \sum_{i=1}^{n} R_i I_i^2 \tag{5.5}$$

Where n is the number of electric motors. I_i and R_i are the electric current and the electric resistance for motor number i.

The required electric E_e energy for all the trajectory is expressed as

$$E_e = \int_{t=t_0}^{T} P(t)dt \tag{5.6}$$

Where t_0 is the trial start time, and T is the trial end time, when the robot reaches the finish line.

The kinetic energy of a trial is given by

$$\begin{cases} E_k = \frac{1}{2}mv_a^2 \\ v_a = \frac{\Delta d}{\Delta t} \end{cases} \tag{5.7}$$

Where v_a is the average velocity for the entire trajectory, Δd is the trajectory length, Δt is the time difference between start and end of a trial, and m is the robot's mass.

The walking efficiency is calculated for each trial as:

$$\eta = \frac{E_k}{E_e} \tag{5.8}$$

The introduction of the efficiency for success map learning will shift the neurons of this map into the area in which the walking efficiency is high. This is done by using the concept of *QARL*.

Figure 5.9 shows *QARL* for success map in the beginning of learning (after 4 successful trials), and at the end of learning. Each sphere corresponds to a succeeded trial whose diameter represents the reward of this trial in the success map. It is to be noticed that, the trial corresponding to the maximum reward at the start of learning, indicated by a circle, will have a small reward at the end of learning. The interest of using this technique is to make success-failure learning search for new trials in the space area where walking efficiency in term of energy is high. In other words, this leads to learn and optimize in a defined space.

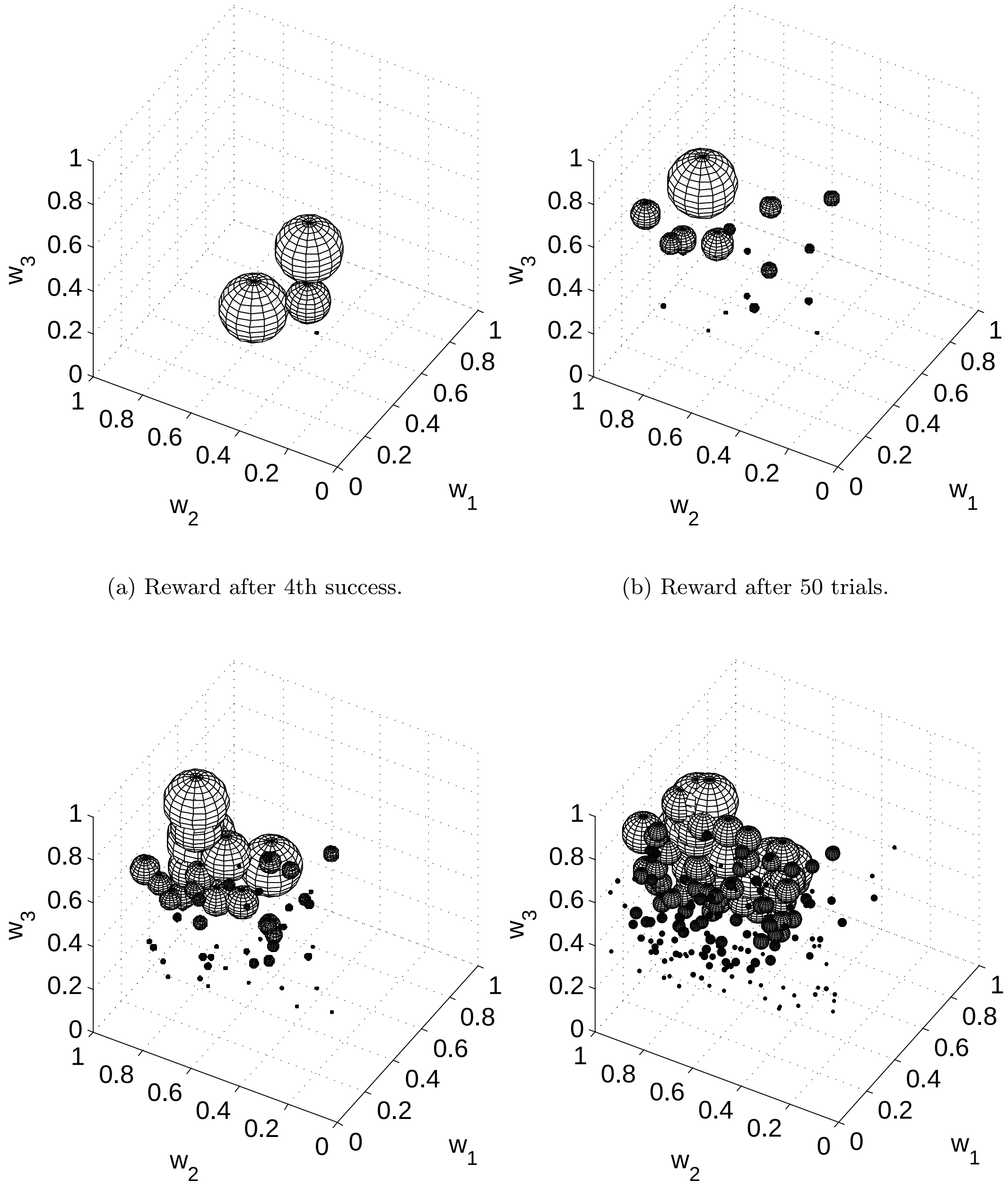
Figure 5.10 shows success maps after learning to walk on flat terrain with and without the technique of Qualitative Adaptive Reward. In Figure 5.10(a), the success map learns all successful trials with the same opportunity, i.e. with the same reward.

In Figure 5.10(b), the success map learns successful trials in accordance with its qualitative adaptive rewards. Trials with high reward influence success map neurons more than trials with low reward. Therefore, the success map will be attracted to the area where reward is high. This is influenced by the differences between highest and lowest rewards (scaling range limits: $[\rho_{min}, \rho_{max}]$), see Equation (5.2). In this study, ρ_{min} and ρ_{max} are set to 0.1 and 2.5.

The application of *QARL* influences the success map neurons to match more efficient patterns in the studied space. (e.g. some walking patterns represented by success map neurons learned without *QARL* show less efficient walking. These effects were reduced when *QARL* was applied.).

For the learning frameworks with and without the application of *QARL* shown in Figure 5.10, the performance was increased by 60% after applying *QARL*, this was calculated by the ratio of the highest efficiency neurons in both success maps (of with and without QARL). The ratio of the lowest efficiency of the neurons of success maps has increased by 40%. In order to provide sufficient precision in the network for our task, we have empirically selected a $5 \times 5 \times 5$ dimensional network space to represent the success and failure maps. Learning occurred with 500 trials for each case, without applying the auto-adjustable vigilance technique, the number of successful trials has increased 16% after applying *QARL*.

Computationally, all the processing of this learning framework in simulations as well as on the real robot can be performed in real-time. Thus, it makes our approach feasible for training on the real robot. Within the same cycle, joint angle commands are calculated in real-time and sent to joint motor circuit boards of NAO



(a) Reward after 4th success.

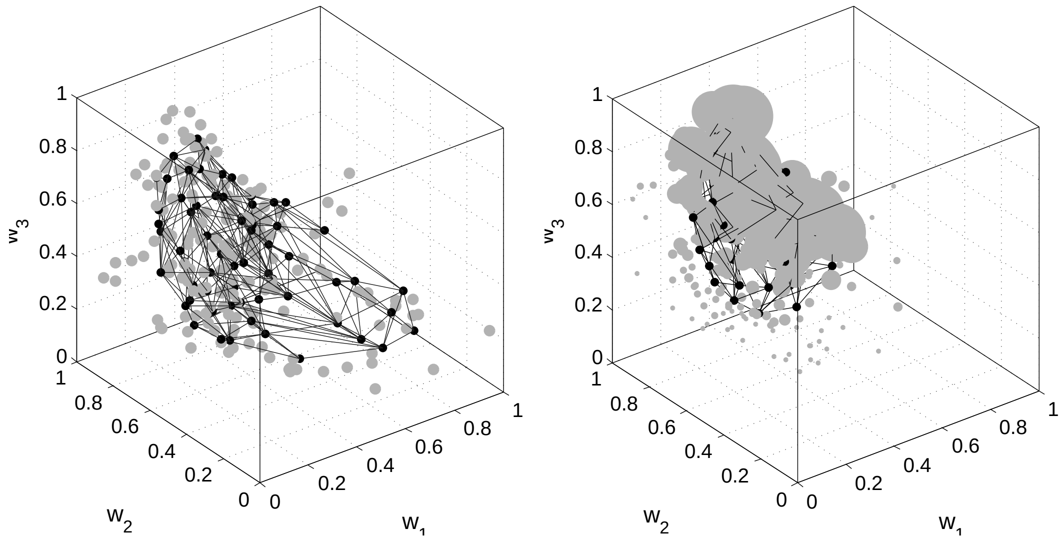
(b) Reward after 50 trials.

(c) Reward after 150 trials.

(d) Reward after 500 trials.

Figure 5.9: Successful trials' reward related to walking efficiency for learning success map. Where w_1 is motor neuron gain in pitch, w_2 is motor neuron gain in roll, and w_3 is σ_s, that related to the oscillation frequency.

every $10[ms]$. This is done inside a high priority thread on the robot. Physically, each trial require about 3 minutes, which includes learning and the experimental



(a) Learning with same reward. (b) Learning with adaptive reward.

Figure 5.10: The effect of *QARL* on success map. Success map after learning with the same reward for all successful trials (a). Success map after learning with adaptive reward (b). Gray spots represent successful trials reward. Note that, the map on the right moves into the area where rewards are high (representing high efficient).

set up. A complete learning session in the robot usually takes about one week (8 hours/day).

Both learning frameworks shown in Figure 5.10 start from scratch (no prior experience). After 200 trials, we have noticed that the rate of success to failure when applying *QARL* is higher than without it. However, the rate of success can be increased by controlling the threshold of vigilance; this is the objective of the next section.

5.6.3 Adaptive Vigilance in Humanoid Walking

The vigilance threshold is auto-adjusted in order to have the same experience for success as for failure according to the following algorithm:

```
    $\forall S_{vig} \in [-D, +D]$    $(initialisation)$
    $if(N_s > N_f)\ then\ \ take\ risks : S_{vig} = S_{vig} - step$
$elseif(N_s < N_f)\ then\ avoid\ risks : S_{vig} = S_{vig} + step$
$else$                 $nochange$
```

Here, N_s and N_f denote the number of successful and failed trials respectively. *step* describes the change in vigilance threshold to have a desired behavior for risks.

It is defined by training, $step = 0.01$ in this study. D is the diameter of the space ($D = \sqrt{3}$ in the three dimensional unit space).

When the success to failure ratio is always less than 1, the threshold of vigilance will gradually increase until a new threshold value that leads to EWS activity for all randomly generated vectors (in our experiment, after 1000 samples have been rejected sequentially), i.e. no more vectors can realize the condition in the decision making phase when applied on the robot. As a consequence, the threshold of vigilance decreased a step then starts the search with random vectors in the space. Decreasing S_{vig} will find executable samples in the space that can be applied on the robot to achieve a trial.

Figure 5.11 shows the rate of success and the rate of failure in learning to walk on flat terrain with and without vigilance adaptation. Note that the success to failure ratio N_s/N_f shows unpredicted changes in the beginning of learning. After 100 learning trials, due to the vigilance adaptation this ratio stays around 1, which contributes to the convergence of the success and failure maps. In case of non-adaptive vigilance, S_{vig} was fixed experimentally to -0.15, the ratio stabilizes at 0.65. Adapting the vigilance ensures having same size of training sets to learn success map and failure map, because both maps have same number of neurons (clusters).

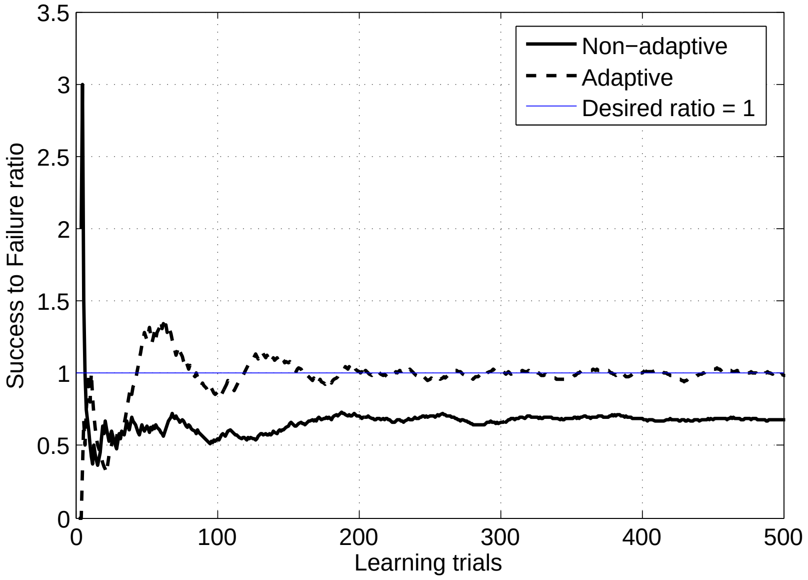
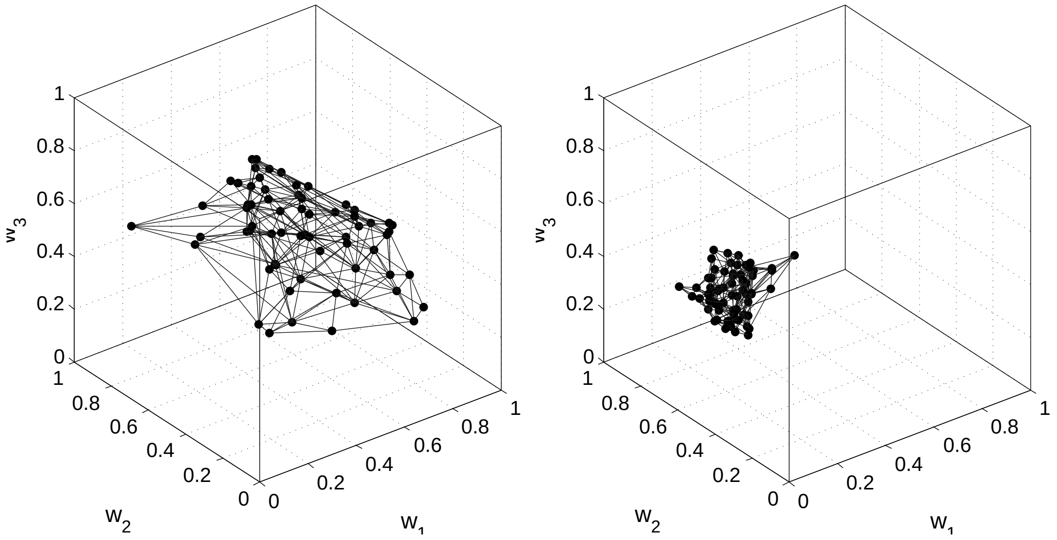


Figure 5.11: Success to failure ratio with and without adaptive vigilance in learning to walk on flat terrain.

5.7 Learning to Walk on Sloped Terrain

In this section, transfer of learning between different walking conditions (flat, uphill, downhill) is not addressed. We assume that there is a success map and a failure map for each situation. Two stages of learning have been implemented on 10° upward slope and 10° downward slope. For each condition, learning starts from scratch. Vigilance adaptation and *QARL* concepts are used. The initial angular positions have same values for all learning stages. Only ankle pitch joint initialized from stance position in order to keep the torso pitch around 10° vertically during walking.

Figure 5.12 shows success maps after learning to walk downhill on the left, and uphill on the right. The two maps and the map responsible of walking on flat terrain (Figure 5.10, right) occupy different areas in the learning space. Note that, the success map for walking downhill occupies greater area in the state space than the area occupied by the success map for walking uphill. However, that difference in the size does not mean the result is much better, it is mostly be related to the complexity of the task. (e.g. Walking uphill being more difficult than walking downhill, therefore the pattern space for uphill condition is smaller than the pattern space for the downhill condition).



(a) 10 ° downward slope.

(b) 10 ° upward slope.

Figure 5.12: Success map after learning with reward on sloped terrain.

Figure 5.13 shows the rate of success and the rate of failure in learning to walk on inclined uphill terrain with and without vigilance adaptation.

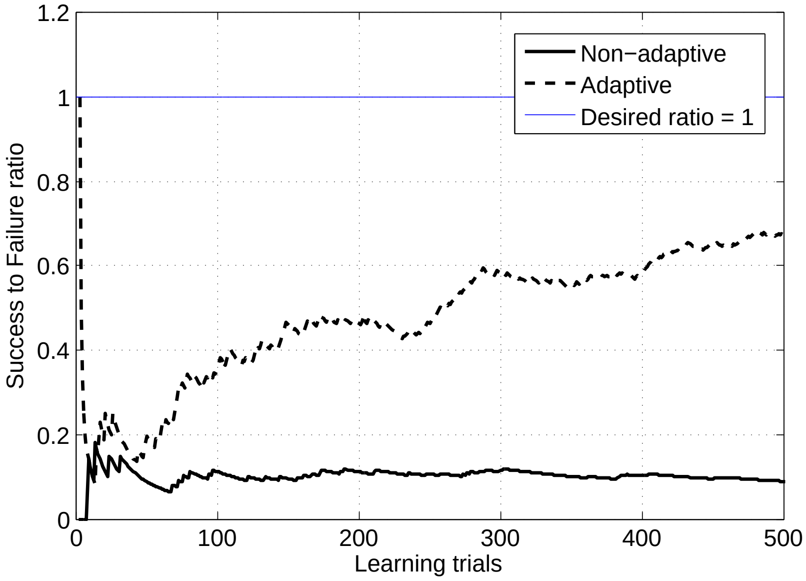


Figure 5.13: Success to failure ratio with and without adaptive vigilance in learning to walk on 10 ° upword slope.

Vigilance Adaptation

Vigilance is auto-adjusted in order to have the same experience for success as for failure ($N_s = N_f$), this ensure each map has enough data for training. In case of fixed vigilance $S_{vig} = -0.15$, the ratio stabilized around 0.1. This leads to converge only the failure map unlike the success map. The difference between this steady value and that with walking on flat terrain proves that learn walking uphill is more difficult than learn walking on the flat. To assure success map convergence without vigilance adaptation, too many learning trials are needed, therefore, this delays the convergence. Due to the vigilance adaptation this ratio look moving toward 1, even some more learning trials is needed to reach the wanted ratio.

5.8 Switching Between Different Sloped Terrains – Exploiting Learnt Experiences

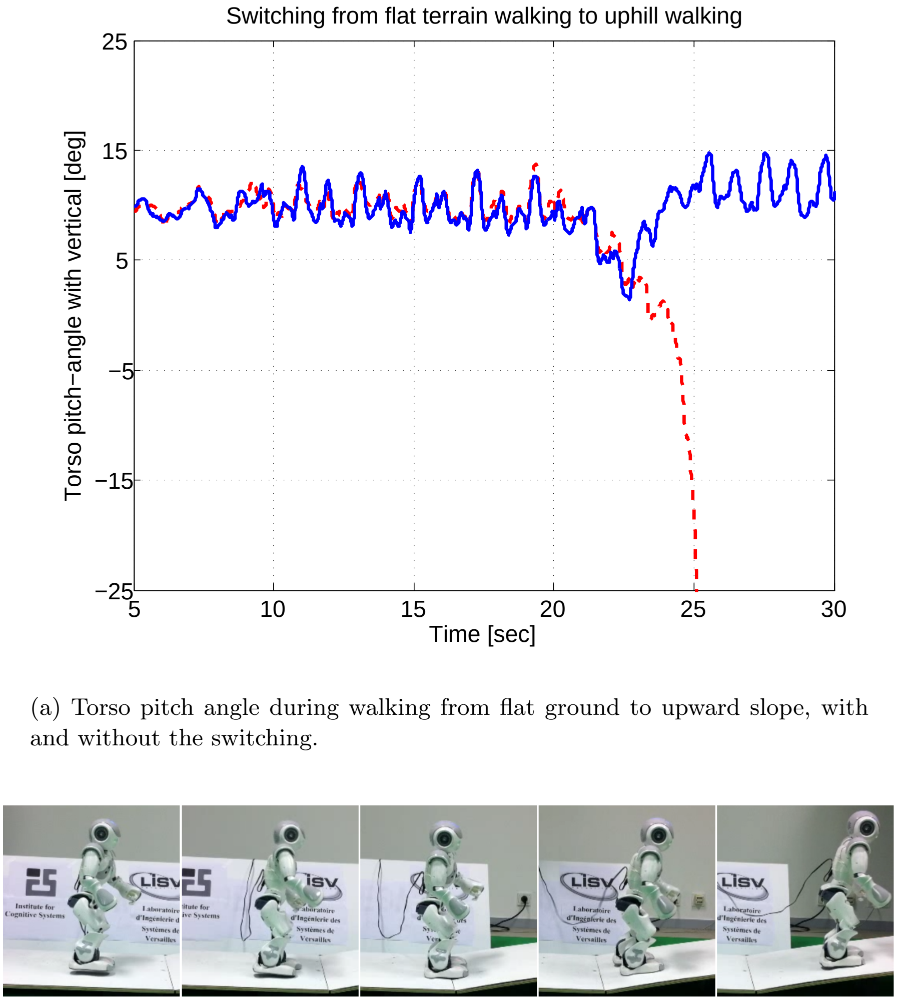
This stage shows walking on different terrains slopes and switching between them, which exploited previously learnt experiences. Inertial sensors are used to detect the change of terrain slope during walking. Detection occurs when the torso reaches previously defined threshold in Sagittal plane, this threshold was defined with taking into account the oscillation range of the torso in walking on flat terrain. After detecting the changes in torso oscillation range, the walking pattern switches from a success map related to the walk on previous terrain slope to a success map related to the walk on the new terrain slope.

The inertial sensor is also used to adjust the center of oscillation of ankle joints in order to keep the robot torso close to the vertical with a small inclination in the walking sense. For the NAO robot, we keep this angle closed to 10 ° with the vertical direction.

Figure 5.14(a) shows torso pitch angle during walking on different slopes, switching from flat to an uphill inclined terrain. Without using this technique the robot falls (indicated by the dashed line). As a compensation technique, switching occurs between a neuron in success map responsible of walking on flat terrain into a neuron of another success map responsible of walking on flat terrain. Therefore, the robot succeeds to continue walking on the new uphill terrain. Figure 5.15(a) shows torso pitch angle during switching from downhill to flat terrain. When the torso pitch angle reaches a pre-defined threshold, switching occurs gradually between a neuron of success map responsible for walking on downhill into a neuron of success map responsible for walking on flat terrain. Figure 5.14(b) and Figure 5.15(b) show snapshots for a NAO humanoid robot while achieving the walking task on different terrain slope and switching between them. (a video is available on: `http://web.ics.ei.tum.de/~nassour/naowalking.wmv`.)

5.9 Conclusion

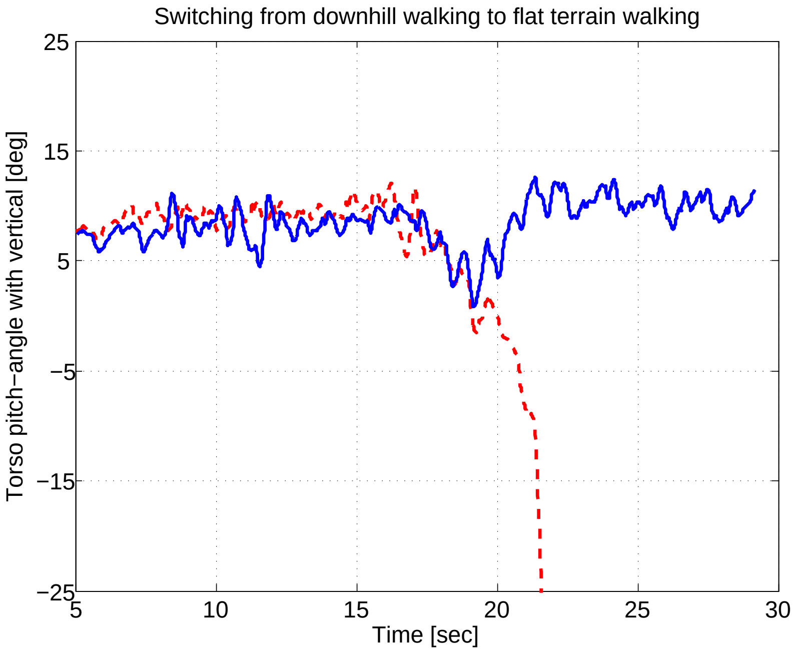
This chapter improves the success-failure learning algorithm. The notion of qualitative adaptive reward was introduced in order to simultaneously learn and optimize. The objectives of the mechanism were to learn from mistakes and to avoid making them again. This was done by building on experiences of past mistakes and successes. We showed how these two experiences could build themselves through the stages of evaluation, decision and then trials. Learning successful trials with reward related walking efficiency make success map match trials where efficiency is high. The ratio of the highest efficiency neurons in both success maps (with/without QARL) has increased by 60, while the number of successful trials has increased 16%. The adaptive vigilance threshold allows having an experience to success as to failure. It can be said that the negative reward is as important as the positive reward. This mechanism was implemented and validated on a NAO humanoid robot which allowed the robot to learn walking on flat as well as sloped terrain. Unlike the offline techniques, with our approach, learning has been done directly on the robot; it does not require thousands of trials in simulation that may not be able to match the dynamics of the real robot.



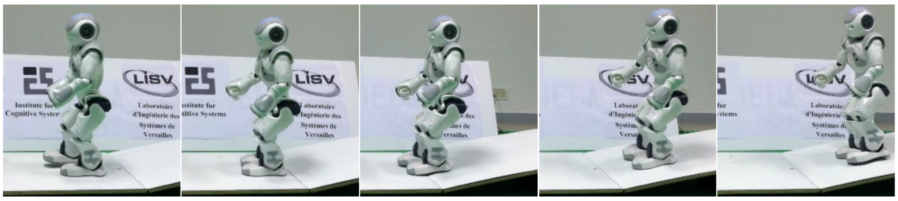
(a) Torso pitch angle during walking from flat ground to upward slope, with and without the switching.

(b) Walking from flat ground to upward slope with switching between success maps neurons.

Figure 5.14: Walking on different sloped terrain and switching from a flat into an uphill terrain. Switching occurs between success maps neurons in order to adapt to the new situation.



(a) Torso pitch angle during walking from downward slope to flat ground, with and without the switching.



(b) Walking from downward slope to flat ground with switching between success maps neurons.

Figure 5.15: Walking on different sloped terrain and switching from a downhill into a flat terrain. Switching occurs between success maps neurons in order to adapt to the new situation.

Chapter 6

Conclusion

This thesis has integrated several key findings from different research fields: machine learning, cognitive neuroscience, psychology, and robotics. We showed how human brain research can be used to investigate brain-inspired computational learning techniques and then implemented them in robots. Two main frameworks were presented.

First, a learning mechanism based on learning from previous successful and failed experiences was introduced (Chapter 3). A memory for success and a memory for failure were constructed through the stages of evaluation, decision and performing trials. This was motivated through scientific understanding of the Anterior Cingulate Cortex (ACC) of the human brain. We showed how negative reward in learning is as important as the positive. The concept of *vigilance* was proposed to manage the behavior in risk taking by switching between exploration and exploitation. This learning mechanism was introduced to learn intrinsic parameters of a low-level controller that drives a simulated planar biped robot to learn a walking task. Walking was learned on both sloped and flat terrain. The success-failure learning is then improved to learn reward in an adaptive manner with an automatically adapted vigilance. This was done by the inspiration of the Orbitofrontal Cortex (OFC) functionality in coding reward in an adaptive way (Chapter 5).

The robot was able to learn to walk and optimize its performance simultaneously. Learning successful trials with reward related to walking efficiency produces a success map that matches trials where efficiency is high. The adaptive vigilance threshold can be used to trade-off experiences of success and failure to further assist learning.

These neural mechanisms were implemented and validated on a NAO humanoid robot, which allowed the robot to learn walking on flat as well as sloped terrain in a real 3D environment. A key advantage of this learning technique is that it allows direct application on the robot and thus does not require thousands of trials in simulation a priori – removing the limitations imposed by matching the dynamics of the real robot and its environment in a simulator.

The second framework presented in this thesis consists of a multi-layered multi-pattern Central Pattern Generator model (MLMP-CPG), introduced in Chapter 4 and validated first with a planar biped. The MLMP-CPG model is generalized in Chapter 5 to a real 3 Dimensional humanoid robot. This efficient biologically-inspired locomotion controller can produce different motion patterns. Some of these can be used for fast body reactions such as stabilization responses to perturbations. A dimension reduction technique was introduced to bring together the parameter space into a space of patterns, thus yielding a rich pattern generator.

In conclusion, the Success-Failure-Learning-based Qualitative Adaptive Reward proposed in this thesis was inspired by the ACC and OFC functionalities to detect mistakes and to increase efficiency through experiences. This learning model was exploited to learn different physical tasks where periodic and discrete patterns were based on an extended CPG model. Finally this enabled a real 3D humanoid robot to learn intrinsic parameters at the low level controller while validating these neural models in a real world setting.

Discussion & future perspectives

The proposed learning mechanism in this thesis is not limited to locomotion tasks. We envision the success-failure learning as an experience based mechanism that can be used for manipulation of upper body parts, and may even lead to ethical decision making or any mental and physical skills that distinguish between two opposite situations (e.g., success and failure). An extension to this work is already underway. Additionally, so far only the success was quantified; introducing a technique to evaluate the failure quality may pose an additional dimension for future research on risk-level management during learning.

The Success-Failure Learning framework presented in this thesis is not just a solution intended for dealing with a specific problem or learning only a specific task (ad hoc). We envisioned that it is a critical part of the learning cycle within a complete cognitive architecture, enabling it to learn and acquire different physical and mental abilities.

This thesis presented a careful union of neuroscience research and robotics. Such an approach sets the stage for a societal contribution that is strongly based on science and engineering development.

Bibliography

[Abernethy, 2005] Abernethy, B. (2005). *The biophysical foundations of human movement.* HUMAN KINETICS, second edition. 122

[Ahn and Picard, 2005] Ahn, H. and Picard, R. W. (2005). Affective-cognitive learning and decision making: A motivational reward framework for affective agent. In *The 1st International Conference on Affective Computing and Intelligent Interaction. October 22-24*. 62, 94

[Albus, 1975] Albus, J. S. (1975). A new approach to manipulator control: the cerebellar model articulation controller (cmac. *Journal of Dynamic Systems, Measurement, and Control*, 97:220–227. 1, 26, 28, 46

[Amrollah and Henaff, 2010] Amrollah, E. and Henaff, P. (2010). On the role of sensory feedbacks in rowat-selverston cpg to improve robot legged locomotion. *Frontiers in Neurorobotics*, 4(00113). 39, 85

[Arbib, 2006] Arbib, M. A. (2006). *Action to Language via the Mirror Neuron System.* Cambridge University Press. 17

[Banich and Compton, 2010] Banich, M. T. and Compton, R. J. (2010). *Cognitive Neuroscience.* Wadsworth Publishing, 3 edition. 1, 15, 26, 27

[Barlow, 1989] Barlow, H. B. (1989). Unsupervised Learning. *Neural Computation*, 1(3):295–311. 17

[Beer, 1995] Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adapt. Behav.*, 3:469–509. 24

[Beer et al., 1992] Beer, R. D., Chiel, H. J., Quinn, R. D., Espenschied, K. S., and Larsson, P. (1992). A distributed neural network architecture for hexapod robot locomotion. *Neural Computation*, 4(3):356–365. 65

[Berryman et al., 2011] Berryman, N., Gayda, M., Nigam, A., Juneau, M., Bherer, L., and Bosquet, L. (2011). Comparison of the metabolic energy cost of overground and treadmill walking in older adults. *European Journal of Applied Physiology*, pages 1–8. 123

[Bicho et al., 2011] Bicho, E., Erlhagen, W., Louro, L., and e Silva, E. C. (2011). Neurocognitive mechanisms of decision making in joint action: A human-robot interaction study. *Human Movement Science*, 30(5):846 – 868. 17, 114

[Bigus, 1996] Bigus, J. P. (1996). *Data Mining With Neural Networks: Solving Business Problems from Application Development to Decision Support.* Mcgraw-Hill (Tx). 20

[Bishop, 1996] Bishop, C. M. (1996). *Neural Networks for Pattern Recognition.* Oxford University Press, USA, 1 edition. 20

[Bracewell et al., b 22] Bracewell, R. M., Balasubramaniam, R., and Wing, A. M. (2005 Feb 22). Interlimb coordination deficits during cyclic movements in cerebellar hemiataxia. *Neurology*, 64(4):751–752. 89

[Brown, 1911] Brown, G. T. (1911). The intrinsic factors in the act of progression in the mammal. *Proceedings of the Royal Society of London*, 84(572):308–319. 31, 32, 80, 121

[Brown and Braver, 2005] Brown, J. W. and Braver, T. S. (2005). Learned predictions of error likelihood in the anterior cingulate cortex. *Science*, 307:1118–1121. 3, 16, 56, 57, 58, 59

[Brown and Braver, 2008] Brown, J. W. and Braver, T. S. (2008). A computational model of risk, conflict, and individual difference effects in the anterior cingulate cortex. *Brain research*, 1202:99 –108. 3, 56, 57, 58, 59, 110

[Brown, 1914] Brown, T. G. (1914). On the fundamental activity of the nervous centres: together with an analysis of the conditioning of rhythmic activity in progression, and a theory of the evolution of function in the nervous system. *J. Physiol.*, 48(1):18–46. 32, 34

[Buxton, 2009] Buxton, R. B. (2009). *Introduction to Functional Magnetic Resonance Imaging: Principles and Techniques*. Cambridge University Press, 2 edition edition. 16

[Byrne, 2003] Byrne, R. W. (2003). Imitation as behaviour parsing. *Philosophical Transactions of the Royal Society of London - Series B: Biological Sciences*, 358(1431):529–536. 25

[Cheng et al., 2007] Cheng, G., Hyon, S.-H., Morimoto, J., Ude, A., Hale, J. G., Colvin, G., Scroggin, W., and Jacobsen, S. C. (2007). Cb: a humanoid research platform for exploring neuroscience. *Advanced Robotics*, 21(10):1097–1114. 7, 112

[Chew and Pratt, 2002] Chew, C.-M. and Pratt, G. A. (2002). Dynamic bipedal walking assisted by learning. *Robotica*, 20(5):477–491. 112

[Choi and Bastian, 2007] Choi, J. T. and Bastian, A. J. (2007). Adaptation reveals independent control networks for human walking. *Nature Neuroscience*, 10(8):1055–1062. 32

[Cohen and Boothe, 1999] Cohen, A. H. and Boothe, D. L. (1999). Sensorimotor interactions during locomotion : Principles derived from biological systems. *Autonomous Robots*, 245(3):239–245. 47

[Cohen et al., 2005] Cohen, M., Heller, A., and Ranganath, C. (2005). Functional connectivity with anterior cingulate and orbitofrontal cortices during decision-making. *Cognitive Brain Research*, 23(1):61 – 70. 6, 16, 110, 111, 114, 115

[Cruse et al., 1995] Cruse, H., Bartling, C., Dreifert, M., Schmitz, J., Brunn, D., Dean, J., and Kindermann, T. (1995). Walking: a complex behavior controlled by simple systems. *Adaptive Behavior*, 3(4):385–418. 63

[D. P. Kroese and Botev, 2011] D. P. Kroese, T. T. and Botev, Z. I. (2011). *Handbook of Monte Carlo Methods*. John Wiley & Sons, New York,. 52

[Doya, 2008] Doya, K. (2008). Modulators of decision making. *Nature Neuroscience*, 11:410 – 416. 114

[Eiben and Smith, 2008] Eiben, A. E. and Smith, J. (2008). *Introduction to Evolutionary Computing*. Springer. 52

[Elman, 1990] Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211. 24

[Endo et al., 2008] Endo, G., Morimoto, J., Matsubara, T., Nakanishi, J., and Cheng, G. (2008). Learning cpg-based biped locomotion with a policy gradient method: Application to a humanoid robot. *International Journal of Robotics Research*, 27:213–228. 2, 31, 34, 46, 47, 48, 49, 73, 80, 82, 83, 118, 119, 121

[Endo et al., 2004] Endo, G., Morimoto, J., Nakanishi, J., and Cheng, G. (2004). An empirical exploration of a neural oscillator for biped locomotion control. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004, April 26 - May 1, 2004, New Orleans, LA, USA*, pages 3036–3042. 2, 34, 35, 42, 43, 44, 82, 83

[EUCogIII, 2012] EUCogIII (2012). First eucogiii members conference - "do robots need cognition? - does cognition need robots?" 23-24 february, vienna. 17

[Fairhall and Bialek, 2002] Fairhall, A. and Bialek, W. (2002). *Adaptive spike coding, in The Handbook of Brain Theory and Neural Networks.* MIT Press, Cambridge, second edition edition. 110

[Franklin et al., 1998] Franklin, S., Kelemen, A., and McCauley, L. (1998). Ida: A cognitive agent architecture. *In IEEE Conf on Systems, Man and Cybernetics. IEEE Press.* 14

[Gallagher et al., 1996] Gallagher, J. C., Beer, R. D., Espenschied, K. S., and Quinn, R. D. (1996). Application of evolved locomotion controllers to a hexapod robot. *Robotics and Autonomous Systems*, 19(1):95–103. 24

[Gehring et al., 1990] Gehring, W. J., Coles, M. G. H., Meyer, D. E., and Donchin, E. (1990). The error-related negativity: An event-related potential accompanying errors. *Journal of Psychophysiology*, 27:S34. 56, 57

[Gehring et al., 2012] Gehring, W. J., Liu, Y., Orr, J. M., and Carp, J. (2012). *Oxford handbook of event-related potential components*, chapter The error-related negativity (ERN/Ne), pages 231–291. New York: Oxford University Press. 57

[Gemba et al., 1986] Gemba, H., Sasaki, K., and Brooks, V. (1986). 'error'potentials in limbic cortex (anterior cingulate area 24) of monkeys during motor learning. *Neuroscience Letters*, 70(2):223 – 227. 56

[Geng et al., 2005] Geng, T., Porr, B., and Wörgötter, F. (2005). Fast biped walking with a reflexive controller and real-time policy searching. In *Neural Information Processing Systems, NIPS 2005, December 5-8, Vancouver, British Columbia, Canada.* 42

[Geng et al., 2006] Geng, T., Porr, B., and Wörgötter, F. (2006). Fast biped walking with a sensor-driven neuronal controller and real-time online learning. *International Journal of Robotics Research*, 25:243–259. 56, 63, 65, 66, 87, 91, 116

[Ghiasi et al., 2010] Ghiasi, A. R., Alizadeh, G., and Mirzaei, M. (2010). Simultaneous design of optimal gait pattern and controller for a bipedal robot. *Multibody System Dynamics*, 23(4):401–429. 52

[Gomes, 2011] Gomes, R. (2011). *Towards open ended learning: budgets, model selection, and representation.* PhD thesis, California Institute of Technology. 16

[Gonzalez-Aguirre et al., 2011] Gonzalez-Aguirre, D., Asfour, T., and Dillmann, R. (2011). Towards stratified model-based environmental visual perception for humanoid robots. *Pattern Recognition Letters*, 32(16):2254 – 2260. 17

[Gouaillier et al., 2009] Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., and Maisonnier, B. (2009). Mechatronic design of nao humanoid. In *IEEE International Conference on Robotics and Automation, ICRA, Japan*, pages 769–774. 112

[Grudic et al., 2003] Grudic, G., Kumar, V., and Ungar, L. H. (2003). Using policy gradient reinforcement learning on autonomous robot controllers. *October*, (October):406–411. 73, 119

[Guyton and Hall, 2006] Guyton, A. C. and Hall, J. E. (2006). *Textbook of medical physiology*. Elsevier Saunders, 11 edition. 123

[Hang and Woon, 1997] Hang, C. C. and Woon, L. C. (1997). Adaptive neural network control of robot manipulators in task space. *IEEE Transactions on Industrial Electronics*, 44(6):746–752. 20

[Hase and Yamazaki, 1999] Hase, K. and Yamazaki, N. (1999). Computational evolution of human bipedal walking by a neuro-musculo-skeletal model. *Artificial Life and Robotics*, 3:133–138. 10.1007/BF02481128. 52

[Hebb, 1949] Hebb, D. O. (1949). *The organization of behavior: A neuropsychological theory*. John Wiley And Sons, Inc., New York. 21, 22

[Hohnsbein et al., 1989] Hohnsbein, J., Falkenstein, M., and Hoorman, J. (1989). Error processing in visual and auditory choice reaction tasks. *Journal of Psychophysiology*, 3:32. 56

[Hoinville, 2007] Hoinville, T. (2007). *Évolution de contrôleurs neuronaux plastiques : de la locomotion adaptée vers la locomotion adaptative*. PhD thesis, University of Versailles St Quentin, Vélizy, France. 39, 85

[Hoinville et al., 2011] Hoinville, T., Siles, C. T., and Hénaff, P. (2011). Flexible and multistable pattern generation by evolving constrained plastic neurocontrollers. *Adaptive Behavior*, 19:187–207. 24, 65

[Hooper, 2000] Hooper, S. (2000). Central pattern generators. *Current Biology*, 10:176–177. 121

[Hopfield, 1982] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational properties. *Proc. Nat. Acad. Sci. (USA)*, 79:2554–2558. 24

[Horvath and Zuckerman, 1993] Horvath, P. and Zuckerman, M. (1993). Sensation seeking, risk appraisal, and risky behavior. *Personality and Individual Differences*, 14(1):41–52. 62

[Ijspeert, 2008] Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21(4):642–653. 20, 31, 80, 121

[Ijspeert et al., 2007] Ijspeert, A. J., Crespi, A., Ryczko, D., and Cabelguen, J.-M. (2007). From swimming to walking with a salamander robot driven by a spinal cord model. *Science*, 315(5817):1416–1420. 2, 34, 40, 41, 65, 82, 88

[Inada and Ishii, 2003] Inada, H. and Ishii, K. (2003). Behavior generation of bipedal robot using central pattern generator(cpg) (1st report: Cpg parameters searching method by genetic algorithm). *Proceedings 2003 IEEERSJ International Conference on Intelligent Robots and Systems IROS 2003 Cat No03CH37453*, 3(October):2179–2184. 3, 52, 53

[Ishiguro et al., 2003] Ishiguro, A., Fujii, A., Hotz, P. E., Ishiguro, A., Fujii, A., and Hotz, P. E. (2003). Neuromodulated control of bipedal locomotion using a polymorphic cpg circuit. *Adaptive Behavior*, 11:2003. 65

[Ivanenko et al., 2007] Ivanenko, Y. P., Cappellini, G., Dominici, N., Poppele, R. E., and Lacquaniti, F. (2007). Modular control of limb movements during human locomotion. *The Journal of Neuroscience*, 27(41):11149–11161. 80

[Iversen and Mishkin, 1970] Iversen, S. D. and Mishkin, M. (1970). Perseverative interference in monkeys following selective lesions of the inferior prefrontal convexity. *Experimental Brain Research*, 11:376–386. 110

[Jain et al., 1996] Jain, A. K., Mao, J., and Mohiuddin, K. (1996). Artificial neural networks: A tutorial. *IEEE Computer*, 29:31–44. 20, 23

[Jordan, 1990] Jordan, M. I. (1990). Attractor dynamics and parallelism in a connectionist sequential machine. *IEEE Computer Society Neural Networks Technology Series*, pages 112–127. 24

[Katić and Vukobratović, 2004] Katić, D. and Vukobratović, M. (2004). Control algorithm for biped walking using reinforcement learning. *2nd Serbian-Hungarian Joint Symposium on Intelligent Systems*. 112

[Kiehn and Butt, 2003] Kiehn, O. and Butt, S. J. (2003). Physiological, anatomical and genetic identification of cpg neurons in the developing mammalian spinal cord. *Progress in Neurobiology*, 70(4):347 – 361. 82, 121

[Kimura et al., 1999] Kimura, H., Akiyama, S., and Sakurama, K. (1999). Realization of dynamic walking and running of the quadruped using neural oscillator. *Autonomous Robots*, 7:247–258. 31, 80, 121

[Kimura H, 2007] Kimura H, Fukuoka Y, C. A. (2007). Biologically inspired adaptive walking of a quadruped robot. *Philosophical Transactions of The Royal Society*, 365(1850):153 – 170. 47

[Kobayashi et al., 2010] Kobayashi, S., de Carvalho, O. P., and Schultz, W. (2010). Adaptation of reward sensitivity in orbitofrontal neurons. *The Journal of Neuroscience*, 30(2):534–544. 110

[Kohonen, 1984] Kohonen, T. (1984). *Self-Organization and Associative Memory*. Springer Verlag, Berlin. 27, 59

[Kohonen, 1988] Kohonen, T. (1988). An introduction to neural computing. *Neural Networks*, 1(1):3 – 16. 20

[Kohonen, 1995] Kohonen, T. (1995). *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg. 59, 94

[Koshland and Smith, 1989] Koshland, G. F. and Smith, J. L. (1989). Mutable and immutable features of paw-shake responses after hindlimb deafferentation in the cat. *Journal of Neurophysiology*, 62(1):162–73. 33

[Koza, 1994] Koza, J. R. (1994). *Genetic Programming II: Automatic Discovery of Reusable Programs*. A Bradford Book, first edition edition. 52

[Kuniyoshi et al., 2004] Kuniyoshi, Y., Yorozu, Y., Ohmura, Y., Terada, K., Otani, T., Nagakubo, A., and Yamamoto, T. (2004). From humanoid embodiment to theory of mind. In Iida, F., Pfeifer, R., Steels, L., and Kuniyoshi, Y., editors, *Embodied Artificial Intelligence*, volume 3139 of *Lecture Notes in Computer Science*, pages 202–218. Springer Berlin / Heidelberg. 17

[Kuo, 2002] Kuo, A. D. (2002). The relative roles of feedforward and feedback in the control of rhythmic movements. *Motor Control*, 6:129–145. 81, 121

[Lafreniere-Roula and McCrea, 2005] Lafreniere-Roula, M. and McCrea, D. A. (2005). Deletions of rhythmic motoneuron activity during fictive locomotion and scratch provide clues to the organization of the mammalian central pattern generator. *Journal of Neurophysiology*, 94(2):1120–1132. 88

[Lee and Oh, 2008] Lee, J. and Oh, J. H. (2008). Walking pattern generation for planar biped walking using q-learning. *Science And Technology*, pages 3027–3032. 2, 50

[Lee and Oh, 2009] Lee, J. and Oh, J. H. (2009). Biped walking pattern generation using reinforcement learning. *International Journal of Humanoid Robotics*, 06(01):1. 49

[Lewis et al., 1992] Lewis, M. A., Fagg, A. H., and Solidum, A. (1992). Genetic programming approach to the construction of a neural network for control of a walking robot. *Proceedings 1992 IEEE International Conference on Robotics and Automation*, 3:2618–2623. 47

[Li et al., 2011] Li, T.-H. S., Su, Y.-T., Lai, S.-W., and Hu, J.-J. (2011). Walking motion generation, synthesis, and control for biped robot by using pgrl, lpi, and fuzzy logic. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 41(3):736–748. 3, 50, 51, 73, 118, 119

[Lin et al., 2006] Lin, C.-M., Fan, W.-C., Chen, C.-H., and Hou, Y.-L. (2006). Robust control for biped robot using cerebellar model articulation controller. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2006, part of the IEEE World Congress on Computational Intelligence, WCCI 2006, Vancouver, BC, Canada, 16-21 July 2006*, pages 2485–2490. 2, 46, 47

[Lipson et al., 2007] Lipson, H., Bongard, J., Zykov, V., and Malone, E. (2007). Evolutionary robotics for legged machines : From simulation to physical reality. *Nature*, 17(9):R330–R332. 52

[Liu et al., 2008] Liu, G., Habib, M., Watanabe, K., and Izumi, K. (2008). Central pattern generators based on matsuoka oscillators for the locomotion of biped robots. *Artificial Life and Robotics*, 12(1):264–269. 2, 35, 36, 83

[Liu et al., 2007] Liu, G. L., Habib, M., Watanabe, K., and Izumi, K. (2007). Cpg based control for generating stable bipedal trajectories under external perturbation. In *SICE, 2007 Annual Conference*, pages 1019–1022. 83

[Ludovic et al., 2009] Ludovic, R., Jonas, B., and Jan, I. A. (2009). Adaptive frequency oscillators and applications. *The Open Cybernetics Systemics Journal*, 3(2):64–69. 80, 121

[MacKay, 2003] MacKay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms*, chapter Supervised Learning in Multilayer Networks, pages 527–533. Cambridge University Press. 23

[Maeda, 2002] Maeda, Y. (2002). Modified q-learning method with fuzzy state division and adaptive rewards. In *Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on*, volume 2, pages 1556 –1561. 50

[Manoonpong et al., 2007] Manoonpong, P., Geng, T., Kulvicius, T., Porr, B., and Wörgötter, F. (2007). Adaptive, fast walking in a biped robot under neuronal control and learning. *PLoS Comput Biol*, 3(7):e134. 2, 24, 34, 42, 43, 101

[Marder and Bucher, 2001] Marder, E. and Bucher, D. (2001). Central pattern generators and the control of rhythmic movements. *Current Biology*, 11(23):R986 – R996. 4, 85

[Marder and Calabrese, 1996] Marder, E. and Calabrese, R. L. (1996). Principles of rhythmic motor pattern generation. *Physiol. Rev.*, 76(3):687–717. 65, 81

[Margaria, 1976] Margaria, R. (1976). *Biomechanics and energetics of muscular exercise*. Oxford University Press, USA, first edition edition. 123

[Markin et al., 2010] Markin, S. N., Klishko, A. N., Shevtsova, N. A., Lemay, M. A., Prilutsky, B. I., and Rybak, I. A. (2010). Afferent control of locomotor cpg: insights from a simple neuromechanical model. *Ann N Y Acad Sci*, 1198:21–34. 82

[Mars et al., 2005] Mars, R. B., Coles, M. G., Grol, M. J., Holroyd, C. B., Nieuwenhuis, S., Hulstijn, W., and Toni, I. (2005). Neural dynamics of error processing in medial frontal cortex. *Neuroimage*, 28(4):1007–1013. 56, 111

[Matsubara et al., 2006] Matsubara, T., Morimoto, J., Nakanishi, J., aki Sato, M., and Doya, K. (2006). Learning cpg-based biped locomotion with a policy gradient method. *Robotics and Autonomous Systems*, 54(11):911–920. 35, 83

[Matsumoto et al., 2003] Matsumoto, K., Suzuki, W., and Tanaka, K. (2003). Neuronal correlates of goal-based motor selection in the prefrontal cortex. *Science*, 301(5630):229–32. 59

[Matsuoka, 1985] Matsuoka, K. (1985). Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biological Cybernetics*, 52(6):367–376. 34, 80, 82, 121

[McCrea and Rybak, 2008] McCrea, D. A. and Rybak, I. A. (2008). Organization of mammalian locomotor rhythm and pattern generation. *Brain research reviews*, 57(1):134–46. 31, 33, 63, 65, 80, 85, 88, 91, 121

[McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus the ideas immanent in nervous activity. *Bulletin Mathematical Biophysics*, 5:115–33. 21

[McMillen et al., 1999] McMillen, D. R., D'Eleuterio, G. M., and Halperin, J. R. (1999). Simple central pattern generator model using phasic analog neurons. *Physical Review E Statistical Physics Plasmas Fluids And Related Interdisciplinary Topics*, 59(6):6994–6999. 80, 82, 121

[Mendel and McLaren, 1970] Mendel, J. M. and McLaren (1970). *Adaptive, Learning and Pattern Recognition Systems: Theory and Applications*, chapter Reinforcement Learning Control and Pattern Recognition Systems, pages 287– 318. Academic Press, New York. 20

[Metropolis and Ulam, 1949] Metropolis, N. and Ulam, S. (1949). The monte carlo method. *Journal of the American Statistical Association*, 44:335 – 341. 51

[Morimoto et al., 2008] Morimoto, J., Endo, G., Nakanishi, J., and Cheng, G. (2008). A biologically inspired biped locomotion strategy for humanoid robots: Modulation of sinusoidal patterns by a coupled oscillator model. *IEEE Transactions on Robotics*, 24:185–191. 31, 80, 121

[Morimoto et al., 2006] Morimoto, J., Endo, G., Nakanishi, J., Hyon, S.-H., Cheng, G., Bentivegna, D. C., and Atkeson, C. G. (2006). Modulation of simple sinusoidal patterns by a coupled oscillator model for biped walking. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, May 15-19, Orlando, Florida, USA*, pages 1579–1584. 2, 44, 45

[Morimoto et al., 2005] Morimoto, J., Nakanishi, J., Endo, G., Cheng, G., Atkeson, C. G., and Zeglin, G. (2005). Poincaré-map-based reinforcement learning for biped walking. *IEEE International Conference on Robotics and Automation, ICRA, Barcelona*, pages 2381–2386. 112

[Nakamura et al., 1998] Nakamura, K., Sakai, K., and Hikosaka, O. (1998). Neuronal activity in medial frontal cortex during learning of sequential procedures. *J Neurophysiol*, 80(5):2671–2687. 26

[Nakanishi et al., 2004] Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., and Kawato, M. (2004). Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems*, 47:79–91. 80, 121

[Namikawa et al., 2011] Namikawa, J., Nishimoto, R., and Tani, J. (2011). A neurodynamic account of spontaneous behaviour. *PLoS Computational Biology*, 7(10). 1, 24, 25, 26

[Nascimento, 1994] Nascimento, J. (February 1994). *Artificial Neural Networks for Control and Optimization.* PhD thesis, University of Manchester Institute of Science and Technology (UMIST), Control Systems Centre, Manchester, United Kingdom. 20

[Nassour et al., 2009] Nassour, J., Henaff, P., Ouezdou, F. B., and Cheng, G. (2009). Experience-based learning mechanism for neural controller adaptation: Application to walking biped robots. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 11-15, St. Louis, MO, USA*, pages 2616–2621. 116

[Neisser, 1967] Neisser, U. (1967). *Cognitive Psychology.* Prentice Hall, 1st edition edition. 15

[Orlovsky et al., 1999] Orlovsky, G., Deliagina, T., and Grillner, S. (1999). *Neuronal Control of Locomotion: From Mollusc to Man.* Oxford University Press. 31, 32, 80, 121

[Oubbati and Palm, 2009] Oubbati, M. and Palm, G. (2009). Self-motivated learning robot. *9th Epigenetic Robotics International Conference (Epirob09): Modeling Cognitive Development in Robotic Systems, Venice, Italy, November 12-14.* 17

[Patnaik, 2007] Patnaik, S. (2007). *Robot Cognition and Navigation An Experiment with Mobile Robots.* Springer. 1, 13, 14

[Pawlowski et al., 2008] Pawlowski, B., Atwal, R., and Dunbar, R. I. M. (2008). Sex differences in everyday risk-taking behavior in humans. *Evolutionary Psychology*, 6(1):29–42. 62

[Perret et al., 1988] Perret, C., Cabelguen, J., and Orsal, D. (1988). *Stance and Motion: Facts and Concepts*, chapter Analysis of the pattern of activity in “knee flexor” motoneurons during locomotion in the cat, pages 133–141. Plenum Press, New York. 32

[Peters et al., 2003] Peters, J., Vijayakumar, S., and Schaal, S. (2003). reinforcement learning for humanoid robotics. In *ieee-ras international conference on humanoid robots (humanoids2003).* 73, 119

[Pinto and Golubitsky, 2006] Pinto, C. M. A. and Golubitsky, M. (2006). Central pattern generators for bipedal locomotion. *Journal of Mathematical Biology*, 53(3):474–489. 121

[Purves et al., 2004] Purves, D., Augustine, G. J., Fitzpatrick, D., Hall, W. C., Lamantia, A.-S., McNamara, J. O., and Williams, S. M. (2004). *Neuroscience, 3rd Edition.* Sinauer Associates, Inc. 1, 31, 32, 89

[Ridderinkhof et al., 2004] Ridderinkhof, K. R., van den Wildenberg, W. P., Segalowitz, S. J., and Carter, C. S. (2004). Neurocognitive mechanisms of cognitive control: The role of prefrontal cortex in action selection, response inhibition, performance monitoring, and reward-based learning. *Brain and Cognition*, 56(2):129 – 140. 17

[Righetti et al., 2006] Righetti, L., Buchli, J., and Ijspeert, A. J. (2006). Dynamic hebbian learning in adaptive frequency oscillators. *Physica D*, 216(2):269–281. 80, 121

[Righetti and Ijspeert, 2006] Righetti, L. and Ijspeert, A. J. (2006). Programmable central pattern generators: an application to biped locomotion control. In *In Proceedings of the 2006 ieee international conference on robotics and automation*, pages 1585–1590. 2, 34, 40, 41, 82

[Rocha and Neves, 1999] Rocha, M. and Neves, J. (1999). Preventing premature convergence to local optima in genetic algorithms via random offspring generation. In *Proceedings of the 12th international conference on Industrial and engineering applications of artificial intelligence and expert systems: multiple approaches to intelligent systems*, IEA/AIE '99, pages 127–136, Secaucus, NJ, USA. Springer-Verlag New York, Inc. 52

[Rolls, 1996] Rolls, E. T. (1996). The orbitofrontal cortex. *Philosophical Transactions of the Royal Society B*, 351:1433–1444. 113

[Rolls, 2000] Rolls, E. T. (2000). The orbitofrontal cortex and reward. *Cerebral Cortex*, 10(3):284–294. 115

[Rosenblatt, 1957] Rosenblatt, F. (1957). The perceptron–a perceiving and recognizing automaton. *Report 85-460-1, Cornell Aeronautical Laboratory.* 22

[Rosenblatt, 1961] Rosenblatt, F. (1961). Principles of neurodynamics: Perceptrons and the theory of brain mechanisms. *Spartan Books, Washington DC.* 23

[Rossignol et al., 2006] Rossignol, S., Dubuc, R., and Gossard, J.-P. (2006). Dynamic sensorimotor interactions in locomotion. *Physiological Reviews*, 86(1):89–154. 121

[Rowat and Selverston, 1991] Rowat, P. and Selverston, A. (1991). Learning algorithms for oscillatory networks with gap junctions and membrane currents. *Network: Computation in Neural Systems*, 2(1):17–41. 4, 34, 63, 80, 82, 85, 121, 122

[Rowat and Selverston, 1997] Rowat, P. F. and Selverston, A. I. (1997). Oscillatory mechanisms in pairs of neurons connected with fast inhibitory synapses. *Journal of Computational Neuroscience*, 4(2):103–127. 2, 36, 37, 38, 39, 40, 83, 84

[Rubinstein and Kroese., 2004] Rubinstein, R. Y. and Kroese., D. P. (2004). *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning.* Springer-Verlag, New York. 52

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323:533 – 536. 23

[Rumelhart and Zipser, 1985] Rumelhart, D. E. and Zipser, D. (1985). Feature discovery by competitive learning. *Cognitive Science*, 9:75–112. 21

[Rybak et al., 2006] Rybak, I. A., Shevtsova, N. A., Lafreniere-Roula, M., and McCrea, D. A. (2006). Modelling spinal circuitry involved in locomotor pattern generation: insights from deletions during fictive locomotion. *The Journal of Physiology*, 577:617–639. 2, 31, 33, 34, 80, 82, 85, 100

[Sabourin et al., 2006] Sabourin, C., Bruneau, O., and Buche, G. (2006). Control strategy for the robust dynamic walk of a biped robot. *The International Journal of Robotics Research*, 25:843–860. 2, 46, 111

[Sakai et al., 1999] Sakai, K., Hikosaka, O., Miyauchi, S., Takino, R., Tamada, T., Iwata, N. K., and Nielsen, M. (1999). Neural representation of a rhythm depends on its interval ratio. *Journal of Neuroscience*, 19(22):10074–81. 26

[Salomon, 2003] Salomon, R. (2003). Biologically inspired robot behavior engineering. chapter Self-adapting neural networks for mobile robots, pages 173–197. Physica-Verlag GmbH, Heidelberg, Germany, Germany. 20

[Schmitt, 2004] Schmitt, L. M. (2004). Theory of genetic algorithms ii: models for genetic operators over the string-tensor representation of populations and convergence to global optima for arbitrary fitness function under scaling. *Theoretical Computer Science*, 310:181–231. 52

[Shik et al., 1966] Shik, M., Orlovsky, G., and Severin, F. (1966). Organization of locomotor synergism. *Biofizika*, 11(5):879–886. 82

[Singer et al., 2004] Singer, T., Seymour, B., O'Doherty, J., Kaube, H., Dolan, R. J., and Frith, C. D. (2004). Empathy for pain involves the affective but not sensory components of pain. *Science*, 303(5661):1157–1162. 56, 110

[Sivanandam, 2007] Sivanandam, S. N. (2007). *Introduction to Genetic Algorithms*. Springer, 1 edition. 52

[Snaider et al., 2012] Snaider, J., McCall, R., and Franklin, S. (2012). Time production and representation in a conceptual and computational cognitive model. *Cognitive Systems Research*, 13(1):59–71. 1, 14

[Sutton and Barto, 1998] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press. 45, 51, 112

[Tabandeh et al., 2006] Tabandeh, S., Clark, C., and Melek, W. (2006). A genetic algorithm approach to solve for multiple solutions of inverse kinematics using adaptive niching and clustering. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 1815 – 1822. 52

[Taga, 1998] Taga, G. (1998). A model of the neuro- musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance. *Biological Cybernetics*, 78(1):9–17. 92

[Taga, 2006] Taga, G. (2006). *Adaptive Motion of Animals and Machines*, chapter Nonlinear dynamics of human locomotion: from real-time adaptation to development, pages 189–204. Springer-Verlag, Tokyo. 121

[Taga et al., 1991] Taga, G., Yamaguchi, Y., and Shimizu, H. (1991). Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, 65(3):147–159. 2, 31, 34, 35, 37, 63, 80, 82

[Tani and Fukumura, 1994] Tani, J. and Fukumura, N. (1994). Learning goal-directed sensory-based navigation of a mobile robot. *Neural Networks*, 7(3):553 – 563. 20

[Thorpe et al., 1983] Thorpe, S. J., Rolls, E. T., and Maddison, S. (1983). The orbitofrontal cortex: Neuronal activity in the behaving monkey. *Experimental Brain Research*, 49:93–115. 110

[Tremblay and Schultz, 1999] Tremblay, L. and Schultz, W. (1999). Relative reward preference in primate orbitofrontal cortex. *Nature*, 398(6729):704–8. 5, 110, 113, 114

[Tremblay and Schultz, 2000] Tremblay, L. and Schultz, W. (2000). Modifications of reward expectation-related neuronal activity during learning in primate orbitofrontal cortex. *Journal of Neurophysiology*, 83(4):1877–85. 110

[van Gelder et al., 2009] van Gelder, J.-L., de Vries, R. E., and van der Pligt, J. (2009). Evaluating a dual-process model of risk: affect and cognition as determinants of risky choice. *Journal of Behavioral Decision Making*, 22(1):45–61. 62

[Van Leijenhorst et al., 2008] Van Leijenhorst, L., Westenberg, P. M., and Crone, E. A. (2008). A developmental study of risky decisions on the cake gambling task: Age and gender analyses of probability estimation and reward evaluation. *Developmental Neuropsychology*, 33(2):179–196. 56

[Van Veen et al., 2001] Van Veen, V., Cohen, J. D., Botvinick, M. M., Stenger, V. A., and Carter, C. S. (2001). Anterior cingulate cortex, conflict monitoring, and levels of processing. *NeuroImage*, 14(6):1302–1308. 115

[Vernon et al., 2007] Vernon, D., Metta, G., and Sandini, G. (2007). The icub cognitive architecture: Interactive development in a humanoid robot. *IEEE 6th International Conference on Development and Learning. ICDL*, pages 122–127. 1, 19

[Wadden and Ekeberg, 1998] Wadden, T. and Ekeberg, O. (1998). A neuro-mechanical model of legged locomotion: single leg control. *Biological Cybernetics*, 79(2):161–173. 63, 65, 87

[Wang et al., 2007] Wang, X., Kruger, D., and Wilke, A. (2007). Towards the development of an evolutionarily valid domain-specific risk-taking scale. *Evolutioniary Psychology*, 5(3):555–568. 62

[Widrow and Hoff, 1960] Widrow, B. and Hoff, M. (1960). Adaptive switching circuits. *IRE WESCON Convention Record*, 4:96–104. 23

[Wos et al., 1985] Wos, L., Pereira, F., Hong, R., Boyer, R. S., Moore, J. S., Bledsoe, W. W., Henschen, L. J., Buchanan, B. G., Wrightson, G., and Green, C. (1985). An overview of automated reasoning and related fields. *Journal of Automated Reasoning*, 1:5–48. 10.1007/BF00244288. 17

[Yamashita and Tani, 2008] Yamashita, Y. and Tani, J. (2008). Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment. *Computational Biology*, 4(11):e1000220. 1, 24, 25

List Of Own Publications

Journals papers

[Nassour et al., 2014] **Nassour, J.**, Hénaff, P., Ouezdou, F. B., and Cheng, G. (2014). Multi-Layer Multi-Pattern CPG for Adaptive Locomotion of Humanoid Robots. *Biological Cybernetics, [in press], 2014.*

[Nassour et al., 2013] **Nassour, J.**, Hugel, V., Ouezdou, F. B., and Cheng, G. (2013). Qualitative Adaptive Reward Learning with Success Failure Maps: Applied to Humanoid Robot Walking. In *IEEE Transactions on Neural Networks and Learning Systems.* Volume: 24, Issue: 1, Page(s): 81-93. January, 2013.

Conferences proceedings

[Nassour et al., 2012] **Nassour, J.** and Cheng, G. (2012). Cognitive Development Through a Neurologically-Based Learning Framework. Paper in *Workshop on Developmental Robotics, International Conference on Humanoid Robots*, Osaka, Japan, Nov. 29th, 2012.

[Nassour et al., 2012] **Nassour, J.** and Cheng, G. (2012). Biologically-inspired neural controller based on adaptive reward learning. In *Front. Comput. Neurosci. Conference Abstract: Bernstein Conference*, Munich, Germany, doi: 10.3389/conf.fncom.2012.55.00164.

[Nassour et al., 2012] **Nassour, J.**, Ouezdou, F. B., and Cheng, G. (2012). Exploiting Past Success Failure for Effective and Robust Task Learning. In *Proceedings of the 5th International Conference on Cognitive Systems (CogSys2012)*, Vienna, Austria.

[Nassour et al., 2011] **Nassour, J.**, Hénaff, P., Ouezdou, F. B., and Cheng, G. (2011). Experience-based Learning Mechanism with a Concept of Vigilance. In *Front. Comput. Neurosci. Conference Abstract: BC11 : Computational Neuroscience & Neurotechnology Bernstein Conference & Neurex Annual Meeting*, Freiburg, Germany, doi: 10.3389/conf.fncom.2011.53.00099.

[Nassour et al., 2011] **Nassour, J.**, Hénaff, P., Ouezdou, F. B., and Cheng, G. (2011). Bipedal Locomotion Control with Rhythmic Neural Circuits. In *International workshop on bio-inspired robots*, Nantes, France, 6-8 April.

[Nassour et al., 2010] **Nassour, J.**, Hénaff, P., Ouezdou, F. B., and Cheng, G. (2010). A study of adaptive locomotive behaviors of a biped robot: Patterns generation and classification. In *From Animals to Animats 11*, volume 6226 of *Lecture Notes in Computer Science*, pages 313–324. Springer Berlin / Heidelberg.

[Nassour et al., 2009] **Nassour, J.**, Hénaff, P., Ouezdou, F. B., and Cheng, G. (2009). The ieee/rsj international conference on intelligent robots and systems. st. louis, mo, usa. In *Experience-based learning mechanism for neural controller adaptation: Application to walking biped robots*, pages 2616–2621.