Enhancing Humanoid Learning Abilities: Scalable Learning through Task-Relevant Features

Takamitsu Matsubara

NAIST-IS-DD0561030

Doctoral Dissertation

Enhancing Humanoid Learning Abilities: Scalable Learning through Task-Relevant Features

Takamitsu Matsubara

November 1, 2007

Department of Information Systems
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of Engineering

Takamitsu Matsubara

Thesis Committee:

Professor Tsukasa Ogasawara	Supervisor
Professor Kenji Sugimoto	Co-supervisor
Professor Shin Ishii	Co-supervisor
Professor Mitsuo Kawato	Co-supervisor
Associate Professor Yukiyasu Kamitani	Co-supervisor
Doctor Jun Morimoto	JST-ICORP,ATR-CNS

Enhancing Humanoid Learning Abilities: Scalable Learning through Task-Relevant Features[1]

Takamitsu Matsubara

Abstract

In this dissertation, we propose a paradigm addressing motor learning on a humanoid robot via Reinforcement Learning (RL) in task-relevant feature space. In our paradigm, the high dimensional state variable is mapped approximately to a low dimensional feature space by considering low dimensionality in the movement of the humanoid robot (or human) during a specific task. Then, learning is efficiently achieved in this low dimensional space. In order to avoid the redundancy problem between such low dimensional spaces and the original space, *i.e.*, joint space, we utilize a dynamic system called a Central Pattern Generator (CPG) or human demonstration data. When we observe specific human movements, in each case independent variables can be much reduced compared to a human's structural degrees of freedom. This might be suitably explained by the coordination or synergy at the joint (sensor) level as proposed by Nicolai Bernstein, which suggests that more abstract features represent human movements rather than joint coordinate, and the dimension of the features could be extremely low. The effectiveness of the paradigm is validated through an application to learning biped walking and a class of whole-body dynamic movements.

Firstly, we present motor learning focusing on biped walking, as a first step, because it is not only one of most characteristic movements in humans and humanoids, but also a well-studied movement from the biological point of view in robotics. CPGs are introduced along with the previous work. One of the advantages of using this model is that appropriate coordination among all joints can be realised, which makes the robot dynamics simpler while entrainment property of the CPG synchronizes the robot to the environment.

[1] Doctoral Dissertation, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0561030, November 1, 2007

We apply RL in a low dimensional feature space, which is only composed of partial information about the robot, coordinated by CPG-arrangement to acquire a sub-optimal control policy for sensory feedback to CPGs.

Secondly, we focus on dynamic representative features in order to explore low-dimensional feature space towards the achievement of motor learning on humanoid robots in real environments. In particular, we present a novel learning approach, learning CoM movements rather than joint movements or end-effectors as typical cases. The CoM is of course one of the dynamically representative features for a class of dynamic movements, as indicated by traditional robotics. Moreover, this approach allows us to keep the Zero Moment Point (ZMP) in the support polygon, by accounting for the ZMP equation, which can prevent the humanoid robot from becoming an under-actuated system. This characteristic makes learning motor skills on humanoid robots simpler and more feasible, within the result of a CoM-Jacobian based weighted-pseudo inverse coordination.

Finally, we address the redundancy problem between the CoM and all joints by utilizing human demonstration data. The proposed redundancy resolution with the learning method described above is a suitable framework to deal with several movements.

Simulation and real hardware experimental results demonstrate the effectiveness of our proposed methods for efficiently achieving the desired motor learning task on a humanoid robot.

Keywords: Motor Learning, Humanoid Robot, Reinforcement Learning, Policy Gradient, Dimensionality Reduction

ヒューマノイドによる全身運動学習に関する研究：運動課題の特徴に基づく効率的学習の実現[2]

松原 崇充

内容梗概

本論文では，運動課題の特徴空間において強化学習を行なうことで，多自由度を持つヒューマノイドロボットによる全身運動学習を効率的に実現する方法を提案する．一般に，ヒトにおける歩行や走行等に代表される特定の全身運動を計測・解析すると，運動の持つ独立な自由度は，その構造的な自由度と比較して極めて低いことが知られている．これは，目的を達成するために，各関節が協働して運動しているからであると解釈できる．言い換えれば，一見複雑に見える全身動作であっても，その動作は低次元特徴空間において説明することができると考えられる．そこで本論文では，ヒューマノイド(ヒト)が持つ特定の運動中の低次元性に着目し，高次元の状態変数を近似的な低次元空間に射影して学習を行なうことで，効率的に全身協調動作の学習を実現する．また，低次元空間において表現される動作を高次元空間において実現する際に生じる冗長問題を，神経振動子により構成されるCentral Pattern Generator(CPG)や，ヒトの動作の観測データ等を用いて解決する．

はじめに，ヒューマノイドの特徴的な動作として，2足歩行運動について考える．ここでは，神経振動子により構成されるCPGを導入する．これにより，各関節への適切な協調性が導入されるだけでなく，環境との引き込み現象により，環境と同期した動作が得られることが期待される．本研究では，ロボット-CPG-環境間の同期特性に着目して，ロボットからの一部のセンサ情報のみに基づくCPGへの適切なセンサフィードバックを学習する方法を提案する．これにより，効率的に2足歩行動作の学習が実現されることを実験的に示す.

次に，ヒューマノイドロボットによる全身関節を用いた動作の学習を考える．ここでは，ヒューマノイドの力学的な特徴に着目して，動作の学習に適した低次元の特徴空間を構築する．具体的には，巨視的な力学的特徴変数の一つである重心の運動に着目する．重心運動を直接的に学習し，さらにこれを適切に全身運動に分配することで，結果的に，目標とする全身協調動作

[2] 奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 博士論文，NAIST-IS-DD0561030，2007年11月1日

を実現する方法を提案する．このような方法により，学習する独立変数が極端に減少するだけでなく，重心と Zero Moment Point (ZMP) 間の関係式を用いることで，学習中にもロボットのバランスや接地条件を力学的に陽に考慮できる．これにより，ベースリンクの持つ自由度を拘束でき，ロボットの力学的な自由度を減少させる．結果的に，各関節が協調した全身パンチ動作が効率的に学習されることを実験的に示す.

さらに，上述の動作学習の際に問題となる重心-全身関節間の冗長性を，ヒトの動作の観測データを基に縮約する方法について述べる．さらに，提案手法を用いて，スクワット動作の実現を行なう．シミュレーションにより，ヒトのスクワット動作より抽出された低次元特徴空間を用いて，ロボットの重心を操作することにより，結果的に，ロボットの全身関節運動がヒトのスクワット動作の特徴をある程度含んだものとなることを，実験的に示す.

最後に，本研究で得られた成果をまとめ，さらに今後の研究課題について述べる．

キーワード:運動学習，ヒューマノイドロボット，強化学習，方策勾配法，方策こう配法，次元縮約

List of publications

Journal Papers

1 **松原崇充**，森本淳，中西淳，佐藤雅昭，銅谷賢治，方策勾配法を用いた動的行動則の獲得：2 足歩行運動への適用，電子情報通信学会論文誌，vol. J88-D2, no. 1, pp. 53-65, 2005.
(**Takamitsu Matsubara**, Jun Morimoto, Jun Nakanishi, Masa-aki Sato, Kenji Doya, Learning a Dynamic Policy by Using Policy Gradient: Application to Biped Walking, Systems and Computers in Japan, Wiley InterScience, vol. 38, no. 4, pp. 25-38, 2007.)

2 **Takamitsu Matsubara**, Jun Morimoto, Jun Nakanishi, Masa-aki Sato, Kenji Doya, Learning CPG-based Biped Locomotion with a Policy Gradient Method, Robotics and Autonomous Systems, Elsevier Science, vol. 54, issue 11, pp. 911-920, 2006.

3 Gen Endo, Jun Morimoto, **Takamitsu Matsubara**, Jun Nakanishi, Gordon Cheng, Learning CPG-based Biped Locomotion with a Policy Gradient Method:Apply to a Humanoid Robot, The International Journal of Robotics Research (Accepted).

Conference Proceedings (reviewed)

1 **Takamitsu Matsubara**, Jun Morimoto, Jun Nakanishi, Masa-aki Sato, Kenji Doya, Learning Sensory Feedback to CPG with Policy Gradient for Biped Locomotion, IEEE International Conference on Robotics and Automation (ICRA), pp. 4175-4180, Barcelona, Spain, April 18-22, 2005.

2 Gen Endo, Jun Morimoto, **Takamitsu Matsubara**, Jun Nakanishi, Gordon Cheng, Learning CPG Sensory Feedback with Policy Gradient for Biped Locomotion for a Full-body Humanoid, The Twentieth National Conference on Artificial Intelligence (AAAI), pp. 1267-1273,

Pittsburgh, PA, USA, July 9-13, 2005.

3 **Takamitsu Matsubara**, Jun Morimoto, Jun Nakanishi, Masa-aki Sato, Kenji Doya, Learning CPG-based biped locomotion with a Policy Gradient Method, IEEE-RAS International Conference on Humanoids Robots Humanoids2005 (Humanoids), pp. 208-213, Tsukuba, Japan, December 5-7, 2005.

4 **Takamitsu Matsubara**, Jun Morimoto, Jun Nakanishi, Sang-Ho Hyon, Joshua G.Hale, Gordon Cheng, Learning to acquire whole-body humanoid CoM movements to achieve dynamic tasks, IEEE International Conference on Robotics and Automation (ICRA), pp. 2688-2693, Roma, Italy, April 10-14, 2007.

5 **Takamitsu Matsubara**, Jun Morimoto, Jun Nakanishi, Sang-Ho Hyon, Joshua G.Hale, Gordon Cheng, Learning to acquire whole-body humanoid CoM movements to achieve dynamic tasks with a policy gradient method, Advances in Neural Information Processing Systems (NIPS '06 Workshop), Vancouver, Canada, December 7-8, 2006.

Awards

1 **松原崇充**，IEEE 関西支部，学生研究奨励賞受賞，2006 年 2 月.

Patents

1 森本淳，**松原崇充**，佐藤雅昭，動的制御装置及び動的制御装置を用いた 2 足歩行移動体，特開 No.2005-199383.

2 森本淳，**松原崇充**，佐藤雅昭，中西淳，遠藤玄，2 足歩行移動装置，特開 No.2005-288594.

3 遠藤玄，Gordon Cheng，森本淳，**松原崇充**，中西淳，ロボット装置及びその制御方法，特開 No. 2006-289602.

Conference Proceedings

1 **松原崇充**，森本淳，逆運動学問題に置ける自然勾配法とヤコビアン擬似逆行列に基づく解法の等価性，日本ロボット学会学術講演会，3N21，千葉，2007 年 9 月.

2 **松原崇充**，丸山淳一，玄相昊，森本淳，人間動作より抽出される低次元特徴空間におけるヒューマノイドの全身運動制御，日本ロボット学会学術講演会，2H13，千葉，2007年9月.

3 丸山淳一，**松原崇充**，中西淳，森本淳，強化学習を用いたステッピングによる転倒回避動作の学習，日本機械学会ロボティクス・メカトロニクス講演会, 1A1-M09 (CD-ROM), 秋田，2007年5月.

4 **松原崇充**, 森本淳, 中西淳，佐藤雅昭，銅谷賢治，変分ベイズ法による自然方策勾配の推定法, 電子情報通信学会技術研究報告，NC2005-52, pp. 37-42，京都，2005年10月.

5 **松原崇充**，Learning to acquire whole-body humanoid CoM movements to achieve dynamic tasks with a policy gradient method, JST-ATR-CMU ジョイントワークショップ, 和歌山，2007年2月.

6 **Takamitsu Matsubara**, Jun Morimoto, Jun Nakanishi, Masa-aki Sato, Kenji Doya, Learning sensory feedback to CPG with a policy gradient method and its application to a real biped robot, 脳と心のメカニズム 第6回 夏のワークショップ，長野，2005年8月.

7 **松原崇充**，森本淳，中西淳，佐藤雅昭，銅谷賢治,2足歩行運動のための動的行動則の獲得, 日本機械学会ロボティクス・メカトロニクス講演会, 1A1-L1-51 (CD-ROM), 名古屋，2004年6月.

8 **松原崇充**, 森本淳, 中西淳，佐藤雅昭，銅谷賢治, 方策勾配法を用いた動的行動則の獲得：2足歩行への適用, 電子情報通信学会技術研告,NC2003-128, pp. 53-58，北海道，2004年1月.

Acknowledgement

まずはじめに，研究活動全般にわたって熱心にご指導くださったATR脳情報研究所，森本淳研究員，中西淳研究員に深く感謝したいと思います．今にして思えば，本学に入学して間もない頃にATR連携計算神経科学講座の見学会があることを知り，興味本意でふらっと訪れたことがすべての始まりでした．当時は，脳情報研究所のロボットが各マスメディアで大きく取り上げられていたこともあり，所属を希望する学生の多くが事前に関連論文や専門書を勉強してきていた中で，ATRはおろかロボットの運動学習について何一つ知らなかった僕は，ある意味でとても目立っていたのかもしれません．そんな僕の「頑張ります」という言葉だけで，情報科学ヒューマノイドロボット研究室に快く迎え入れ，興味深い研究テーマを与えて下さり，それから4年9ケ月，最後まで見捨てることなく熱心にご指導頂きました．基礎的な数学やプログラミングに始まり論理的な考え方から人生訓に至るまで，多くのことを教え，また研究の進行と僕の遅々たる成長を本当に根気強く見守って下さいました．森本さん，中西さんに，まずはじめに改めて深く感謝致します．本当にありがとうございました．

川人光男先生，神谷之康先生は，毎週の定例研究会などを通じて，常に研究をより良い方向へと導いて下さいました．お二人の豊富な知識と発想，それを実現する堅実な実行力には強い憧れを抱きました．また，川人先生には，ATR脳情報研究所所長として，常に僕たち学生に学ぶ機会や整った研究環境を与えて下さいました．ここに改めて深く感謝致します．銅谷賢治先生には，修士課程のときに直接ご指導を頂きました．純粋な好奇心から産み出される鋭い質問を通じて，逆に論理展開と考察力を勉強させて頂きました．ここに改めて感謝致します．

ATR脳情報研究所情報科学ヒューマノイドロボット研究室のGordon Cheng室長には，研究活動全般にわたって多大なるご指導・ご助力を頂きました．研究室や個別のミーティングを通じての研究指導，また僕の拙い英語表現も日々指導して下さいました．この学位論文を書き上げるにあたっても，辛抱強く表現や構成を直して下さいました．ここに改めて感謝致します．

情報科学ヒューマノイドロボット研究室の皆様には，日々の議論を通じて大変お世話になりました．玄相昊さんは，ヒューマノイドロボットのバランス制御に関する理論及び実験に必要な知識を根気強く授けて下さいました．ZMP制御によって，ロボットのバランス制御が実現できたときには本当に感動しました．Joshua G. Haleさんには，ロボットの実験環境の構築に多大なるご助力を頂きました．遠藤玄さんには，CPGを用いたヒューマノイドロボットの歩行制御技術に関して，多大なるご指導・ご助力を頂きました．こ

こに改めて感謝致します．計算脳イメージング研究室の佐藤雅昭室長，山下宙人さんには，ご専門のベイズ推定に関して，確率統計の基礎しか持ち得ない僕に懇切丁寧に教えて頂きました．ここに改めて深く感謝致します．認知神経科学研究室の田中宏和さんには，神経科学や物理学の豊富な知識に基づく有意義でかつ濃密な雑談を毎日のようにして頂きました．ここに改めて深く感謝致します．ありがとうございました．

ATR連携計算神経科学講座で一緒に頑張ってきた同級生や後輩たちにも大変お世話になりました．林卓治くんや森村哲朗くんとは，特に修士課程のときによく一緒に遊びました．柴田和久くんや藤原祐介くんとは，博士課程まで共に頑張りました．まさに三本の矢の如く，折られないように束になって耐え凌ぎました．南部功夫くんとは，サッカーをしたり，夜更けによく京都で牛丼を食べたりしました．有木由香さんとは，ロボット研究室の数少ない学生同士，共に支えあって頑張りました．また，夜な夜な無駄話に花を咲かせては，お互いを窮地に追い込みました．Mike Linくんは，休日には出無精な僕を外へと連れ出してくれました．Michael Mistryくんとは，研究の興味が近いこともあり，お互いの苦労を分かち合いました．藤田肇さんや坂東誉司さん，川脇大さんとは毎週火曜日にカレーを食べました．4年9ケ月間もの研究生活を楽しく過ごせたのは，良き友人たちに恵まれたからであると思います．ここに改めて深く感謝致します．ありがとうございました．

基幹講座として修士課程・博士課程で所属した応用システム科学講座及びロボティクス講座の皆様に，改めて深く感謝致します．特に，ロボティクス講座の金岡恵さんと近藤誠宏さんは，研究室に不在がちな僕をいつも気に掛けて下さいました．ここに改めて深く感謝致します．

お忙しい中，論文審査を引き受けて下さいました小笠原司教授，杉本謙ニ教授，石井信教授，川人光男教授，神谷之康准教授，森本淳博士(JST-ICORP, ATR-CNS)に深く感謝致します．杉本先生には，修士課程の基幹講座教授として，研究会や輪講を通じてご指導頂き，また，いつも温かい言葉で励まして下さいました．小笠原先生には，博士課程の基幹講座教授として，研究会等でのご指導，また，研究進捗や進路状況について，常に細やかなお気使いを下さりました．ここに改めて感謝致します．

最後に，博士課程への進学を認め，その生活を支えてくれた両親や家族と友人たちに深く感謝致します．本学位論文の提出に至るまで，本当に多くの方々に支えられてきました．ここに，改めて感謝を申し上げるとともに，今後もご指導くださいますよう厚くお願い申し上げます．

Chapter 1

Introduction

Humanoid robots with their similar physical structure to human is expected to help us with many tasks within our normal living environment, without the specific need of additional environmental customisation. With this in mind, recently, there has been a growing interest in the development of humanoid robots, and their control methods with the aim of achieving whole-body dynamic movements on these systems (Hirai, Hirose, Haikawa, & Takenaka, 1998; Kuroki, Ishida, Yamaguchi, Fujita, & Doi, 2001; Morimoto, Endo, Nakanishi, Hyon, Cheng, Bentivegna, & Atkeson, 2006; Hyon & Cheng, 2006). In particular, over the last decade, a number of methods for achieving various tasks on a humanoid robot have been explored, mainly to achieve biped walking as well as balancing (Nagasaka, 2000; Kagami, Kanehiro, Tamiya, Inaba, & Inoue, 2001; Sugihara & Nakamura, 2002; Kajita, Kanehiro, Kaneko, Fujiwara, Harada, Yokoi, & Hirukawa, 2003). Even though a number of real humanoid robots have demonstrated whole-body dynamic movements with these existing methods, it is still currently infeasible to introduce humanoid robots into our own living spaces, in order to help us in our daily lives. This is in large part due to their lack of ability to adapt to new environments as easily as humans and animals, *i.e.*, due to lack of the motor learning ability. In this dissertation, we propose a paradigm addressing this problem via Reinforcement Learning (RL) in task-dependent feature space. In our paradigm, the high dimensional state variable typically associated is approximately mapped to a low dimensional feature space by considering low dimensionality in the movement of the humanoid robot (or human) during a specific task. Then, learning is efficiently achieved in this low dimensional space. In order to avoid the redundancy problem between such low dimensional spaces and the original space, *i.e.*, joint space, we utilize a dynamical system called a Central Pattern Generator (CPG) or data from human demonstration. In Section 1.1, we briefly explain the problem we wish to address. We then describe the *key* essence of these problems, as addressed throughout this dissertation in Section 1.2. Section 1.3 presents the outline and contributions of each chapter.

1.1 Motor learning on humanoid robot via reinforcement learning

As one of the candidate solutions for granting the motor learning skill to humanoid robots, RL is a promising method compared with other learning frameworks because it requires no expert teacher or idealized desired behavior for skill-improvement. RL is a framework for improving the control rules of an agent, *i.e.*, a robot, through iterative interaction with the environment according to a trial-and-error paradigm without the use of an explicit model of the environment (Sutton & Barto, 1998; Doya, 2000). However, with an increase of dimensionality in state and action space, RL often requires not only a large number of iterations, but also requires large computational cost, especially in the case of learning of a complex control policy – *motor learning.* Although there have been many attempts to apply RL methods for several robots in simulation and real hardware systems for acquiring desired movements, most of the robots to which learning has been successfully applied so far have a small number of Degrees of Freedom (DoFs), and not as many as the 20 to 60 DoFs typically offered by humanoid robots (Kimura, Miyazaki, & Kobayashi, 1997; Kimura, Yamashita, & Kobayashi, 2001; Morimoto & Doya, 2001; Tedrake, Zhang, & Seung, 2004; Matsubara, Morimoto, Nakanishi, Sato, & Doya, 2005). In this dissertation, we focus on learning motor skills with full body humanoid robots via RL. In order to overcome *"the curse of dimensionality"*(Bellman, 2003; Sutton & Barto, 1998), we utilize task-dependent features extracted from biological knowledge, physics and statistical methods applied for human demonstration. This chapter presents an introduction with an outline of this dissertation.

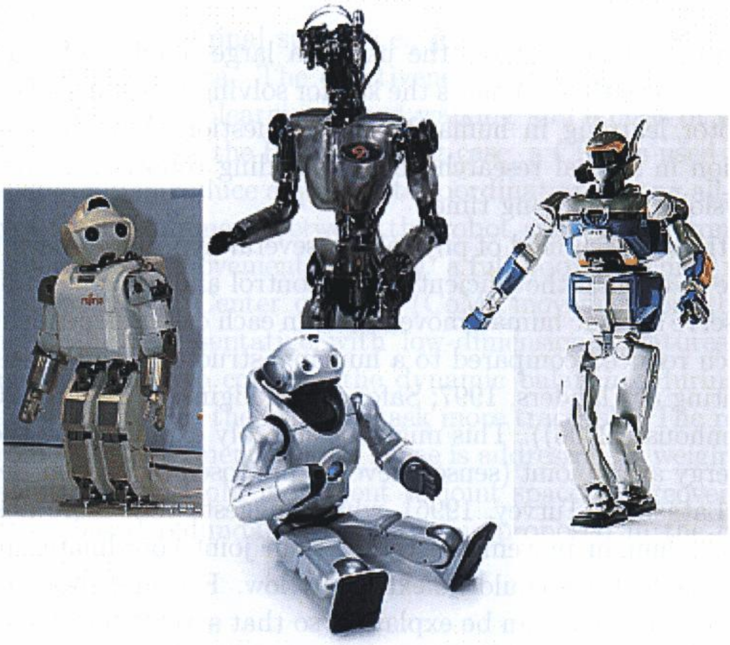


Figure 1.1: Typical humanoid robots

1.2 A key to *the curse of dimensionality*: Learning in task-relevant feature space

Figure 1.1 presents typical examples of humanoid robots. Roughly speaking, these humanoid robots have 20 to 60 joints, respectively. The dimension of the state in the equations of motion can be more than double of the number of joints, and the control input can be equal to the number of joints – if joints are all actuated. In such cases, RL often requires a large number of trials with the increase of dimensions in state and action space, which is well-known as "the curse of dimensionality" (Bellman, 2003). Thus, motor learning on a humanoid robot by RL with typical approaches can be said to be intractable.

Conversely, humans have much more complex bodily structure than that of humanoid robots. But yet, they can somehow achieve motor learning by trial and error in feasible manner, in both times and iterations, even if the target task requires the use of a large number of joints. How does human achieve this? What is the key for solving "the curse of dimensionality" for motor learning in human? These questions have been drawing much attention in several research fields, including robotics, neuroscience as well as physiology for a long time.

In the research field of physiology, several hypotheses have been proposed to-date to explain the efficient motor control and learning in humans. When we observe specific human movements, in each case independent variables can be much reduced compared to a human's structural DoFs (see, for example, (Soechting & Flanders, 1997; Safonova, Hodgins, & Pollard, 2004; Dejmal & Zacksenhouse, 2006)). This might be suitably explained by the coordination or synergy at the joint (sensor) level as proposed by Nicolai Bernstein (Bernstein, Latash, & Turvey, 1996), which suggests that more abstract features represent human movements rather than joint coordinate, and the dimension of the features could be extremely low. For adult humans, the efficient motor learning skill can be explained so that several coordination or synergy have been already explored and memorized in their brain through the large amount of experience, and appropriately utilized in order to achieve efficient learning even for a novel motor task. If we could extract such an abstract feature space suitable for the target movement, it may be possible to grant motor learning skill to humanoid robots like humans.

In a similar sense, dimensionality reduction techniques have been empirically applied for simplifying learning or optimisation problem in robotics and dynamic simulations (Chalodhorn, Grimes, Maganis, Rao, & Asada, 2006; Safonova et al., 2004). However, with simple behavior-based dimensionality reduction approaches, yet no general framework has been proposed. This dissertation addresses a functional-dimensionality reduction approach, which extracts task relevant features from the functional meaning of the target task, while also naturally introduces coordination or synergy among all-joints as a result.

Thus, in this dissertation, we propose a paradigm addressing motor learning on a humanoid robot via RL in task-relevant feature space. In our paradigm, the high dimensional state variable typically associated is approximately mapped to a low dimensional feature space by considering low dimensionality in the movement of the humanoid robot (or human) during a specific task. Then, learning is efficiently achieved in such a low dimensional space. In order to avoid the redundancy problem between such low dimensional spaces and the original space, *i.e.*, joint space, we utilize a CPG or human demonstration data. The effectiveness of the paradigm is validated through an application to learning biped walking and a class of whole-body dynamic movements. For the biped walking case, a CPG is used as the key, which does not only introduce appropriate coordination among all-joints, but also achieves the entrainment between the robot and the environment. For a whole-body dynamic movement case, *e.g.* a full-body punching movement, the control focusing on Center of Mass (CoM) movement can be the key. It's a dynamically representative with low-dimensional features, and also it brings us to be able to consider the dynamic balancing during learning. This characteristic makes the learning task more tractable. The redundancy problem from CoM movement to joint space is addressed by weighted pseudo inverse resolution from CoM movement to joint space. Moreover, a human demonstration based redundancy resolution is proposed in the consecutive chapter.

1.3 Contributions and organization of chapters

In this dissertation, we explore how humanoid robots can achieve motor learning while considering its applicability in real environment. The main contribution of this dissertation is to show the importance of the learning in task-relevant feature space, in order to achieve motor learning on humanoid robots. Moreover, novel learning approaches for achieving biped walking and some classes of whole-body dynamic skills, which require cooperation in all-joints and specific momentum or CoM movements for well-performance, on humanoid robots are applied in simulations and real hardware systems.

The organization of the following chapters is as follows: In Chapter 2, we first describe the RL methods at the viewpoint of suitability for motor learning on humanoid robots with their mathematical definitions. Motor learning requires in a RL method to deal with continuous state and action space as well as, their efficiency in time and calculation costs. We also look at several methods by considering these requirements. In Chapter 3, we present motor learning focusing on biped walking, as a first step, because it is not only one of most characteristic movements in humans and humanoids, but also a well-studied movement from the biological point of view in robotics. The

CPG is introduced along with previous work (Taga, Yamaguchi, & Shimizu, 1991; Sato, Nakamura, & Ishii, 2002; Fukuoka, Kimura, & Cohen, 2003; Endo, Morimoto, Nakanishi, & Cheng, 2004). The advantages of using this model are that appropriate coordination among all joints can be realised, which makes the robot dynamics simpler while entrainment property of the CPG synchronizes the robot to the environment. We apply RL in a low dimensional feature space, which is only composed of partial information about the robot, coordinated by CPG-arrangement to acquire a sub-optimal control policy for sensory feedback to CPGs.

Despite of our success for learning CPG-based biped waking, it is still open problem, as to how to introduce an appropriate joint coordination for other movements, such as appropriate arrangement of CPGs in biped walking case. One's hope is to find a low dimensional feature space, which can be commonly effective in motor learning for all-daily movements. However, several results (d'Avella, Saltiel, & Bizzi, 2003; Tresch, Cheung, & d'Avella, 2006; Safonova et al., 2004) suggest that basis of synergy is typically task dependent even though there are some common ones. Second hope may be to find a general principle to automatically extract task-dependent and low-dimensional features, however, it also seems unrealistic due to the variety of the movements. Instead of the search for a universal approach, we move to considering a general framework for a limited class of movements. In Chapter 4, we focus on a class of whole-body motor skills, in which all-joints are cooperative and momentum or CoM movements are correlated with its performance, because it can be expected for such tasks that the learning is achieved in extremely low-dimensional feature space compared with original dimension of the humanoid robots. In particular, we present a novel learning approach, learning CoM movements rather than joint movements or end-effectors as typical cases. The CoM is of course one of the dynamically representative features as pointed out by traditional robotics. Moreover, this approach allows us to keep the Zero Moment Point (ZMP) in the support polygon, by accounting for the ZMP equation, which can avoid the humanoid robot to become an under-actuated system. This characteristic makes learning motor skills on humanoid robots simpler and more feasible, within the result of a CoM-Jacobian based weighted-pseudo inverse coordination.

The CoM learning approach can be effective for achieving dynamic target tasks. However, the kinematic configuration in the movement is determined by the weighted matrix used in pseudo inverse calculation and it is still open problem on how to choose the weighted matrix appropriately. The difficulty mainly comes from the redundancy of joint space to CoM space. Chapter 5 addresses the redundancy problem between the CoM and all joints by utilizing human demonstration data. The solution of this problem with the learning method presented in Chapter 4 is a suitable framework to deal with several movements. In Chapter 6, we conclude this doctoral dissertation, and propose possible future extensions of our study.

In summary, we outline the contributions with their associated chapters:

Chapter 2: describes reinforcement learning from the viewpoint of motor learning in humanoid robotics, and identifying the related problems.

Chapter 3: demonstrates the effectiveness of dimensionality reduction achieved by CPG-arrangement through learning a biped walking behavior.

Chapter 4: demonstrates the scalability of the learning to full-body humanoid robots in task relevant feature space, a case study of learning of a dynamic punching task utilizing the CoM is presented in this chapter.

Chapter 5: describes a novel redundancy resolution approach utilizing human demonstration used for CoM control.

Chapter 6: concludes this doctoral dissertation.

Chapter 2

Reinforcement Learning for Motor Learning on Humanoid Robot

RL can be a useful tool for achieving motor learning on robots because it requires no expert teacher or idealized desired behavior for skill-improvement. However, for humanoid robots, applying RL in a straightforward manner can easily suffer from "the curse of dimensionality" due to the large number of DoFs associated. All RL methods require the integration over state or state-action space as well as state or state-action value estimation. The cost for the process can be exponentially high with the increase of dimensionality in the state and action space. Moreover, motor learning tasks on robots, in general, require the RL algorithm which is able to deal with continuous state and action space in achieving smooth control. In this chapter, we introduce various RL methods in relation to their suitability for motor learning tasks. In general, RL methods can be categorized as *value-function* based RL and *policy gradient* based RL. In conjunction with consecutive chapters, we point out the challenges in RL for motor learning tasks on humanoid robots. In Section 2.1, mathematical definitions of RL are described. Sections 2.2 and 2.3 presents the value-function based RL and the policy gradient based RL with several algorithms, respectively. Applications of policy gradient methods to a simple toy problem are discussed in Section 2.4 to emphasize the advantages and disadvantages of these methods. Finally, in Section 2.5, we explain the difficulties of achieving motor learning via RL on humanoid robots by means of presenting the dynamics of a typical humanoid robot.

2.1 System model definitions in RL

In this section, we describe the RL framework. Here, we focus on the Markov Decision Problem (MDP) as depicted in Fig.2.1, which can be modeled by a system consisting of a state space S, an action space U and which will be considered infinite. State transitions are governed by a state transition probability $p(s'|s,a)$, where s, $s' \in S$ and $a \in U$. The stochastic policy which controls the MDP is defined by $\pi(s,a;\theta)$ $(= p(a|s))$ and the reward function is defined as $r(s,a)$. The objective of the RL is to acquire the optimal policy $\pi^*(s,a)$ in the sense of maximizing an accumulated reward. The paradigm to achieve the objective with the explicit model of both $p(s'|s,a)$ and $r(s,a)$ is well-known as the Dynamic Programming (DP) (Bersekas, 2000, 2001).

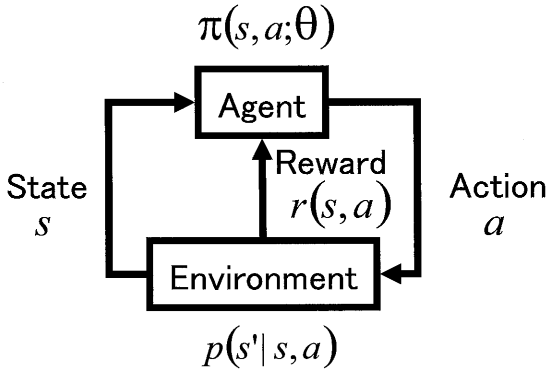


Figure 2.1: Markov Decision Process(MDP)

2.2 Value-function based RL method

First, we define the value of the state s and value of state-action pair (s, a) as follows:

$$V^{\pi}(s) = \mathbf{E}_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right\}, \tag{2.1}$$

$$Q^{\pi}(s, a) = \mathbf{E}_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right\}, \tag{2.2}$$

$$\tag{2.3}$$

where, $V(s)$ is the state value function, and $Q(s, a)$ is the state-action value function respectively. $E_{\pi}\{\cdot\}$ means the expectation over state and action space.

From the above definitions, we can derive the following Bellman-equations, which should be satisfied in each state and action.

$$V^{\pi}(s) = \int_a \pi(s, a) \int_{s'} p(s'|s, a) \left\{r(s', a) + \gamma V^{\pi}(s')\right\} dads', \tag{2.4}$$

$$Q^{\pi}(s, a) = \int_{s'} p(s'|s, a) \left\{r(s', a) + \gamma V^{\pi}(s')\right\} ds'. \tag{2.5}$$

One of objectives for RL is to find an optimal policy π^* which satisfies the following conditions:

$$V^*(s) = \max_{\pi} V^{\pi}(s), \tag{2.6}$$

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a), \tag{2.7}$$

where, $V^*(s)$ and $Q^*(s, a)$ are optimal value functions, respectively. For the optimal policy, the Bellman's optimality equation is defined as

$$V^*(s) = \max_{a} \int_a \pi(s, a) \int_{s'} p(s'|s, a) \left\{r(s', a) + \gamma V^*(s')\right\} dads', \tag{2.8}$$

$$Q^*(s, a) = \int_{s'} p(s'|s, a) \left\{r(s', a) + \gamma \max_{a'} Q^*(s', a')\right\} ds'. \tag{2.9}$$

$$\tag{2.10}$$

In other words, the optimal policy is defined as follows:

$$\pi^* = \arg\max_a Q^*(s,a) = \arg\max_a \left\{ \int_{s'} p(s'|s,a) \left\{ r(s',a) + \gamma V^*(s') \right\} \right\}. \tag{2.11}$$

To find an optimal policy π^*, which maximizes value function at all states and implicitly assumes "greedy-policy", is called a value-function based RL. Q-learning or Sarsa-learning are very famous algorithms for achieving the learning and have been widely used for variety of applications (Sutton & Barto, 1998). Both has a convergence proof in discrete state and action case with appropriate conditions (Watkins & Dayan, 1992; Sutton & Barto, 1998).

In the above, the value functions are defined in discrete time. It would be more preferable to consider continuous time definitions for robotic applications such as our purpose. Fortunately, RL in continuous time has been proposed in (Doya, 2000; Morimoto & Doya, 2001). For the continuous-time deterministic system,

$$\dot{s}(t) = f(s(t), a(t)) \tag{2.12}$$

where, $s \in S$ is the state and $a \in U$ is the control input. We denote the immediate reward for the state and action as

$$r(t) = r(s(t), a(t)). \tag{2.13}$$

For the above continuous-time system, the continuous-time value function is defined as

$$V^\pi(s(t)) = \int_t^\infty e^{-\frac{h-t}{\tau}} r(s(h), a(h)) dh, \tag{2.14}$$

where, τ is the time constant for discounting future rewards. As same as the discrete-time cases, the optimal policy μ^* should satisfy the following condition:

$$V^*(s(t)) = \max_a \left[\int_t^\infty e^{-\frac{h-t}{\tau}} r(s(h), a(h)) dh \right], \tag{2.15}$$

where, V^* is the optimal value function. According to the principle of optimality, the condition for the optimal value function at time t is given by

$$\frac{1}{\tau}V^*(s(t)) = \max_{a(t)\in U}\left[r(s(t),a(t)) + \frac{\partial V^*(s(t))}{\partial s}f(s(t),a(t))\right], \qquad (2.16)$$

which is a discounted version of the Hamilton-Jacobi-Bellman (HJB) equation (Doya, 2000). Thus, the optimal policy μ^* is represented

$$\mu^*(s(t)) = \arg\max_{a\in U}\left[r(s(t),a) + \frac{\partial V^*(s)}{\partial s}f(s(t),a)\right]. \qquad (2.17)$$

Based on these definitions, the continuous time counter parts of value function based RL algorithms have been proposed (Doya, 2000).

Although there have been successfully applied both continuous and discontinuous time based methods to several applications including robotics, however, it can be problematic for continuous state and action system with several reasons. The motor learning obviously requires continuous state and action settings because discontinuity in motor command is not desirable for robotic hardware in general. In order to manage such a request, a function approximator is often used to approximate value functions over continuous state and action space. Sutton *et al.* proposed TD(0) and TD(λ) and prove their convergence with a class of linear function approximator (Sutton, 1988). Tsitsiklis *et al.* presented a proof of convergence and a bound on the resulting approximation error with a linear function approximator and also presented the possibility of divergence when temporal difference learning is used in the presence of a nonlinear function approximator (Tsitsiklis & Roy, 1996). Furthermore, the performance of the greedy policy derived from the approximated value function does not guarantee to improve on each policy improvement iteration, and in fact can be worse than the old policy due to the approximation error over all states (Baxter & Bartlett, 2001b). These results suggest that to apply this kind of value function based RL with a greedy policy to motor learning tasks, especially for dealing with a real robot within a real environment can be dangerous. In next subsection, we briefly describe another approach for RL, *the policy gradient methods*, which can be more appropriate for our purpose.

2.3 Policy gradient based RL method

As discussed in the previous subsection, the value-function based RL approach with function approximators has several limitations. An alternative approach in RL with function approximators is the policy gradient approach. This approach has a stochastic policy explicitly expressed by a function with its own parameters called *policy parameters*, rather than greedy policy implicitly assumed in the value function based approaches. Then, the policy parameters are updated in the direction towards increasing a performance criterion $\eta(\theta)$ (*e.g.* the average reward $\eta(\theta) = \lim_{n\to\infty} \frac{1}{n} E\{r_1 + r_2 + \cdots + r_n | \pi\} = \int_s d^\pi(s) \int_u \pi(s,a) r(s,a) dads$) such that $\theta \leftarrow \theta + \alpha \frac{\partial \eta(\theta)}{\partial \theta}$, where $d^\pi(s)$ is the stationary distribution of the state s, the parameter α determines step size of the gradient decent. REINFORCE (Williams, 1992) is one early example of the policy gradient type of RL methods.

According to the policy gradient theorem presented in (Sutton, McAllester, Singh, & Mansour, 2000), the gradient of the policy with respect to policy parameters can be obtained by the following equations:

$$\frac{\partial \eta}{\partial \theta} = \int_s d^\pi(s) \int_a \pi(s,a) \frac{\partial \log \pi(s,a)}{\partial \theta} \left[Q^\pi(s,a) - b(s) \right] dads. \tag{2.18}$$

Moreover, even if we use a function approximator $f_w(s,a)$ for the action-value function $Q(s,a)$, which satisfies that $\int_s d^\pi(s) \int_a \pi(a,s)[Q^\pi(s,a) - f_w(s,a)] \frac{\partial f_w(s,a)}{\partial w} dsda = 0$, and $\frac{\partial f_w(s,a)}{\partial w} = \frac{\partial \pi(s,a)}{\partial \theta} \frac{1}{\pi(s,a)}$, the following theorem is also derived:

$$\frac{\partial \eta}{\partial \theta} = \int_s d^\pi(s) \int_a \pi(s,a) \frac{\partial \log \pi(s,a)}{\partial \theta} f_w^\pi(s,a) dads, \tag{2.19}$$

which is easily derived from the fact that the gradient of the policy parameterization is orthogonal to the approximation error (Sutton et al., 2000). Based on these theorems, the convergence proof is also given. As a slightly different approach to obtain the gradient, REINFORCE algorithm-type methods have been proposed. The method approximately uses the actual returns $R_t = \sum_{k=1}^\infty \gamma^k r_{t+k}$ instead of each $Q(s_t, a_t)$ (Williams, 1992; Baxter & Bartlett, 2001b; Kimura & Kobayashi, 1998). As pointed out in (Sutton et al., 2000), the compatible function $f_w(s,a)$ satisfies that $\int_a \pi(s,a) f_w(s,a) da =$

$w^T \frac{\partial}{\partial \theta} \int_a \pi(s,a) = 0$, where $f_w(s,a) = w^T \frac{\partial \pi(s,a)}{\partial \theta} \frac{1}{\pi(s,a)}$. It would be more natural to approximate the advantage function $A(s,a) = Q(s,a) - V(s)$ rather than $Q(s,a)$ by compatible function $f_w(s,a)$ because $\int_a A(s,a)da = \int_a \{Q(s,a) - V(s)\}\, da = V(s) - V(s) = 0$. In this case, the reward baseline $b(s)$ can be interpreted as $V(s)$ and it may reduce the variance of the estimated gradient (Sutton et al., 2000; Kimura & Kobayashi, 1998), while it has been pointed out that $V(s)$ is not the optimal reward baseline for variance reduction (Greensmith, Bartlett, & Baxter, 2004).

Although this method searches a local minimum rather than global minimum like value-function based approach, however, it has several advantages. One of the advantages of this approach is that it can be applicable for dealing with continuous state and action case with function approximators because the approximation error for value functions does not result in policy degradation. This characteristic is really suitable for motor learning on humanoid robots. Further interesting topics around the policy gradient method to motor learning are the applicability for Partially Observable Markov Decision Processes(POMDPs), the variance reduction techniques for gradient estimation, and natural policy gradient approaches. We briefly explain these topics in following paragraphs.

Policy Gradient for POMDPs

For most of the applications in robotics, sensors have limited performance, *i.e.*, the signal can always be deteriorated by noise. Such a situation can be modelled as a POMDP, in which we have only access to the observation y, and not to the state s. This is a well studied problem. (For a recent review, see (Aberdeen, 2003) and (Murphy, 2000)). Although value function based RL is not applicable in such situations, the policy gradient method is still applicable by considering the following equations:

$$\pi(s,a) = p(a|s) = \int p(a|y)p(y|s)dy, \tag{2.20}$$

$$\frac{\partial \pi(s,a)}{\partial \theta} = \int \frac{\partial p(a|y)}{\partial \theta} p(y|s)dy \tag{2.21}$$

$$\tag{2.22}$$

and by substituting the above with Eq.(2.18), we can calculate the gradient even for POMDPs (*e.g.* in (Baxter & Bartlett, 2001b)). This characteristic permits the learning to apply to real environment. This approach for POMDP is often referred as a memory-less resolution for POMDPs. The simple extension to the memory-based resolution was studied by (Aberdeen, 2003).

Variance Reduction Techniques

The gradient for average reward with respect to policy parameters is given as,

$$\frac{\partial \eta}{\partial \theta} = \int_s d^\pi(s) \int_a \pi(s,a) \frac{\partial \log \pi(s,a)}{\partial \theta} \left[Q^\pi(s,a) - b(s) \right] dads. \tag{2.23}$$

One of the most simple approximations can be achieved via the Monte Carlo method using samples from the policy and the environment. Under certain conditions, the approximation is simply given by $\frac{\hat{\partial \eta}}{\partial \theta} \approx \frac{1}{T} \sum_{i=1}^{T} \frac{\partial \log \pi(s_i,a_i)}{\partial \theta} [Q^\pi(s_i, a_i) - b(s_i)]$ According to the low of large numbers, $\mathbf{E}(\frac{\hat{\partial \eta}}{\partial \theta}) = \frac{\partial \eta}{\partial \theta}$ and $Var(\frac{\hat{\partial \eta}}{\partial \theta}) = Var\left\{ \frac{\partial \log \pi(s_i,a_i)}{\partial \theta} [Q^\pi(s_i, a_i) - b(s_i)] \right\} /T$ means that $\lim_{T \to \infty} \frac{\hat{\partial \eta}}{\partial \theta} = \frac{\partial \eta}{\partial \theta}$.

However, in practice, it is impossible to take infinite number of samples for a reliable gradient estimation, not even with numerical simulations. Therefore, we have to admit some variance in the estimated gradient. Variance reduction techniques with baseline $b(s)$ as a control variable are proposed in order to address this problem (Williams, 1992; Kimura & Kobayashi, 1998; Weaver & Tao, 2001; Greensmith et al., 2004). The recent theoretical analysis provided showed an optimal value of $b(s)$ for a state-dependent case (Greensmith et al., 2004), given as follows:

$$b^*(s) = \frac{\mathbf{E}_a \left[\left(\frac{\partial \ln \pi(s,a)}{\partial \theta} \right)^2 Q(s,a) \right]}{\mathbf{E}_a \left[\left(\frac{\partial \ln \pi(s,a)}{\partial \theta} \right)^2 \right]}. \tag{2.24}$$

For implementation, the gradient for the current baseline to approach the optimal baseline can be obtained by an algorithm based on the Monte Carlo approximation (Greensmith et al., 2004). The use of an optimal baseline can reduce the variance of the estimated gradient, on the other hand, the estimation of the optimal baseline may also suffer from some variance in the estimated value.

Natural Gradient Approach

Although the policy gradient method has numerous advantages as described above, nevertheless, it sometimes demonstrates non-monotonic performance as described by (Kakade, 2002). It can be interpreted that it is a non-covariant gradient decent, *i.e.*, a different parameterization of the policy may lead to a different gradient direction (Kakade, 2002; Bagnell & Schneider, 2003; Peters, Vijayakumar, & Schaal, 2003). To solve this problem, we first define the metric by using the Kullback Leibler divergence, in order to measure the effect of changes in policy parameters to state and action probabilistic distributions, as given by,

$$\begin{aligned}
D(\theta|\theta+d\theta) &= \int_s\int_a p(s,a;\theta)\log\frac{p(s,a;\theta)}{p(s,a;\theta+d\theta)}dads \\
&\approx \int_s\int_a p(s,a;\theta)\left\{\frac{\partial\log p(s,a;\theta)}{\partial\theta}d\theta - d\theta^T\frac{\partial^2\log p(s,a;\theta)}{\partial^2\theta}d\theta\right\}dsda \\
&= d\theta^T\left\{\int_s\int_a p(s,a;\theta)\left\{\frac{\partial\log\{d^\pi(s)\pi(s,a;\theta)\}}{\partial\theta}\frac{\partial\log\{d^\pi(s)\pi(s,a;\theta)\}^T}{\partial\theta}\right\}dsda\right\}d\theta \\
&= d\theta^T\left\{\int_s\int_a p(s,a;\theta)\left\{\frac{\partial\{\log d^\pi(s)+\log\pi(s,a;\theta)\}}{\partial\theta}\frac{\partial\{\log d^\pi(s)+\log\pi(s,a;\theta)\}^T}{\partial\theta}\right\}dsda\right\}d\theta \\
&\approx d\theta^T\left\{\int_s\int_a p(s,a;\theta)\left\{\frac{\partial\log\pi(s,a;\theta)}{\partial\theta}\frac{\partial\log\pi(s,a;\theta)^T}{\partial\theta}\right\}dsda\right\}d\theta \\
&= d\theta^T I(\theta)d\theta. \qquad (2.25)
\end{aligned}$$

We then consider the gradient direction with constraint $D(\theta|\theta+d\theta)=\varepsilon$. The policy gradient based on the above metric is referred to as a natural policy gradient, which was originally discovered by (Kakade, 2002) and further studied in (Bagnell & Schneider, 2003; Peters et al., 2003). However, in (Kakade, 2002), the Fisher information matrix, $F_s(\theta) = E_\pi\left[\frac{\partial\log\pi}{\partial\theta}\frac{\partial\log\pi}{\partial\theta}\right]$ of the policy $\pi(s,a;\theta)$ was firstly defined, with the intuitive idea such that the average reward is technically a function of the distribution. The metric is then defined by calculating expectation of the matrix with a steady state distribution $d^{\pi}(s)$ as $F(\theta) = E_{d^{\pi}(s)}\left[F_s(\theta)\right] = I(\theta)$. Further studies (Bagnell & Schneider, 2003; Peters et al., 2003) pointed out that Kakade's work was not based on a proper probability manifold to derive the metric $F(\theta)$. They further explained that the metric is exactly based on the distribution over paths (or trajectory denoted by $\tau = [s_{0:H}, a_{0:H}]$) as $p(\tau;\theta)$ (Bagnell & Schneider, 2003; Peters et al., 2003). As we presented above, we suggest a way to define the meaning of the metric in which the metric is simply based on the steady-state and policy distributions, which may be more intuitive and can be easily understood.

According to Amari *et al.*(Amari, 1998), the natural gradient can be obtained with the metric as given by, $\frac{\hat{\partial \eta}}{\partial \theta} \propto I(\theta)^{-1}\frac{\partial \eta}{\partial \theta}$. The advantage is that the gradient does not affect the parameterization of the policy further, which often makes the learning process much faster with high computing cost. Several algorithms have been explored in (Peters et al., 2003; Bagnell & Schneider, 2003), in order to efficiently achieve the natural gradient estimation by taking an affinity with the *compatible function* based policy gradient (Sutton et al., 2000; Konda & Tsitsiklis, 2003).

2.4 An example: 3-State MDP

Prior to our consideration of the motor learning via RL on complex robotic system, let us first examine a simple toy example, 3-state MDP depicted in Fig.2.2, as proposed by (Baxter & Bartlett, 2001a). The objective is simply to acquire a policy which maximizes the average reward. For this problem, we are able to obtain an exact policy gradient by a simple matrix calculation, as we have the model of the environment, which consists of a few discrete states and actions (Baxter & Bartlett, 2001a). In our case, we further the result of (Baxter & Bartlett, 2001a), we calculate the natural policy gradient for the problem analytically, and compare both methods in an ideal situation, thus, we can calculate both gradients without approximating any models or values from samples.

Assuming that the reward depends only on the state $s \in \{A, B, C\}$ rather than action $a \in \{a_1, a_2\}$, the average reward can be written in the form,

$$\eta(\theta) = \sum_s d^\pi(s;\theta) r(s) = \mathbf{d}^T \mathbf{r} \tag{2.26}$$

where, $\mathbf{d}^T = [d^\pi(A), d^\pi(B), d^\pi(C)]$, $\mathbf{r}^T = [r(A), r(B), r(C)] = [0, 0, 1]$. The policy $\pi(s, a)$ is defined as,

$$\pi(s, a_1) = \frac{e^{\mu_1(s)}}{e^{\mu_1(s)} + e^{\mu_2(s)}}, \pi(s, a_2) = \frac{e^{\mu_2(s)}}{e^{\mu_1(s)} + e^{\mu_2(s)}}, \tag{2.27}$$

where, $\mu_1(s) = \theta_1\phi_1(s) + \theta_2\phi_2(s), \mu_2(s) = \theta_3\phi_3(s) + \theta_4\phi_4(s)$, and $\phi_1(A) = \frac{12}{18}, \phi_1(B) = \frac{6}{18}, \phi_1(C) = \frac{5}{18}, \phi_2(A) = \frac{6}{18}, \phi_2(B) = \frac{12}{18}, \phi_2(C) = \frac{5}{18}$. Each $d^\pi(s)$ has to satisfy *the balanced equation* to be the stationary distribution of the state s as

$$d^\pi(s') = \sum_s \sum_a p(s'|s, a)\pi(s, a) d^\pi(s), \tag{2.28}$$

where,

$$\mathbf{P} = \begin{bmatrix} \sum_a p(A|A, a)\pi(A, a) & \cdots & \sum_a p(C|A, a)\pi(A, a) \\ \vdots & \ddots & \vdots \\ \sum_a p(A|C, a)\pi(C, a) & \cdots & \sum_a p(C|C, a)\pi(C, a) \end{bmatrix}. \tag{2.29}$$

With the above settings, the policy gradient for the average reward criterion presented in Eq.(2.18) is given as follows (Baxter & Bartlett, 2001b):

$$\frac{\partial \eta}{\partial \theta} = \mathbf{d}^T \frac{\partial \mathbf{P}}{\partial \theta} \left(\mathbf{I} - \mathbf{P} + e\mathbf{d}^T \right)^{-1} \mathbf{r}. \tag{2.30}$$

$$\tag{2.31}$$

Furthermore, the natural gradient can be obtained by,

$$\frac{\hat{\partial \eta}}{\partial \theta} = \mathbf{I}(\theta)^{-1} \frac{\partial \eta}{\partial \theta}, \tag{2.32}$$

where,

$$[\mathbf{I}(\theta)]_{ij} = \sum_s \sum_a d^\pi(s) \pi(s,a) \left[\frac{\partial \log \pi(s,a)}{\partial \theta_i} \frac{\partial \log \pi(s,a)}{\partial \theta_j} \right]. \tag{2.33}$$

The result for both ordinal and natural policy gradient method for the 3-state MDP is presented in Fig.2.3. As presented above, in an ideal situation, we can easily obtain the (sub-)optimal policy. However, for problems we do not have any prior knowledge of the environment, the calculation of the policy is replaced from trivial matrix calculations to an estimation problem, which is not trivial anymore, especially for high dimensional cases.

2.5 Motor learning on humanoid robot via reinforcement learning

In this section, we would like to consider applications of RL for motor learning on humanoid robots. First, we start by looking at the general form representing the dynamics of the robot like typical manipulators. Assuming that the robot having the rigid bodies and the base link is exactly located in the inertial frame, we can derive the following equations of motion using Lagrangian formalism:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \tau, \tag{2.34}$$

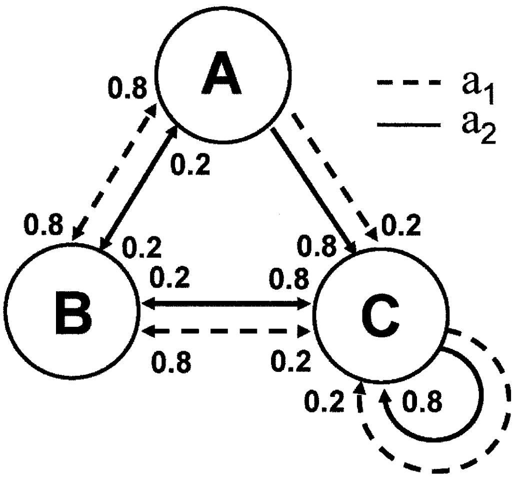


Figure 2.2: 3-state Markov Decision Process

where, $\mathbf{M}$ is the inertia matrix, $\mathbf{h}$ is the centrifugal, gyroscopic and Coriolis effects, and $\mathbf{g}$ is the generalized gravity force vector. In the case of the humanoid robot, however, the base link is not fixed on the inertial frame and it is represented by its position and orientation from inertial frame. In such a floating base dynamics of the humanoid robot can be represented in following equations:

$$\mathbf{M}'(\mathbf{q},\phi)\begin{bmatrix}\ddot{\mathbf{q}}\\ \ddot{\phi}\end{bmatrix}+\mathbf{h}'(\mathbf{q},\dot{\mathbf{q}},\phi,\dot{\phi})+\mathbf{g}'(\mathbf{q},\phi)=\begin{bmatrix}\tau\\ 0\end{bmatrix}+\mathbf{J}_s^T F_r, \tag{2.35}$$

where, ϕ is the state of passive joints which consists of linear DoFs and rotational DoFs. $\mathbf{J}_s$ is contact point Jacobian, and F_r is contact force. See more details in (Sentis & Khatib, 2005).

Most straightforwardly, for applying RL to achieve the desired motor skill, we probably set state $s^T = [\mathbf{q}, \hat{\mathbf{q}}, \phi, \hat{\phi}]$ and action $a^T = [\tau]$. As pointed out in previous sections, even though the policy gradient learning methods can deal with continuous state and action settings by using function approximators, however, the calculation of the gradient requires the integration over state and action space, which is computationally extremely high with increase of the number of dimension in state and action space. Recent studies in this research field report some complex robotic applications of RL such

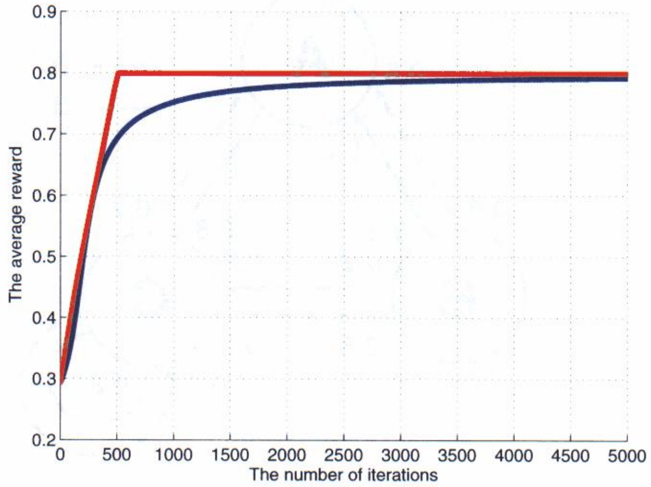


Figure 2.3: The average rewards by analytical PG and NPG. The step size parameter α is tuned so that the initial slope of the average reward is close. $\alpha = 0.1$ and $\alpha = 0.003$ were used for PG and NPG, respectively. NPG does not depend on the current policy parameters in performance, however, PG does.

as 3-link standing-up robot(Morimoto & Doya, 2001), 4-legged locomotion robot(Kimura & Kobayashi, 1998) and simple-structured passive dynamic walker(Tedrake et al., 2004). Even though, these robots have relatively complex bodies than typical manipulators, however, it is still small number of DoFs compared with typical humanoid robots. No studies have demonstrated the achievement of the motor learning via RL on robots with large number of DoFs such as humanoid robots.

2.6 Conclusion and discussion

In this chapter, we presented the RL methods while considering their applicability to motor learning for humanoid robots. As one of the candidate methods for our purpose, we presented a policy gradient method with an outlined of the difficulty involved, in particular in high dimensional problems, such as applying it to motor learning on humanoid robots.

Figure 2.4 presents the target robots in this dissertation. Left is the plan-

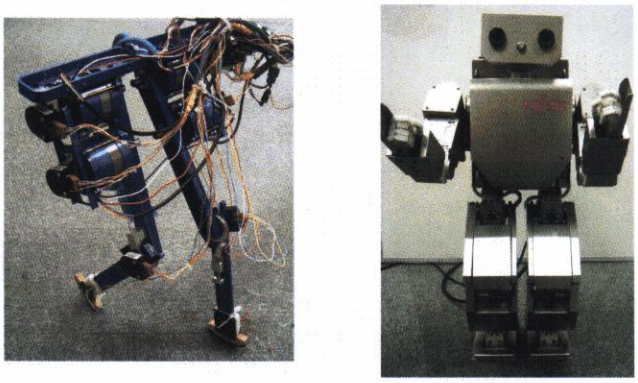


Figure 2.4: Target robots in this dissertation.Left if 5-link biped robot(presented in Chapter 3). Right is a full-body humanoid robot(presented in Chapter 4 and 5).

ner 5 link-biped robot (named DB-chan), and right is a full-body humanoid robot (HOAP2, developed by Fujitsu). Although DB-chan is simpler than full-body humanoid robots, however, it can still provide sufficient complexity for our initial studies.

In our first study, in the next chapter, we consider the application of RL to biped walking on the robot (DB-chan) via the proposed paradigm, which is learning in task-relevant feature space.

Chapter 3

Learning Relevant Sensory-Feedback to Biped Walking

In this chapter, we consider learning of biped walking, since walking is one of most important and fundamental motor skills for humanoid robots. We propose a learning framework for CPG based biped locomotion with a policy gradient method. The arrangement of the CPG-based controller makes the motion of the robot constrained to walking, thus, making the learning problem more tractable. We demonstrate that a sensory feedback controller can appropriately adjust the rhythm of the CPG with the proposed learning method within a few hundred trials in simulations. We then consider the investigation of the linear stability of a periodic orbit of the acquired walking pattern making an allowance for its approximated return map. Furthermore, we apply the controllers acquired in numerical simulations to our physical 5-link biped robot in order to empirically evaluate the robustness of walking in the real environment. Experimental results demonstrate that the robot was able to successfully walk using the acquired controllers, even in the cases of an environmental changes: 1) by placing a seesaw-like metal sheet on the ground; and 2) a parametric change of the robot dynamics with an additional weight on a shank – which was not modeled in the numerical simulations. This demonstrates the robustness of the acquired controller.

3.1 Introduction

Biology can provide rich examples of robust control systems. Recently, there has been a growing interest in biologically inspired control approaches for biped locomotion using neural oscillators (*e.g.* (Matsuoka, 1985)) as the CPG. Notably, Taga (Taga et al., 1991) demonstrated the effectiveness of this approach for biped locomotion in achieving desired walking behavior in unpredicted environments in numerical simulations. Following this pioneering work, several attempts have been made to explore neural oscillator based control for legged locomotion (Fukuoka et al., 2003; Endo et al., 2004). Neural oscillators have desirable properties such as entrainment through interaction with the environment. However, in order to achieve the desired behavior of the oscillators, much effort is required in manually tuning their parameters. Our goal in this study is to develop an efficient learning framework for CPG-based locomotion of biped robots.

Past parameter optimization methods for CPG-based locomotion controllers have been proven successful, with genetic algorithm (Hase & Yamazaki, 1998) and value-function based RL (Sato et al., 2002), applied to determine the open parameters of the CPG while considering its high dimensional state space. However, these methods often require a large number of iterations to obtain the solution, and typically suffer from high computational costs with an increase of dimensionality of the state space. These undesirable features make it infeasible to directly apply these methods to real robots for real-time implementation.

In this chapter, we focus on learning appropriate sensory feedback to the CPG in order to achieve the desired walking behavior. The importance of sensory feedback to CPG in order to achieve adaptation to the environment was pointed out by (Taga et al., 1991). We propose a learning framework for a CPG-based biped locomotion controller using a policy gradient RL method for a 5-link biped robot (Fig. 3.1). The policy gradient method is a technique for maximizing an accumulated reward with respect to the parameters of a stochastic policy by trial-and-error in an unknown environment (Williams, 1992; Kimura & Kobayashi, 1998; Sutton et al., 2000; Baxter & Bartlett, 2001b; Konda & Tsitsiklis, 2003). However, the policy gradient method also suffers from high computational costs with an increase of dimensionality of the state space when the use of function approximator with a large number of parameters is desirable to achieve smooth control. Thus, in order to reduce the dimensionality of the state space used for learning, we only use partial physical states of the robot in our proposed learning system, *i.e.* we do not use internal states of the CPG and the rest of the states of the robot for learning. Although it is an approximation with only the use of partial states for learning, our CPG-arrangement introduces strong coordination between states of the robot and internal state of CPGs, thus, forming a tight coupling of the states of the robot and the CPGs. Based on the above insight, using only partial states of the robot and the CPG may be appropriate in making the complex motor learning task of walking more tractable. As a result, within this setting, we can regard the proposed learning framework as a partially observable Markov decision process (POMDP). Typical RL approaches, for example, TD-learning or Q-learning, find a deterministic optimal policy that maximizes the values of all the states simultaneously assuming that the environment has the Markov property (Sutton & Barto, 1998). However, as discussed in (Singh, Jaakkola, & Jordan, 1994), stochastic policies often show better performance than deterministic policies in POMDPs. Moreover, the effectiveness of a policy gradient method in POMDPs for a 4-legged robot (Kimura et al., 2001) has been empirically demonstrated, and also on a passive-dynamics based biped robot (Tedrake et al., 2004). We therefore elected to use a policy gradient method for our learning system among other possible RL methods.

This chapter is organized as follows: In Section 3.2, we introduce a CPG, which is used for the generation of walking behavior in our study. In Section 3.3, we describe a policy gradient RL method for a CPG-based biped locomotion controller. In Section 3.4, we present the proposed control architecture for our 5-link robot to achieve biped walking behavior. In Section 3.5, we first demonstrate the effectiveness of the proposed learning framework with numerical simulations. We then, investigate the linear stability of a periodic orbit of the acquired walking pattern considering its approximated return map. In Section 3.6, we present experimental results suggesting that successful biped walking with our physical 5-link robot can be achieved using the learned controller in numerical simulations. Furthermore, we analyze convergence properties of steady-state walking with variations of initial conditions based on a return map, and experimentally demonstrate the robustness of the acquired walking against environmental changes and parametric changes in the robot dynamics. In Section 3.7, we summarize this chapter and discuss the approaches of policy search in motor learning and control.

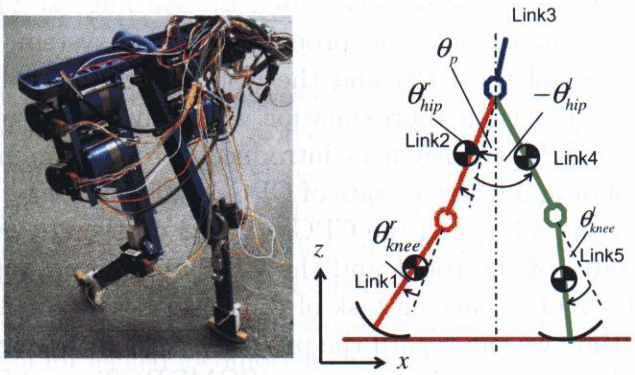


Figure 3.1: 5-link biped robot (left) and its model (right). x-z plane is defined as "sagittal plane".

3.2 Central pattern generator

The CPG-based controller is composed of a neural oscillator model and a sensory feedback controller, which maps the states of the robot to the input and to the neural oscillator model. In Section 3.2.1, we present the neural oscillator model. Then, we introduce the sensory feedback to the neural oscillator model.

3.2.1 Neural oscillator model

We use a neural oscillator model proposed by Matsuoka (Matsuoka, 1985). The oscillator dynamics of i-th unit are:

$$\tau_{CPG}\dot{z}_i(t) = -z_i(t) - \sum_{j=1}^{n} w_{ij}q_j(t) - \beta p_i(t) + z_0 + v_i(t), \tag{3.1}$$

$$\tau'_{CPG}\dot{p}_i(t) = -p_i(t) + q_i(t), \tag{3.2}$$

$$q_i(t) = \max(0, z_i(t)), \tag{3.3}$$

where n is the number of neurons, $z_i(t)$ and $p_i(t)$ are internal states of a CPG. τ_{CPG}, τ'_{CPG} are time constants for the internal states. w_{ij} is a inhibitory synaptic weight from the j-th neuron to the i-th neuron. z_0 is a bias. $v_i(t)$ is a feedback signal which will be defined in Eq.(3.4) below.

Sensory feedback

The feedback signal to the neural oscillator model $v_i(t)$ in Eq.(3.1) is given by

$$v_i(t) = v_i^{\max} g(a_i(t)), \tag{3.4}$$

where $g(a_i) = \frac{2}{\pi}\arctan\left(\frac{\pi}{2}a_i\right)$, and $v_i^{\max}$ is the maximum value of the feedback signal. The output of the feedback controller $\mathbf{a} = (a_1, \cdots, a_m)^T$ is sampled from a stochastic policy:

$$\begin{aligned}&\pi(\mathbf{x}, \mathbf{a}; \mathbf{W}^\mu, \mathbf{w}^\sigma)\\ &\quad = \frac{1}{(\sqrt{2\pi})^m |\mathbf{D}(\mathbf{w}^\sigma)|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{a} - \boldsymbol{\mu}(\mathbf{x}; \mathbf{W}^\mu))^T \mathbf{D}^{-1}(\mathbf{w}^\sigma)(\mathbf{a} - \boldsymbol{\mu}(\mathbf{x}; \mathbf{W}^\mu))\right),\end{aligned} \tag{3.5}$$

where $\mathbf{x}$ is partial states of the robot. $\mathbf{W}^\mu$ is the $m \times k$ parameter matrix, $\mathbf{w}^\sigma$ is the m-dimensional parameter vector of the policy, where m is the number of outputs, and k is the number of parameters. $\boldsymbol{\mu}(\mathbf{x}; \mathbf{W}^\mu)$ is the mean vector of the policy. The covariance matrix $\mathbf{D}$ is defined as $\mathbf{D}(\mathbf{w}^\sigma) = \mathbf{S}^T(\mathbf{w}^\sigma)\mathbf{S}(\mathbf{w}^\sigma)$. We can equivalently represent $\mathbf{a}$ by

$$\mathbf{a}(t) = \boldsymbol{\mu}(\mathbf{x}(t); \mathbf{W}^\mu) + \mathbf{S}(\mathbf{w}^\sigma)\mathbf{n}(t), \tag{3.6}$$

where $\mathbf{n}(t) \in \Re^m$ is a noise vector and $n_i(t)$ is sampled from the normal distribution with the mean of 0 and the variance of 1. Note that the matrix $\mathbf{S}(\mathbf{w}^\sigma)$ must be chosen such that $\mathbf{D}(\mathbf{w}^\sigma)$ is positive definite.

3.3 Learning sensory feedback to CPG with a policy gradient method

In this section we describe the use of a policy gradient method in order to acquire a policy of the sensory feedback controller to the neural oscillator model. In Section 3.3.1, we define the value function and temporal difference error (TD error) in continuous time and space (Doya, 2000), which is used in the policy gradient method. We then describe the learning method to improve the policy of the sensory feedback controller.

3.3.1 Learning the value function

Consider the dynamics of the robot including the CPG defined in continuous-time and continuous-states

$$\frac{d\mathbf{x}^{all}(t)}{dt} = f(\mathbf{x}^{all}(t), \mathbf{a}(t)), \tag{3.7}$$

where $\mathbf{x}^{all} \in X \subset \Re^l$ is all the states of the robot and the CPG, and $\mathbf{a} \in A \subset \Re^m$ is output of the feedback controller to the CPG. We denote the immediate reward as

$$r(t) = r(\mathbf{x}^{all}(t), \mathbf{a}(t)). \tag{3.8}$$

The value function of states $\mathbf{x}^{all}(t)$ based on a policy $\pi(\mathbf{x}^{all}, \mathbf{a})$ is defined as

$$V^{\pi}(\mathbf{x}^{all}(t)) = E\left\{ \int_t^{\infty} e^{-\frac{s-t}{\tau}} r(\mathbf{x}^{all}(s), \mathbf{a}(s))\, ds \,\middle|\, \pi \right\}, \tag{3.9}$$

where τ is a time constant for discounting future rewards. The consistency condition for the value function is given by the time derivative of Eq.(3.9) as

$$\frac{dV^{\pi}(\mathbf{x}^{all}(t))}{dt} = \frac{1}{\tau} V^{\pi}(\mathbf{x}^{all}(t)) - r(t). \tag{3.10}$$

We denote a current estimate of the value function as $V(\mathbf{x}^{all}(t)) = V(\mathbf{x}^{all}(t); \mathbf{w}^c)$, where $\mathbf{w}^c$ is the parameter of the function approximator. If the current estimate of the value function V is perfect, it should satisfy the consistency condition as Eq.(3.10). If this condition is not satisfied, the prediction should be adjusted to decrease the inconsistency

$$\delta(t) = r(t) - \frac{1}{\tau} V(t) + \dot{V}(t). \tag{3.11}$$

This is the continuous-time counterpart of the TD error (Doya, 2000). Because we consider a learning framework in POMDPs, *i.e.*, we observe only partial states $\mathbf{x}$ from all states $\mathbf{x}^{all}$, the TD error does not usually converge to zero. However, Kimura *et.al* (Kimura & Kobayashi, 1998) suggested that the approximated value function can be used to reduce the variance of the gradient estimation in Eq.(3.13) even if the consistency condition in Eq.(3.10) is not satisfied.

The update laws for the parameter vector of the value function $\mathbf{w}^c$ and the eligibility trace vector $\mathbf{e}^c$ for $\mathbf{w}^c$ are defined respectively as

$$\dot{\mathbf{w}}^c(t) = \alpha\delta(t)\mathbf{e}^c(t), \qquad \dot{\mathbf{e}}^c(t) = -\frac{1}{\kappa^c}\mathbf{e}^c(t) + \frac{\partial V_{\mathbf{w}^c}}{\partial \mathbf{w}^c}, \tag{3.12}$$

where α is the learning rate and κ^c is the time constant of the eligibility trace.

Learning a policy of the sensory feedback controller

In (Kimura & Kobayashi, 1998), Kimura *et al.* presented that by using TD error $\delta(t)$ and eligibility trace vector $\mathbf{e}^a(t)$, it is possible to obtain an estimate of the gradient of the expected actual return V_t with respect to the parameter vector $\mathbf{w}^a$ in the limit of $\kappa^a = \tau$ as

$$\frac{\partial}{\partial \mathbf{w}^a} E\{\ V_t \mid \pi_{\mathbf{w}^a}\} \ = E\{\delta(t)\mathbf{e}^a(t)\}, \tag{3.13}$$

where

$$V_t = \int_t^\infty e^{-\frac{s-t}{\tau}} r(s)\, ds, \tag{3.14}$$

$\mathbf{w}^a$ is the parameter vector of the policy $\pi_{\mathbf{w}^a} = \pi(\mathbf{x}, \mathbf{a}; \mathbf{w}^a)$, and $\mathbf{e}^a(t)$ is the eligibility trace vector for the parameter vector $\mathbf{w}^a$. The parameter vector $\mathbf{w}^a$ is represented as $\mathbf{w}^a = (\mathbf{w}_1^{\mu T}, \cdots, \mathbf{w}_m^{\mu\ T}, \mathbf{w}^{\sigma T})^T$, where $\mathbf{w}_j^\mu$ is j-th column vector of the parameter matrix $\mathbf{W}^\mu$. The update laws for the parameter vector of the policy $\mathbf{w}^a$ and the eligibility trace vector $\mathbf{e}^a(t)$ can be derived respectively as

$$\dot{\mathbf{w}}^a(t) = \beta\delta(t)\mathbf{e}^a(t), \qquad \dot{\mathbf{e}}^a(t) = -\frac{1}{\kappa^a}\mathbf{e}^a(t) + \frac{\partial \ln \pi_{\mathbf{w}^a}}{\partial \mathbf{w}^a} \tag{3.15}$$

where β is the learning rate and κ^a is the time constant of the eligibility trace. In the case of $\kappa^a = \tau \approx \infty$, the actual return used as a criterion for the policy improvement in this algorithm would be similar to the average reward criterion widely used in other policy gradient methods (Sutton et al., 2000; Baxter & Bartlett, 2001b; Konda & Tsitsiklis, 2003). (see Section 3.7.3 for more details)

3.4 Control architecture for 5-link robot

In this chapter, we use a planar 5-link biped robot (Fig. 3.1) developed in (Morimoto, Zeglin, & Atkeson, 2003). The experimental setting is depicted

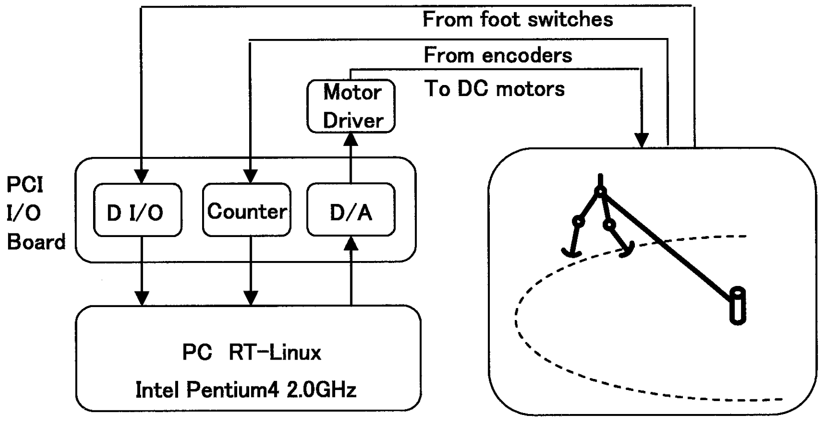


Figure 3.2: Experimental setting

in Fig. 3.2. The height of the robot is 0.4m and the total mass is about 2kg. The length of each link of the leg is 0.2m. The masses of the body, thigh and shank are 1.0kg, 0.43kg and 0.05kg, respectively. The motion of the robot is constrained within the sagittal plane which is defined as shown in Fig. 3.1(right) by a tether boom. Direct drive motors directly actuate the hip joints, and motors drive the knee joints through a wire transmission mechanism with a reduction ratio of 2.0. These transmission mechanisms with low reduction ratio provide high back drivability at the joints. Foot switches detect foot contact with the ground. The robot is an under-actuated system having rounded soles with no ankles. Thus, it is challenging to design a controller to achieve biped locomotion with this robot since no actuation can be applied between the stance leg and the ground unlike many of the existing biped robots that have flat feet with ankle joint actuation. In the following, we denote the left hip and knee angles by θ^l_{hip} and θ^l_{knee}, respectively. Similar definitions also apply to the joint angles of the right leg.

Figure 3.3 illustrates our control architecture for the biped robot, which consists of the CPG-based controller for the hip joints and the state-machine controller for the knee joints. Section 3.4.1 presents a CPG-based controller which generates periodic walking patterns. Section 3.4.2 presents a state-machine controller which ensure foot clearance at appropriate timing according to the state of the hip joint and the foot contact information with

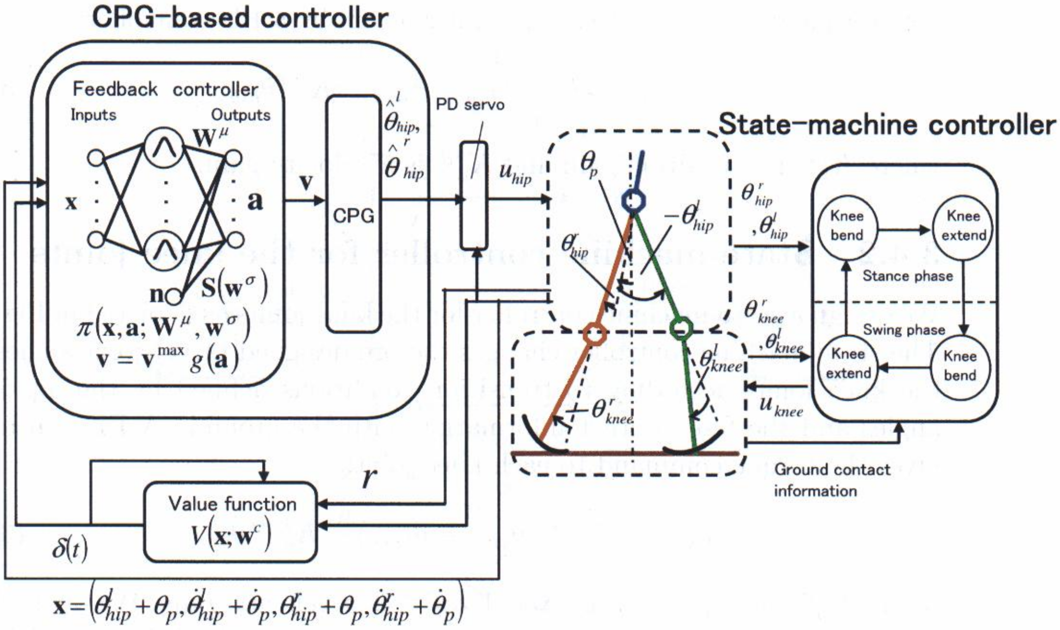

CPG-based controller
Feedback controller
Inputs
Outputs
$\mathbf{W}^{\mu}$
$\mathbf{x}$
$\mathbf{a}$
$\mathbf{n}$
$\mathbf{S}(\mathbf{w}^{\sigma})$
$\pi(\mathbf{x},\mathbf{a};\mathbf{W}^{\mu},\mathbf{w}^{\sigma})$
$\mathbf{v}=\mathbf{v}^{\max}g(\mathbf{a})$
$\mathbf{v}$
CPG
$\hat{\theta}^{l}_{hip}$, $\hat{\theta}^{r}_{hip}$
PD servo
u_{hip}
State-machine controller
θ_p
$-\theta^{l}_{hip}$
θ^{r}_{hip}
$-\theta^{r}_{knee}$
θ^{l}_{knee}
θ^{r}_{hip}, θ^{l}_{hip}
θ^{r}_{knee}, θ^{l}_{knee}
u_{knee}
Knee bend
Knee extend
Stance phase
Swing phase
Knee extend
Knee bend
Ground contact information
r
Value function
$V(\mathbf{x};\mathbf{w}^{c})$
$\delta(t)$
$\mathbf{x}=(\theta^{l}_{hip}+\theta_p,\dot{\theta}^{l}_{hip}+\dot{\theta}_p,\theta^{r}_{hip}+\theta_p,\dot{\theta}^{r}_{hip}+\dot{\theta}_p)$

Figure 3.3: Proposed control architecture for 5-link biped robot

the ground.

3.4.1 CPG-based controller for the hip joints

In the proposed control architecture, the hip joints of the robot are driven by the CPG-based controller described in Section 3.2. The hip joint controller is composed of four neurons ($i = 1 \sim 4$) in Eqs.(3.1) - (3.3): $i = 1$: extensor neuron for left hip, $i = 2$: flexsor neuron for left hip, $i = 3$: extensor neuron for right hip, $i = 4$: flexsor neuron for right hip. For the sensory feedback in Eq.(3.1), we consider the states of the hip joints $\mathbf{x} = (\theta^{l}_{hip} + \theta_p, \dot{\theta}^{l}_{hip} + \dot{\theta}_p, \theta^{r}_{hip} + \theta_p, \dot{\theta}^{r}_{hip} + \dot{\theta}_p)^T$ as the input states. The target joint angle for the hip joint is determined by the oscillator output q_i:

$$\hat{\theta}^{l}_{hip} = -q_1 + q_2, \qquad \hat{\theta}^{r}_{hip} = -q_3 + q_4. \tag{3.16}$$

The torque output u at each hip joint is given by a PD controller:

$$u_{hip} = K_p^{hip}(\hat{\theta}_{hip} - \theta_{hip}) - K_d^{hip}\dot{\theta}_{hip}, \tag{3.17}$$

where K_p^{hip} is a position gain and K_d^{hip} is a velocity gain.

3.4.2 State-machine controller for the knee joints

We design a state-machine controller for the knee joints as depicted in Fig 3.4. The state-machine controller changes the pre-designed target joint angles for the knee joints according to transition conditions defined by the hip joint angles and the foot contact information with the ground. A PD controller gives the torque command to each knee joint:

$$u_{knee} = K_p^{knee}(\hat{\theta}_{knee} - \theta_{knee}) - K_d^{knee}\dot{\theta}_{knee}, \tag{3.18}$$

where K_p^{knee} is a position gain and K_d^{knee} is a velocity gain. We define four target joint angles, $\theta_1 \sim \theta_4$, for the state-machine controller (Fig. 3.4). We use the hip joint angles and the foot contact information to define the transition conditions of the state-machine controller. The transition conditions defined by the hip joint angles are given by, $\theta^l_{hip} - \theta^r_{hip} < b$ or $\theta^r_{hip} - \theta^l_{hip} < b$, where b is a threshold of the transition conditions.

3.5 Numerical simulations

In Section 3.5.1, we present function approximators for the value function and policy of sensory feedback to CPG. In later part of this section, we describe reward function designed to achieve biped walking through learning. In Section 3.5.2, we present the parameter settings in the controller used for the simulations.

3.5.1 Function approximator for the value function and the policy

We use a normalized Gaussian Network (NGnet) (Doya, 2000) to model the value function and the mean of the policy. This function approximator was used in previous studies of RL in continuous time and space, and shown

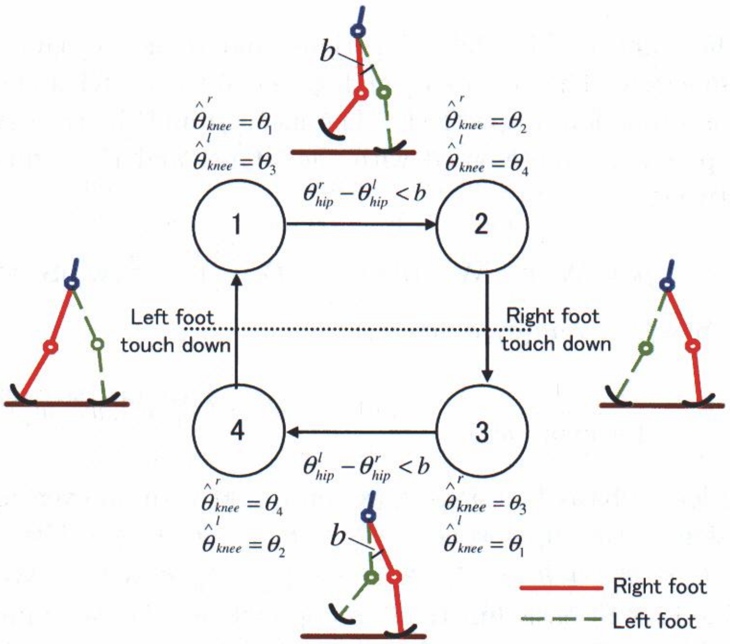

b
$\hat{\theta}^r_{knee} = \theta_1$
$\hat{\theta}^l_{knee} = \theta_3$
$\hat{\theta}^r_{knee} = \theta_2$
$\hat{\theta}^l_{knee} = \theta_4$
$\theta^r_{hip} - \theta^l_{hip} < b$
1
2
Left foot touch down
Right foot touch down
4
3
$\theta^l_{hip} - \theta^r_{hip} < b$
$\hat{\theta}^r_{knee} = \theta_4$
$\hat{\theta}^l_{knee} = \theta_2$
b
$\hat{\theta}^r_{knee} = \theta_3$
$\hat{\theta}^l_{knee} = \theta_1$
Right foot
Left foot

Figure 3.4: State transition 1 ∼ 4 in the state-machine controller for knee joints

to be effective in the examples of a swing-up task of an inverted pendulum and a dynamic stand-up behavior of a real robot (Doya, 2000; Morimoto & Doya, 2001). With this method it is possible to achieve smooth control compared to the tile-coding approach often used in discrete RL (Sutton & Barto, 1998). In addition, practical feasibility of this function approximator was demonstrated for real-time implementation of the control policy on a hardware robot to achieve the desired behavior (Morimoto & Doya, 2001). The variance of the policy is modeled by a sigmoidal function (Kimura & Kobayashi, 1998). The value function is approximated with the NGnet:

$$V(\mathbf{x}; \mathbf{w}^c) = \mathbf{w}^{cT}\mathbf{b}(\mathbf{x}) \tag{3.19}$$

where, $\mathbf{b}(\mathbf{x}) = (b_1(\mathbf{x}), b_2(\mathbf{x}), \cdots, b_K(\mathbf{x}))^T$,

$$b_k(\mathbf{x}) = \frac{\phi_k(\mathbf{x})}{\sum_{l=1}^{K}\phi_l(\mathbf{x})} \quad \text{and} \quad \phi_k(\mathbf{x}) = e^{-\|\mathbf{s}_k^T(\mathbf{x}-\mathbf{c}_k)\|}. \tag{3.20}$$

K is the number of the basis functions, and $\mathbf{w}^c$ is the parameter vector of value function. The vectors $\mathbf{c}_k$ and $\mathbf{s}_k$ define the center and the size of the k-th basis function, respectively. The mean $\boldsymbol{\mu}$ and the covariance matrix $\mathbf{D}$ of the policy are represented with the NGnet and the sigmoidal function, respectively:

$$\boldsymbol{\mu}(\mathbf{x};\mathbf{W}^{\mu}) = \mathbf{W}^{\mu T}\mathbf{b}(\mathbf{x}), \qquad \mathbf{D}(\mathbf{w}^{\sigma}) = \mathbf{S}^T(\mathbf{w}^{\sigma})\mathbf{S}(\mathbf{w}^{\sigma}), \tag{3.21}$$

where, $\mathbf{S}(\mathbf{w}^{\sigma}) = diag(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$,

$$\sigma_i = \frac{1}{1+\exp(-w_i^{\sigma})} \qquad \text{and} \qquad \mathbf{w}^{\sigma} = (w_1^{\sigma}, w_2^{\sigma}, w_3^{\sigma}, w_4^{\sigma})^T. \tag{3.22}$$

We locate basis functions $\phi_k(\mathbf{x})$ on a grid with an even interval in each dimension of the input space $(-\frac{\pi}{3} \leq \theta^l_{hip} + \theta_p \leq \frac{\pi}{3}, -3.0\pi \leq \dot{\theta}^l_{hip} + \dot{\theta}_p \leq 3.0\pi, -\frac{\pi}{3} \leq \theta^r_{hip} + \theta_p \leq \frac{\pi}{3}, -3.0\pi \leq \dot{\theta}^r_{hip} + \dot{\theta}_p \leq 3.0\pi)$. We used $9216 (= 12 \times 8 \times 12 \times 8)$ basis functions to approximate the value function and the mean of the policy respectively.

Rewards

We used the following simple reward function:

$$r = k_{\nu} \max(0, \nu), \tag{3.23}$$

where, the reward is designed to encourage forward progress of the robot by giving a reward proportional to the forward velocity of walking ν. In this study, the parameter for the reward is chosen as $k_{\nu} = 0.05$. The robot also receives a punishment (negative reward) $r = -1$ for 0.5s if it falls over.

3.5.2 Parameters for the controllers

Parameters of the neural oscillators used in Eqs.(3.1) - (3.3) are $\tau_{CPG} = 0.041$, $\tau'_{CPG} = 0.36$, $\beta = 2.5$, $z_0 = 0.4$, $w_{12} = w_{21} = w_{34} = w_{43} = 2.0$, $w_{13} = w_{31} = w_{24} = w_{42} = 1.0$. Initial values of the internal states are given by $z_1(0) = 0.05$, $z_2(0) = 0.05$. We select the learning parameters as $\tau = 1.0$, $\alpha = 50$, $\beta^{\mu} = 20$, $\beta^{\sigma} = 10$, $\kappa^c = 0.1$, $\kappa^{\mu} = 1.0$, $\kappa^{\sigma} = 1.0$. PD gains for hip joints are set to $K_p^{hip} = 4.0$N·m/deg and $K_d^{hip} = 0.07$N·m·s/deg, respectively. These CPG parameters were roughly tuned to achieve some desirable natural frequency and amplitude through numerical simulations. However, note that as seen in Fig.3.6, the robot cannot walk only with the CPG, *i.e.*, appropriate learned sensory feedback is necessary for successful walking. Moreover, we will demonstrate that choice of CPG parameters does not significantly affect the performance of learning in our proposed framework (see Section 3.5.3 below). Parameters of the state-machine are $\theta_1 = 32$deg, $\theta_2 = 16$deg, $\theta_3 = 15$deg, $\theta_4 = 7.5$deg and $b = 8.6$deg. PD gains for knee joints are chosen as $K_p^{knee} = 8.0$N·m/deg and $K_d^{knee} = 0.09$N·m·s/deg, respectively.

3.5.3 Simulation results

In the following simulations, the initial posture of the robot is determined as $\theta^l_{hip} = 5.5$deg, $\theta^r_{hip} = -5.5$deg, $\theta_p = 0.0$deg, $\theta^l_{knee} = 20.5$deg, $\theta^r_{knee} = 0.0$deg (see the definition of each angle in Fig. 3.3) and the initial velocity of the robot is randomly sampled from an uniform distribution between 0.05m/s and 0.20m/s. In these simulations, we define that a learning episode is successful when the biped robot does not fall over for 10 successive trials. We applied the policy gradient method with these settings to the biped robot. Figure 3.5 (solid line) shows an accumulated reward at each trial with the policy gradient method. An appropriate feedback controller of the CPG-based controller was acquired in 181 trials (averaged over 50 experiments). Figure 3.6(1) shows the initial walking pattern before learning, where the robot falls over after a few steps. Figure 3.6(2) shows an acquired walking pattern at the 1000-th trial with the learned sensory feedback of the CPG-based controller.

As a comparison, we also implemented a value-function-based RL method proposed in (Doya, 2000). The result is also presented in Fig.3.5 (dash-dot line). Although the value-function-based RL could also acquire appropriate biped walking controllers, it required a larger number of trials compared with the policy gradient method (1064 trials was required with the value-function-based RL on average). Moreover, we observed that learning was unstable with higher learning rate in updating of policy parameters with value-function-based RL. The result is consistent with the notion that value-function-based RL methods are not suitable for POMDPs, as pointed out in (Singh et al., 1994). This is further discussed in Section 3.7.3 in more detail.

We observed a large phase difference between the target and actual trajectories in the hip joints while the knee joint trajectories achieved good tracking of the target. This is due to the choice of low PD gains for the hip joints. Despite this large phase difference between the target and actual hip joint trajectories, the robot could achieve successful walking. This suggests that our method does not necessarily require very accurate tracking with a high gain servo which is typically offered in model-based trajectory planning approaches (Kagami et al., 2001; Kagami, Kitagawa, Nishiwaki, sugihara, & Inaba, 2002; Hirai et al., 1998).

In order to investigate sensitivity of learning against the changes in the CPG parameters, we varied the CPG parameters which characterize the frequency (τ_{CPG}, τ'_{CPG}) and amplitude (z_0) from the values chosen above by ±25%, respectively. In all cases, we could acquire successful walking within 1000 trials. We did not observe significant difference in the resultant walking with the learned feedback controller even with these varied CPG parameters. This suggests that careful tuning of CPG parameters is not a prerequisite in our learning framework.

3.5.4 Linear stability of a periodic orbit in learned biped walking

In this section, we analyze linear stability of a periodic orbit in learned biped walking around a fixed point using a return map (Strogatz, 1994). The return map is defined as a mapping of the states of the robot and CPG from the 4th step to the 6th step when the right hip is in swinging phase and the angle is 0.2rad. The return map is an 18 dimensional mapping which consists of the states of the robot and CPG except the walking distance of the robot. Initial velocity of the robot was randomly sampled from an uniform distribution between 0.05m/s and 0.15m/s, introducing perturbations in each dimension.

We analyzed the linearized return map which was approximated using 1500 sampled data, and confirmed all eigenvalues were inside of the unit circle. The result implies that the periodic biped walking is locally stable around the fixed point.

3.6 Hardware experiments

In this section, we implement the proposed control architecture on the physical biped robot depicted in Fig. 3.1. We use the same parameters for the CPG and state-machine and also the same PD gains as used in the numeri-

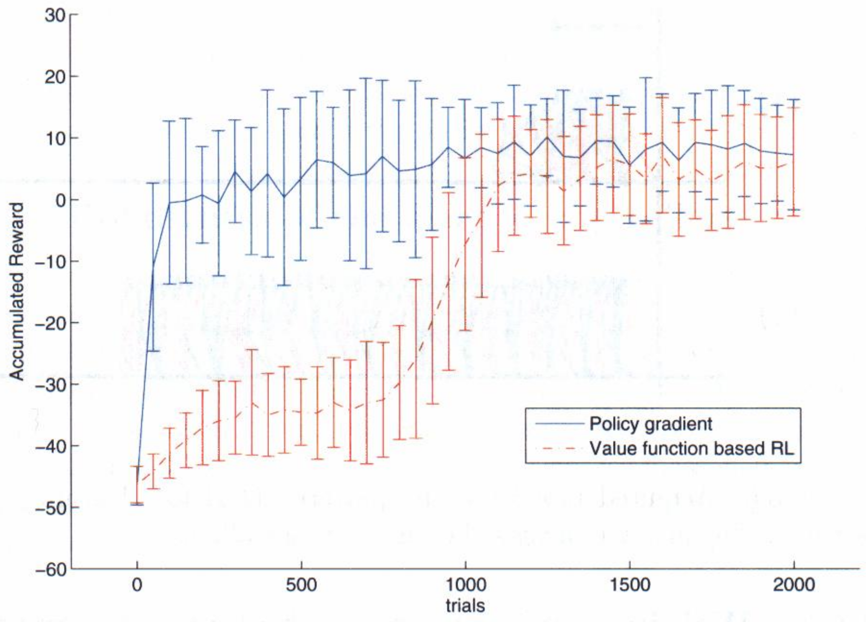


Figure 3.5: Accumulated reward at each trial: Averaged by 50 experiments and smoothed out by taking a 10-moving average. Error bar is the standard deviation. Solid line: policy gradient method. Dash-dot line: value-function-based RL.

cal simulations presented in Section 3.5.2. In the state-machine controller, a low-pass filter, with the time constant of 0.03s, is used to avoid discontinuous change in the target angles of the knee joint, which is practically undesirable. To initiate locomotion in the experiments, we first suspend the robot with the legs swinging in the air, and then place the robot on the ground manually. Thus, the initial condition of each run was not consistent. Occasionally, the robot could not start walking or fell over after a couple of steps when the timing was not appropriate.

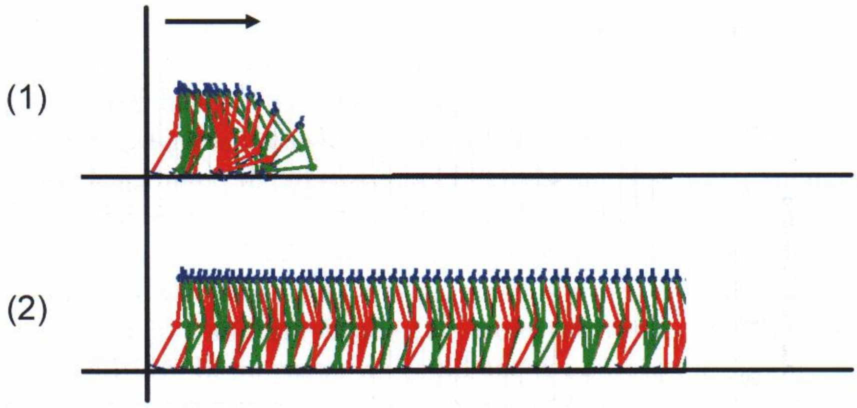


Figure 3.6: Acquired biped walking pattern: (1)Before learning. (2)After learning. The arrow indicates the direction of walking.

3.6.1 Walking performance of the learned controller in the real environment

We implemented ten feedback controllers acquired in the numerical simulations, and then we confirmed that seven controllers out of ten successfully achieved biped locomotion with the physical robot. Figure 3.7 shows the walking pattern without a learned feedback controller, and Fig.3.8 shows snapshots of a walking pattern using one of the feedback controllers. Figure 3.11 presents trajectories of a successful walking pattern at each joint in right foot.

3.6.2 Convergence property from various initial conditions

The robot could achieve biped walking even though the initial conditions in these experiments were not consistent. In order to investigate the convergence property to steady-state walking with variations in initial conditions, we analyze linear stability of a periodic orbit in learned biped walking around a fixed point using its return map (Strogatz, 1994). We consider a one dimensional return map with respect to the successive step length d defined as a distance between the right and left foot when right leg touches down with the ground. In Fig 3.12, we plot the return map obtained in the experiments. The absolute value of the slope of the return map is 0.82 which is less than 1. The result implies that the walking with the physical robot converges to the steady-state walking even if the initial conditions are not consistent.

3.6.3 Robustness of the learned controllers

We experimentally investigate the robustness of the learned controller against environmental changes and parametric changes in the robot dynamics. As an example of an environmental change, we placed a seesaw-like metal sheet with a slight change in the slope on the ground (Fig.3.9). As an example of a parametric change in the robot dynamics, we added a weight (150g) on the right shank (Fig.3.10), which is about 38% of right leg mass increase. Figures 3.9 and 3.10 suggest the robustness of the learned walking against environmental changes and parametric changes in the robot dynamics, respectively.

3.7 Summary and discussion

3.7.1 Summary

In this chapter, we presented a learning framework for a CPG-based bipedal walking controller with a policy gradient method. Numerical simulations demonstrated that an appropriate sensory feedback controller to the CPG could be acquired with the proposed learning architecture to achieve a desired walking behavior. We showed that the acquired walking pattern has a locally stable periodic orbit based on a linearized return map around a fixed point. We also implemented the learned controller on the physical 5-link biped robot. We then analyzed the convergence property of the learned walking behavior, for steady-state walking under variations of initial conditions based on a return map. Experimentally we demonstrated the robustness of the learned controller against environmental changes and parametric changes in robot dynamics, such as placing a seesaw-like metal sheet on the ground, as well as adding a weight to the physical robot. As an immediate next step, we address improvement of the acquired controller by additional learning on the physical robot.

3.7.2 Issues in motor skill learning with reinforcement learning

In this study, our general interest is in the acquisition of motor skills or dynamic behavior of complex robotic systems such as humanoid robots. This chapter has focused on the development of a learning framework for a simple biped robot to achieve the desired walking behavior. Among learning motor skill problems, in particular, learning biped locomotion is a challenging task, which involves dynamic interaction with the environment, and it is desirable that the controller be robust enough to deal with uncertainties of the environment and unexpected disturbances.

Model-based approaches for motion generation of biped robots have been successfully demonstrated to be effective (Kagami et al., 2001, 2002; Hirai et al., 1998). However, they typically require precise modeling of the dynamics of the robot and the structure of the environment. For that reason, we employed our proposed CPG-based control framework with a policy gradient method, which does not require such precise modeling to achieve robust locomotion in an unstructured environment. Our empirical results demonstrated the effectiveness of the proposed approach. However, in general, there are difficulties in the application of the reinforcement-learning framework to motor skill learning with robotic systems. First, in motor control, it is desirable to use smooth continuous actions, *i.e.*, the output of the policy should be smooth and continuous, computed from the current state, which is typically measured by sensors taken from the real robotic systems. Previously, in many applications of RL, discretization techniques have been widely used to provide continuity (Sutton & Barto, 1998). However, as pointed out in (Doya, 2000), coarse discretization may result in poor performance, and fine discretization would require a large number of states and iteration steps. Thus, in order to deal with continuous state and action, we find it useful to use function approximators. Moreover, the use of algorithms derived in continuous time is also suitable for dynamic systems (Doya, 2000). Secondly, when considering hardware implementation of the policy for robot control, as calculation of motor commands needs to be performed in real-time. Thus, computationally efficient representation of the policy should be considered. To our knowledge, there have been only few successful applications of motor skill learning for physical robotic systems (Kimura et al., 2001; Morimoto & Doya, 2001), in which the dimensionality of the systems is kept relatively small. In this research, we used CPG-based controller to achieve robust biped walking for rather high dimensional system. The use of CPG-based controller also makes learning of such a complex motor task much simpler as it introduce the periodic rhythm required for walking. However, still other alternative approaches and algorithms can be considered. In the following section, we discuss several possible policy search approaches that might be applicable to the learning problem in this chapter.

3.7.3 Comparison to alternative policy search approaches

In this chapter, we adopted a policy gradient method proposed in (Kimura & Kobayashi, 1998), as a method for policy search in a CPG-based locomotion controller. This section discusses possible alternative policy search approaches, for example, genetic algorithms (Hase & Yamazaki, 1998; Tuchiya, Kimura, & Kobayashi, 2004), value-function-based reinforcement (Sutton & Barto, 1998; Doya, 2000), and other policy gradient algorithms such as GPOMDP (Baxter & Bartlett, 2001b) and IState-GPOMDP (Aberdeen &

Baxter, 2002).

Genetic algorithms (GAs) are optimization methods inspired by evolutionary processes. This method is known to be effective for applying to complex search problems (typically discrete problems) in a large space, it has been applied to policy search in biped locomotion (Hase & Yamazaki, 1998), and to locomotion of a snake-like robot (Tuchiya et al., 2004). However, the optimization process does not use the gradient information, which is useful to determine how to improve the policy in a systematic manner. Also, there are a number of open design parameters, for example, the number of individuals and the probability of mutation, which need to be determined somewhat in a heuristic manner. Moreover, there is a problem of policy coding – it is unclear how to represent a policy in an appropriate way for any given problem.

Value function based RL methods have been successfully applied to many policy search problems (Doya, 2000; Morimoto & Doya, 2001; Tesauro, 1994; Mataric, 1994). However, value function based RL assumes MDPs (Markov decision process) in which all states are observable, therefore, it is not suitable for POMDPs as pointed out in (Singh et al., 1994). In fact, we performed additional numerical simulations to test a value function based RL for the locomotion task in the same simulation settings, which could conceived as a POMDP (see the result in Fig. 3.5). However, the value function based RL[1] needed a larger number of trials to acquire an appropriate feedback controller compared with the policy gradient method (Kimura & Kobayashi, 1998) in this POMDP environment. There is still a possibility to consider full state observation including all the robot states and the internal states of the CPG to make the learning problem of biped locomotion an MDP. However, due to the significant increase of dimensionality of the state space, it is computationally too expensive for real-time implementation on a hardware system with the current settings. These observations above indicate that policy search algorithms that are capable of handling the POMDP situation would be preferable.

Policy gradient methods are policy search techniques which are suitable for POMDPs (Williams, 1992; Kimura & Kobayashi, 1998; Baxter & Bartlett, 2001b). In this chapter, we chose to use the policy gradient algorithm proposed in (Kimura & Kobayashi, 1998) as a policy search method,

[1] We used the value function based RL in continuous time and state proposed in (Doya, 2000) and also used for the real robot control in (Morimoto & Doya, 2001).

which has been empirically shown to be effective as a learning method for physical legged robotic systems (Kimura et al., 2001; Tedrake et al., 2004). We would like to mention that this algorithm is essentially equivalent to the policy gradient algorithm, GPOMDP, developed by Baxter (Baxter & Bartlett, 2001b). Although objective functions used in Kimura's algorithm (expected actual return) and Baxter's GPOMDP algorithm (average reward) are different, (Baxter & Bartlett, 2001b) shows that the gradients of the expected actual return is proportional to the gradient of the average reward. Both, Kimura's formulation and Baxter's formulation obtain a gradient of the average reward with respect to the policy parameters as long as the probability distribution of all the states and actions are known, *i.e.*, the environment is completely known. However, in practice, we need to estimate the environment's dynamics from sampled data when there is no prior knowledge of the environment. In such a case, Kimura's algorithm which uses an approximated value function as *the reward baseline* (introduced in (Williams, 1992)) is empirically shown to be more advantageous in reducing the variance of the estimation of the policy gradient (Kimura & Kobayashi, 1998).

Finally, we would like to mention the internal-state policy gradient algorithm for POMDP (IState-GPOMDP), which has internal states with memory as an extension of the GPOMDP algorithm (Aberdeen & Baxter, 2002). Conceptually, this framework has a similarity to the structure of our learning system with the CPG, in that it contains internal states. Thus, there might be a potential possibility to optimize the parameters of the learning system including the mapping from the oscillator output to the torque at hip joints, which was implemented by a pre-designed PD controller in Eq.(3.17). However, learning additional parameters would be computationally too expensive due to the complex representation of the entire policy, and therefore would not be suitable for real-time implementation in a hardware system.

Although policy gradient methods are generally considered to be suitable for POMDPs, these methods find a local optimal solution only within the parameter space of the state-dependent policy designed in advance. In this study, we manually selected the partial states (only hip joint states) for the policy from all the states including the robot and the CPG. Because of this simplification, the real-time implementation was achieved. On the other hand, this simplification might deteriorate the performance of the resultant policy acquired through learning. One of the key factors for successful learning in this study was the choice of that partial states selected by our intuition, which are likely to be dominant states in our proposed CPG-based biped locomotion controller. If a different simplification is introduced for this learning task, for example, if CPG's internal states are only used for learning, the acquired controllers may not be good enough to achieve biped walking. We leave this for our future studies.

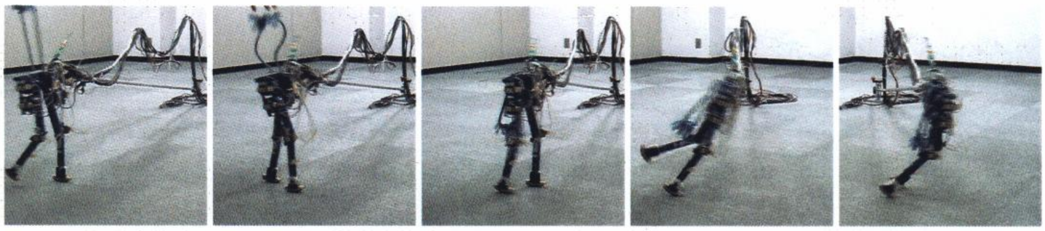


Figure 3.7: Initial walking pattern without a feedback controller. The robot could not walk.

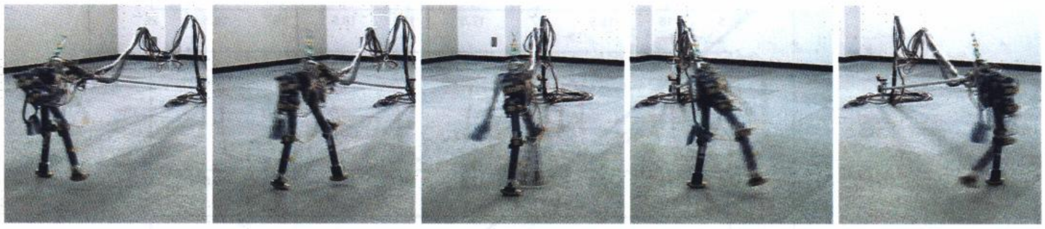


Figure 3.8: Successful walking pattern with a learned feedback controller in numerical simulation with 1000 trials.

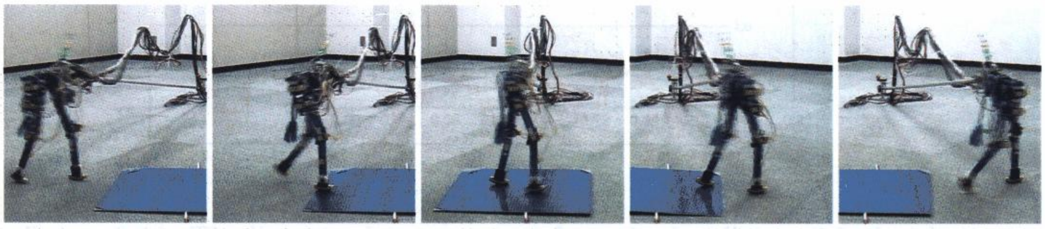


Figure 3.9: Example of an environmental change. Walking pattern on a seesaw-like metal sheet

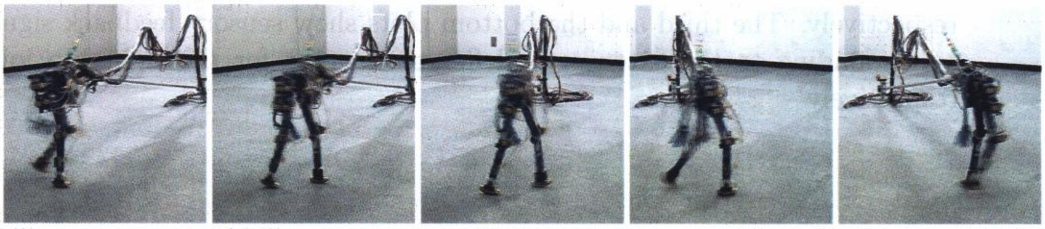


Figure 3.10: Example of a parametric change of the robot dynamics. Walking pattern with an additional weight of 150g on the right shank.

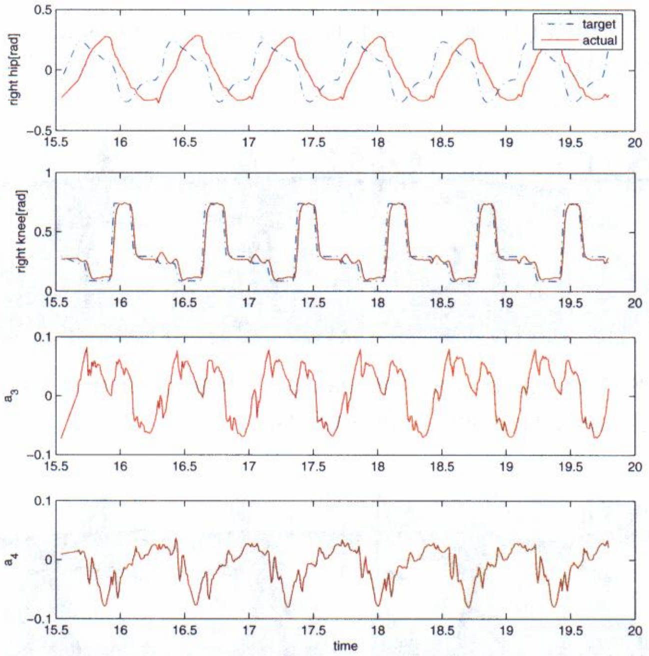


Figure 3.11: Joint angles and sensory feedback signals of successful walking with the physical robot using an controller acquired in numerical simulations. The top and second plots are joint trajectories of the right hip and knee, respectively. The third and the bottom plots show sensory feedback signals corresponding to the extensor and flexsor neurons for the right hip joint, respectively.

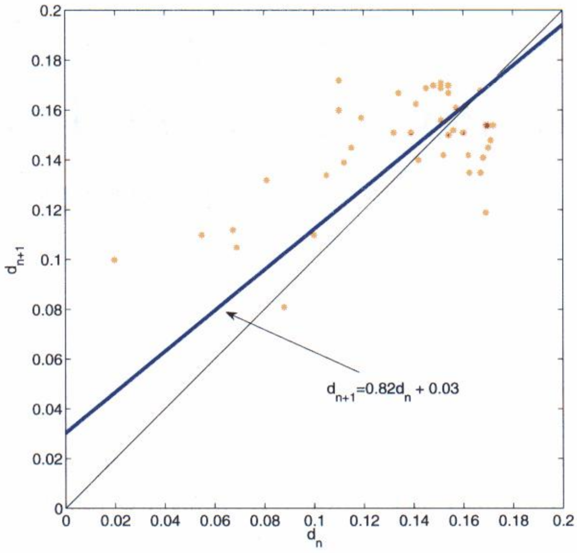


Figure 3.12: The linearized return map of acquired walking with the physical robot. d is a step length when the right leg touches down with the ground. The thick line is the return map from d_n to d_{n+1}, and the thin line represents the identity map.

Chapter 4

Achieving Dynamic Tasks through Learning Humanoid CoM Movements

In the previous chapter, it was presented that the learning of a biped walking behavior was successfully achieved via the use of a policy gradient RL method on a 5-link robot. A neural oscillator model was introduced as CPGs, which imposes natural joint coordination for walking while making the complex motor learning problem tractable. Although the result in previous chapter demonstrates that learning in low-dimensional feature space could achieve motor learning on humanoid robots, it is still faced with the problem, as to how we can find an appropriate constraint, which could make dynamics of the robot reduced one with a capability for executing the target motor task. One's ideal hope is to find a low dimensional feature space that can be commonly effective for all motor learning for all-daily movements. However, several studies (d'Avella et al., 2003; Tresch et al., 2006; Safonova et al., 2004) suggest that the basis of synergies is typically task dependent, even if a few common one can be observed. Additionally, we could look toward a general principle to automatically extract such a task-dependent and low-dimensional features. However, due to the countless variety of the movements, this could proof to be unrealistic. Now let us first consider a general framework for some-limited class of movements.

In this chapter, we focus on a class of whole-body motor skills, in which all-joints are cooperative and momentum or CoM movements are correlated with its performance, because it can be expected for such tasks that the learning is achieved in extremely low-dimensional feature space compared with original dimension of the humanoid robots. In particular, this chapter presents a novel approach to acquire whole-body dynamic movements on humanoid robots, with a focus on learning a control policy for the CoM, rather than joint movements or other end-effectors in achieving its task. The direct CoM control keeps the ZMP in the supporting polygon even during learning trials; the ZMP equation is taken into account for the CoM control. The weighted pseudo inverse with CoM Jacobian is then utilized to distribute the CoM movement to all-joint movement. This approach can allow us to execute learning in low-dimensional feature space which consists of CoM position and executing time variable, and results in a coordinated full-body movement. The proposed method is applied for a ball-punching task as one of dynamic full-body movements with a full-body humanoid robot in demonstrating the effectiveness of this method.

4.1 Introduction

A number of methods for achieving various tasks on a humanoid robot have been explored, mainly to achieve biped walking and balancing (Nagasaka, 2000; Kagami et al., 2001; Sugihara & Nakamura, 2002; Kajita et al., 2003). However, even though a number of real humanoid robots have demonstrated dynamic whole-body movements based on these methods, it is still infeasible to set up humanoid robots into our own living spaces in order to help us, due to their lack of ability to adapt to a new environment like humans and animals.

RL is a promising method compared with other learning frameworks as one of the candidate solutions for granting humanoid robots with such ability. With an increase of dimensionality in state and action space, however, RL often requires not only a large number of iterations, but also has a large computational cost, especially in the case of learning a complex control policy. Although there have been many attempts to apply RL methods for several robots in simulation and real hardware systems in order to acquire a desired movement, most of the robots to which learning has been applied so far have a small number of DoFs, and not as many as 20 to 30 DoFs as typically offered by humanoid robots (Kimura et al., 1997, 2001; Morimoto & Doya, 2001; Tedrake et al., 2004; Matsubara et al., 2005).

In this chapter, we suggest a novel approach for acquiring dynamic whole-body movements on humanoid robots, focused on learning a control policy for the CoM of the robot rather than joint trajectories. Humanoid robots cannot however, directly control their own CoM through joint torques, because they are not constrained by the ground. Moreover, unreasonable target joint trajectories are infeasible, because they could make the robot fall over due to dynamic inconsistency. It has been shown that in order to achieve a desirable CoM movement, it is important to directly control the ground reaction force (Nagasaka, 2000). Furthermore, if we keep the ZMP (Vukobratović & Borovac, 2004) in the support polygon during CoM control, it may be possible to prevent robots from falling over due to moments on the edges of their feet. A CoM-Jacobian-based redundancy resolution is used to compute angular velocities for all joints to achieve a whole-body movement consistent with the desired CoM movement was proposed by (Sugihara & Nakamura, 2002). The above framework makes the learning problem of whole-body movement a more reasonable task.

We demonstrate the effectiveness of our proposed framework by applying it to a ball-punching task in numerical simulations using a commercial humanoid robot, HOAP2.

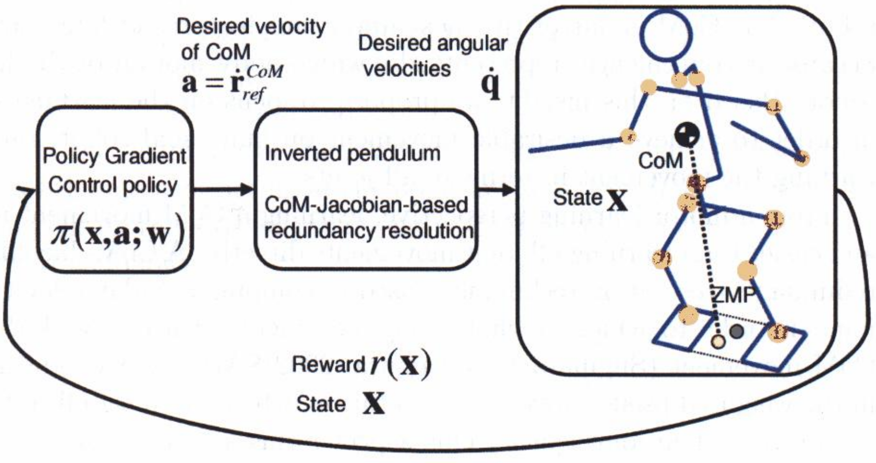

Desired velocity of CoM
$\mathbf{a} = \dot{\mathbf{r}}_{ref}^{CoM}$
Desired angular velocities
$\dot{\mathbf{q}}$
Policy Gradient Control policy
$\pi(\mathbf{x},\mathbf{a};\mathbf{w})$
Inverted pendulum
CoM-Jacobian-based redundancy resolution
State $\mathbf{X}$
CoM
ZMP
Reward $r(\mathbf{X})$
State $\mathbf{X}$

Figure 4.1: An approach for learning a desired whole-body movement on a humanoid robot

The organization of this chapter is as follows. In Section 4.2, we briefly introduce the ZMP and the ZMP-equation (see the details in Appendix B). Next we describe how we can control the CoM by manipulating the ZMP based on the ZMP equations in Section 4.3. In Section 4.4, we present the policy-gradient method we use to learn an appropriate control policy for a desired full-body movement on a humanoid robot. In Section 4.5, we present a concrete example of the learning system for a ball-punching task on a humanoid robot. In Section 4.5.1, we describe the results achieved by applying the proposed method in numerical simulations. In Section 4.6, we implement the acquired CoM movement on the humanoid robot HOAP2, and demonstrate the effectiveness of our method in a real environment. Finally, we summarize this chapter.

4.2 An approach for learning a desired whole-body movement on a humanoid robot

In this section, we briefly describe our suggested approach for learning a desired whole-body movement on a humanoid robot. Figure 4.1 shows a rough sketch of our suggested approach. The approach focuses on learning a CoM movement suitable for the achievement of the task on a humanoid robot. The CoM is one of the most important features of humanoid robots because it conveniently represents the whole-body motion of the humanoid robot. Based on this insight, we propose to focus on the learning the CoM in order to achieve a desirable movement on humanoid robot, rather than learning the movement in terms of all joints.

From a motor learning perspective, learning a CoM movement is simpler and easier than learning all joint movements directly. A CoM-Jacobian-based redundancy-resolution technique is used to compute angular velocities for all joints in order to achieve a whole-body movement consistent with the desired CoM movement (Sugihara & Nakamura, 2002). We use a weighting matrix in the weighted pseudo-inverse computation, which has a significant effect on the movement in joint space. This aspect forms a focus in next chapter. In this work, we use weights that were manually specified. The following two sections describe the components of the method: CoM control based on the ZMP, distribution of a CoM movement into joint space, and RL of a CoM movement.

4.3 CoM controller based on the ZMP equation

As mentioned in Section 4.1, for the last decade, many humanoid robots have achieved various dynamic tasks such as biped walking and balancing. Most proposed methods are based on the ZMP – an equation representing the dynamics of a humanoid robot's CoM in an approximated manner (see Appendix B). This section describes a method for achieving control of the CoM based on the ZMP.

4.3.1 ZMP compensation control

According to Nagasaka (Nagasaka, 2000), assuming a mass-concentrated model, the relationship between the moment acting on the ZMP and the objective ZMP is given as

$$\boldsymbol{n}^{ZMP} = \boldsymbol{n}^{OZMP} + (\boldsymbol{r}^{OZMP} - \boldsymbol{r}^{ZMP}) \times \boldsymbol{f}^{CoM}, \tag{4.1}$$

$$\boldsymbol{n}^{OZMP} = (\boldsymbol{r}^{CoM} - \boldsymbol{r}^{OZMP}) \times \boldsymbol{f}^{CoM}, \tag{4.2}$$

where $\boldsymbol{n}^{ZMP} \in \boldsymbol{R}^3$ and $\boldsymbol{n}^{OZMP} \in \boldsymbol{R}^3$ are the moments on the ZMP and objective ZMP respectively. $\boldsymbol{r}^{ZMP} \in \boldsymbol{R}^3$ and $\boldsymbol{r}^{OZMP} \in \boldsymbol{R}^3$ are the position vector of ZMP and objective ZMP from origin, and $\boldsymbol{f}^{CoM} \in \boldsymbol{R}^3$ is the force acting on the CoM, respectively

From the definition of the ZMP: the point such that horizontal components of the moment acting at the point are zero, we can derive a control law to compensate the ZMP to the objective ZMP by kinematically manipulating the CoM as follows:

$$\begin{aligned} \Delta r^{CoM}_{x,i+1} &= K(r^{ZMP}_x - r^{OZMP}_x) + \Delta r^{CoM}_{x,i} \\ &+(\Delta r^{CoM}_{x,i} - \Delta r^{CoM}_{x,i-1}) + K\Delta r^{CoM}_{x,i}, \end{aligned} \tag{4.3}$$

$$\begin{aligned} \Delta r^{CoM}_{y,i+1} &= K(r^{ZMP}_y - r^{OZMP}_y) + \Delta r^{CoM}_{y,i} \\ &+(\Delta r^{CoM}_{y,i} - \Delta r^{CoM}_{y,i-1}) + K\Delta r^{CoM}_{y,i}, \end{aligned} \tag{4.4}$$

where $K = f_{z,i}^{CoM}\Delta t^2/(r_{z,i}^{CoM} - r_{z,i}^{OZMP})$ and Δt is a discrete time step. Δr is the deviation of position during Δt. It is straightforward to approximate the desired velocity of the CoM as

$$\dot{r}_x^{CoM} \approx \Delta r_{x,i+1}^{CoM}/\Delta t, \tag{4.5}$$

$$\dot{r}_y^{CoM} \approx \Delta r_{y,i+1}^{CoM}/\Delta t. \tag{4.6}$$

Under such control, the robot can be regarded as an inverted pendulum with its supporting point at the objective ZMP.

4.3.2 Calculating the reference ZMP according to the inverted-pendulum model

As we mentioned above, since the horizontal components of the moment on the ZMP are zero, the mass-concentrated model of the humanoid robot can be regarded as an inverted pendulum. Based on this analogy, we can apply a simple PID controller to control CoM by manipulating the ZMP as described in (Sugihara & Nakamura, 2002).

The dynamics of the mass-concentrated model approximately linearized around an equilibrium point are given as

$$\ddot{r}_x^{CoM} = \omega^2(r_x^{CoM} - r_x^{ZMP}), \tag{4.7}$$

$$\ddot{r}_y^{CoM} = \omega^2(r_y^{CoM} - r_y^{ZMP}), \tag{4.8}$$

where, $\omega = \sqrt{\frac{\ddot{r}_z^{CoM}+g}{r_z^{CoM}-r_z^{ZMP}}}$. The dynamics equations above represent the horizontal movement of the CoM. Due to the symmetry of the x and y components, we are able to focus on the x component in following derivation without loss of generality. By differentiating Eq.(4.7) and ignoring the change in ω by assuming that $\ddot{r}_z^{CoM} = 0$ and r_z^{CoM} is constant, the following equation can be derived.

$$\dddot{r}_x^{CoM} = \omega^2(\dot{r}_x^{CoM} - \dot{r}_x^{ZMP}). \tag{4.9}$$

In order to control the CoM r_x^{CoM} with the reference r_{xref}^{CoM} as a target, we can apply following simple controller as

$$\dot{r}_x^{ZMP} = (K_P + \frac{K_I}{s} + K_D s)(\dot{r}_{xref}^{CoM} - \dot{r}_x^{CoM}), \tag{4.10}$$

$$\dot{r}_{xref}^{CoM} = K_C(r_{xref}^{CoM} - r_x^{CoM}). \tag{4.11}$$

K_P, K_I, K_D and K_C are gains. By the final-value theorem, it may easily be proven that r_x^{CoM} converges to r_{xref}^{CoM} with appropriate settings for the gains.

By integrating the two components presented in this section, the CoM can be controlled by manipulating the ZMP. In the next section, we describe a CoM-Jacobian-based redundancy-resolution technique used to achieve a whole-body movement consistent with the desired CoM movement.

4.3.3 Distributing the CoM movement into joint space

In this section, we present a CoM-Jacobian-based redundancy-resolution technique used to achieve a whole-body movement consistent with the desired CoM movement (Sugihara & Nakamura, 2002). We also present the CoM controller used in our framework which is based on the CoM Jacobian.

Whole body motion generation for balancing

Sugihara *et al.* (Sugihara & Nakamura, 2002) proposed the concept of, and a calculation method for the CoM Jacobian which relates the velocity of the CoM with the angular velocities of all joints as

$$\dot{\boldsymbol{r}}^{CoM} = \boldsymbol{J}_C(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{4.12}$$

where $\boldsymbol{J}_C(\boldsymbol{q}) \in \boldsymbol{R}^{3\times n}$ is the CoM Jacobian, and n is the the number of DoFs in the robot. By using the CoM Jacobian and the weighted pseudo-inverse calculation, we can distribute the CoM velocity to the angular velocities of all the joints according to a sum-squared minimization of all the joint angular velocities as follows:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_C^+ \dot{\boldsymbol{r}}^{CoM} + (\boldsymbol{I} - \boldsymbol{J}_C^+ \boldsymbol{J}_C)\boldsymbol{k}, \tag{4.13}$$

where,

$$\boldsymbol{J}_C^+ = \boldsymbol{W}^{-1}\boldsymbol{J}_C^T(\boldsymbol{J}_C \boldsymbol{W}^{-1}\boldsymbol{J}_C^T)^{-1}, \tag{4.14}$$

$\boldsymbol{W} = diag\{w_i\}(i = 1, \cdots, n)$, and $\boldsymbol{k} \in \boldsymbol{R}^n$ is an arbitrary vector. $I \in \boldsymbol{R}^{n\times n}$ is the identity matrix. The above redundancy-resolution technique with a weighting matrix determines a whole-body motion consistent with the desired CoM movements.

CoM controller in the double support case

We composed the following controller for the CoM assuming both feet are contacting the ground:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+}\dot{\boldsymbol{r}} + (\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\boldsymbol{k}, \tag{4.15}$$

where, $\dot{\boldsymbol{r}} \in \boldsymbol{R}^{6} = [\dot{\boldsymbol{r}}_{C} - \dot{\boldsymbol{r}}_{rl}, \dot{\boldsymbol{r}}_{C} - \dot{\boldsymbol{r}}_{ll}]^{T}$ and $\boldsymbol{J}(\boldsymbol{q}) \in \boldsymbol{R}^{6\times n} = [\boldsymbol{J}_{C}(\boldsymbol{q}) - \boldsymbol{J}_{rl}(\boldsymbol{q}), \boldsymbol{J}_{C}(\boldsymbol{q}) - \boldsymbol{J}_{ll}(\boldsymbol{q})]^{T}$. $\boldsymbol{k} \in \boldsymbol{R}^{6}$ is an arbitrary vector. $\boldsymbol{r}_{C}$ is a position vector of CoM from base-link defined on the waist, and $\boldsymbol{r}_{ll}$ and $\boldsymbol{r}_{rl}$ are position vector of left and right foot from base-link. $\dot{\boldsymbol{r}}$ and $\boldsymbol{J}(\boldsymbol{q})$ are corresponding velocity vector and Jacobian of each $\boldsymbol{r}$ defined above, respectively. The variables are defined in Fig.4.2.

The desired $\dot{\boldsymbol{r}}$ to control the CoM according to the desired trajectory is given by Eqs.(4.5), (4.6) and (4.10).

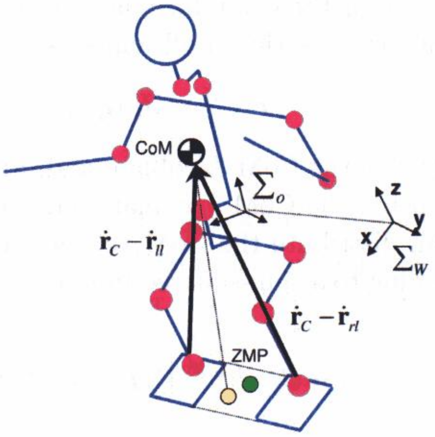

CoM
Σ_o
z
y
x
Σ_W
$\dot{\mathbf{r}}_C - \dot{\mathbf{r}}_{ll}$
$\dot{\mathbf{r}}_C - \dot{\mathbf{r}}_{rl}$
ZMP

Figure 4.2: The definition of the variables.

4.4 Reinforcement learning for CoM movement

In this section, we present a RL method used for the proposed learning framework. Specifically, we use a policy-gradient method for learning CoM motion. The policy-gradient method is a kind of RL method which maximizes the average reward with respect to parameters controlling action rules known as the *policy* (Williams, 1992; Kimura et al., 1997; Baxter & Bartlett, 2001b). Compared with most standard value-function-based RL methods, this type of method has particular features suited to robotic applications. Firstly, the policy-gradient method is applicable to Partially Observable Markov Decision Processes (Aberdeen, 2003). It is almost impossible to consider all possible states of the robot because even if it has a complete set of sensors there will be a degree of noise. It is also possible to consider a partial set of states as input for a RL system. Secondly, the policy-gradient method is a stochastic gradient-descent method. The policy can therefore be improved upon every update. In this section, we briefly describe a framework for RL with the policy-gradient method.

4.4.1 Reinforcement Learning with a policy-gradient method

Assuming a Markov Decision Process, the average reward, discounted cumulative reward and value functions are defined as

$$\eta(\boldsymbol{\theta}) = \lim_{T\to\infty} \mathbf{E}\left[\frac{1}{T}\sum_{t=0}^{T} r(\mathbf{x}_t)\right], \tag{4.16}$$

$$\eta_\beta(\boldsymbol{\theta}) = \lim_{T\to\infty} \mathbf{E}\left[\frac{1}{T}\sum_{t=0}^{T} \beta^t r(\mathbf{x}_t)\right], \tag{4.17}$$

$$V_\beta^\pi(\mathbf{x}) = \mathbf{E}\left[\sum_{k=0}^{\infty} \beta^k r_{t+k+1} \middle| \mathbf{x}_t = \mathbf{x}\right], \tag{4.18}$$

$$Q_\beta^\pi(\mathbf{x}, a) = \mathbf{E}\left[\sum_{k=0}^{\infty} \beta^k r_{t+k+1} \middle| \mathbf{x}_t = \mathbf{x}, a_t = a\right]. \tag{4.19}$$

$\eta(\boldsymbol{\theta})$ is the average reward and $\eta_\beta(\boldsymbol{\theta})$ is the discounted cumulative reward.

$V^{\pi}_{\beta}(\mathbf{x})$ and $Q^{\pi}_{\beta}(\mathbf{x}, a)$ are state-value function and action-value function, respectively (Sutton & Barto, 1998). $\mathbf{x}$ is the state, a is the action and $\boldsymbol{\theta}$ is the parameters of the stochastic policy. β is a discounting factor. The goal of RL here is to maximize the average reward.

If we can calculate the gradient of $\eta(\boldsymbol{\theta})$ with respect to policy parameters $\boldsymbol{\theta}$, it is possible to search for a (sub-)optimal policy in policy-parameter space by updating the parameters to $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}+\alpha\nabla\eta(\boldsymbol{\theta})$. $\nabla\eta(\boldsymbol{\theta})$ is the gradient of $\eta(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. Various derivations and algorithms have been proposed in order to estimate the gradient based on sampling, though interaction with the environment. According to Kimura and Kobayashi (Kimura & Kobayashi, 1998), the gradient is given by

$$\begin{aligned}
\nabla\eta &= (1-\beta)\nabla\eta_{\beta} && (4.20)\\
&= (1-\beta)\int d(\mathbf{x})\int \pi(a,\mathbf{x})\Big[\nabla \log d(\mathbf{x}) \\
&\quad +\frac{1}{1-\beta}\nabla \log \pi(a,\mathbf{x})\Big] Q^{\pi}_{\beta}(\mathbf{x}, a) da d\mathbf{x} && (4.21)\\
&= \int d(\mathbf{x})\int \pi(a,\mathbf{x})\Big[\,(1-\beta)\,\nabla \log d(\mathbf{x}) \\
&\quad +\nabla \log \pi(a,\mathbf{x})\Big]\,\left\{Q^{\pi}_{\beta}(\mathbf{x}, a) - V^{\pi}_{\beta}(\mathbf{x})\right\} da d\mathbf{x} && (4.22)\\
&= \lim_{T\to\infty,\beta\to 1}\frac{1}{T}\sum_{t=0}^{T}\nabla \log \pi(a_t,\mathbf{x}_t)\sum_{s=t}^{T}\beta^{s-t}\delta(\mathbf{x}_s, a_s)\\
&= \lim_{T\to\infty,\beta\to 1}\frac{1}{T}\sum_{t=0}^{T}\delta(\mathbf{x}_t, a_t)\sum_{s=0}^{t}\beta^{t-s}\nabla \log \pi(a_s,\mathbf{x}_s).\\
& && (4.23)
\end{aligned}$$

Where, $\pi(\mathbf{x}, a; \boldsymbol{\theta}) = P(a|\mathbf{x}; \boldsymbol{\theta})$ is the stochastic policy, which maps a state $\mathbf{x}$ to an action a stochastically. $\nabla\pi(\mathbf{x}, a; \boldsymbol{\theta})$ means the deviation of $\pi(\mathbf{x}, a; \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. $d(\mathbf{x})$ is the stationary distribution of $\mathbf{x}$. $\delta(\mathbf{x}, a)$ is TD-error defined as $\delta(\mathbf{x}_t, a_t) = r(\mathbf{x}_t)+\beta\int p(\mathbf{x}_{t+1}|\mathbf{x}_t, a_t)V^{\pi}_{\beta}(\mathbf{x}_{t+1})d\mathbf{x}_t - V^{\pi}_{\beta}(\mathbf{x}_t)$. Equation (4.20) is presented in (Baxter & Bartlett, 1999) as Theorem.1, and Eq.(4.21) is derived in (Sutton et al., 2000). The derivation of Eq.(4.22) is based on $\int \nabla\pi(\mathbf{x}, a)V^{\pi}_{\beta}(\mathbf{x})da = 0$. If we neglect $V^{\pi}_{\beta}(\mathbf{x})$, the algorithm is exactly same as the GPOMDP algorithm developed in (Baxter & Bartlett, 2001b). As pointed out in (Baxter & Bartlett, 2001b), the discounting factor β controls a bias-variance trade-off in the policy-gradient estimated by sampling.

In fact, we update the policy parameters according to the following rule: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha D_t \delta\left(\mathbf{x}_t, a_t\right)$, where, D is updated by $D_t = \beta D_{t-1} + \nabla \log \pi(\mathbf{x}_t, a_t)$. However, to gain TD-error $\delta\left(\mathbf{x}_t, a_t\right)$, we need the state-value function $V_\beta^\pi(\mathbf{x})$. In this chapter, we simultaneously approximate it by using function approximator $\hat{V}_\beta^\pi(\mathbf{x}; \mathbf{w})$ and a simple TD-learning method presented as $\mathbf{w} = \mathbf{w} + \alpha \delta_t \frac{\partial \hat{V}_\beta^\pi(\mathbf{x};\mathbf{w})}{\partial \mathbf{w}}$. TD-error $\delta(\mathbf{x}_t, a_t)$ is then approximately calculated by $\delta(\mathbf{x}_t, a_t) = r(x_t) + \beta \hat{V}_\beta^\pi(\mathbf{x}_{t+1}) - \hat{V}_\beta^\pi(\mathbf{x}_t)$. Note that β should be satisfy $0 \leq \beta < 1$ in order to prevent the state-value function from diverging.

4.5 Application to learning of a dynamic task: ball-punching

In order to demonstrate the effectiveness of our proposed learning framework for learning whole-body movements with a humanoid robot, we applied it to the learning of a dynamic ball-punching motion. The goal is to make the ball-punching stronger while the learning process focused on the CoM motion. This section describes the implementation of a punching motion and the learning process.

Punching motion projected onto the null-space of the CoM controller

A punching motion was straightforwardly implemented by tracking a target trajectory in task space. In this study, we achieve the tracking control in the null-space of the CoM controller by introducing the following vector as the arbitrary vector in Eq.(4.15):

$$\boldsymbol{k} = \tilde{\boldsymbol{J}}_{ra}^{+}(\dot{\boldsymbol{r}}_{ra} - \boldsymbol{J}_{ra}\boldsymbol{J}^{+}\dot{\boldsymbol{r}}), \tag{4.24}$$

where, $\boldsymbol{J}_{ra} \in \boldsymbol{R}^{3 \times n}$ is the Jacobian relating the right hand velocity in task space $\dot{\boldsymbol{r}}_{ra}$ with $\dot{\boldsymbol{q}}$ as $\dot{\boldsymbol{r}}_{ra} = \boldsymbol{J}_{ra}\dot{\boldsymbol{q}}$, and $\tilde{\boldsymbol{J}}_{ra}^{+} = \boldsymbol{J}_{ra}(\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})$. Introducing this vector yields target tracking with the right hand in the null-space of the CoM controller (Yoshikawa, 1990).

The Gaussian policy and function approximator for the state-value function

We implemented the following Gaussian policy as a stochastic policy for controlling the CoM.

$$\pi(\mathbf{x}, a; \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(a - \mu(\mathbf{x}; \boldsymbol{\theta}))^2}{2\sigma^2}\right), \tag{4.25}$$

where $\mu(\mathbf{x}; \boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x})$. $\mathbf{x}$ is the state, a is the action. We located the Gaussian basis functions $\boldsymbol{\phi}(\mathbf{x})$ on a grid with even intervals in each dimension of the observation space as in (Doya, 2000; Matsubara et al., 2005). The function approximator for the state-value function is also modeled as $\hat{V}_\beta^\pi(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$

The reward function

The purpose of the ball-punching task is to make the punching as strong as possible. We designed the reward function according to this objective as

$$r = (t - 2.5)\sqrt{\boldsymbol{v}_b^T \boldsymbol{v}_b} \tag{4.26}$$

because the velocity of the ball $\boldsymbol{v}_b$ hit by punching is proportional to the momentum of the ball. The term associated with time t is incorporated in the reward function in order to avoid local-minima motions which involve the robot falling forwards and disregard the timing of the punch. The value 2.5 is a bias to achieve distribution of the reward to positive and negative. The negative reward -5 is given when the both feet leave the ground in order to avoid acquiring a punching motion with jumping.

4.5.1 Simulation settings and results

We applied the proposed learning framework to the acquisition of a strong punching movement on a Fujitsu HOAP2 humanoid robot (see Fig.4.4) in numerical simulation. In this study, we focus on controlling x-axis component of the CoM, *i.e.*, the output of the policy is the target velocity of the x-axis component of CoM $\dot{r}_{xref}^{CoM}$. More precisely, the desired CoM is constrained one-dimensional movement. The output of the policy is distributed to x and y-axis component of CoM as $\dot{r}_x^{CoM} = \sin(\phi) \cdot \dot{r}_{xref}^{CoM}$ and $\dot{r}_y^{CoM} = \cos(\phi) \cdot \dot{r}_{xref}^{CoM}$), where ϕ is the angle from y-axis to x-axis in clockwise direction and $\phi = \pi/3$ as depicted in Fig.4.3. This setting can be considered suitable to sufficiently use the area of support polygon because a diagonal line is larger than x and y-axis line. The state-space was simply defined as $\mathbf{x} = \left(r_x^{CoM}, t\right)$. We allocated 100 (= 10 × 10) basis functions $\boldsymbol{\phi}(\mathbf{x})$ in the state-space $(-1.0 < r_x^{CoM} < 0.0, 0.5 < t < 4.0)$ to represent the mean of the policy $\mu(\mathbf{x})$. The ball was modeled as a simple point mass (0.1kg) and the contact between the robot and the ball is simulated using a spring-damper model. A spring-damper model was also used to model the floor contact. The integration time-step for the robot was 0.2ms, and the time interval for learning was 50ms.

For the CoM and right-arm controllers, it is required to set the weighting matrix suitable for this task to appropriately achieve a whole-body motion. In order to avoid using the DoFs in the right arm (which are used for the punching motion) for the CoM controller, the weights in right arm are set to smaller (0.01) than other joints (1.0) in the CoM controller described in Eq.(4.13). For the right-arm controller described in Eq.(4.24), to achieve a punching motion mainly using the right arm, we set the weights in body joint larger (3.0) than other joints (1.0). The target trajectory for the right-arm controller in order to achieve a punching motion was designed as $r_{ra_{xref}} = p \sin\left(2\pi f\left(t - t_a\right)\right) + q \ (t \geq t_a)$, and we set the parameters so that the amplitude $p = -0.03$m, the bias $q = 0.21$m, the frequency $f = 1.5$ Hz and the bias $t_a = 3.5$ by considering the HOAP2's physical model. During $0 < t < 3.5$, $r_{ra_{xref}}$ is the constant p.

Figure 4.5 shows the reward at each episode according to the policy-gradient method. The curve means that the (sub-)optimal punching motion with maximal reward was acquired in around 3000 episodes. Figure 4.6 is an acquired policy for controlling x-axis component of CoM, and Figure 4.7 presents a whole-body punching motion acquired with the control policy.

The punching motion with keeping the CoM at the initial point yielded the ball momentum about 0.037 kgm/sec. The acquired punching motion without any probabilistic factors made the ball momentum about 0.085 kgm/sec on average (the standard deviation was 0.005) , which means the ball momentum generated by the learned policy was about 2.3 times greater than initial performance.

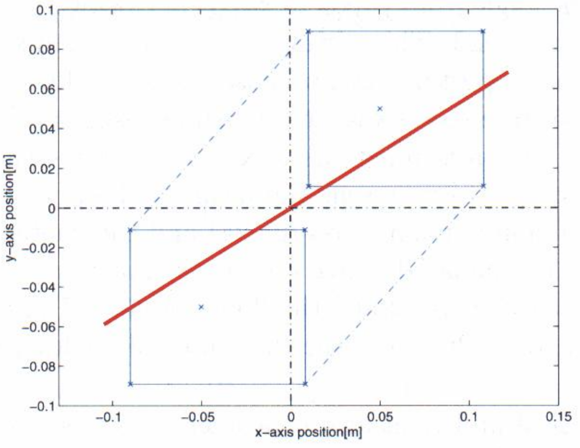


Figure 4.3: The one-dimensional CoM movement controlled by the policy is the red line. The blue line is each foot, and dash line means the support polygon.

4.6 Experiments on real hardware system

In this section, we implemented the proposed controller on a real humanoid robot, HOAP2. We then implemented acquired control policy for CoM to the robot as well as the previous chapter's result. However, here we implemented the CoM trajectories generated successful punching trials in simulation with a leaned policy, because the policy could not achieve the desired punching movements in real environment due to the serious modeling error between the simulator and the real robot. Figures 4.8 and 4.9 are the initial motion and the acquired punching motion in the real environment, respectively. Figure 4.10 is a CoM trajectories with the acquired CoM motion. These results suggest that the cooperative whole-body punching movement with learned control policy in simulations is achieved in real hardware system. In order to visualize the effectiveness of learned punching motions, we set a toy car in front of the robot to be hit by the punch. The distance of the hit car can measure the effectiveness of initial and learned punch, respectively. Figures 4.11 and 4.12 are sequential snapshots until hitting to the car and after hitting. The upper and lower sequence are initial and learned movements, respectively. The results suggests that the punching motion, *i.e.*, the acquired

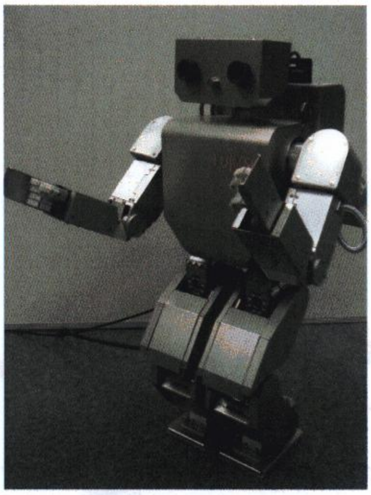


Figure 4.4: Fujitsu humanoid robot HOAP2. 6DoF for the legs, 4DoF for the arms and 1DoF for the waist. The total weight is about 7kg, and the height is about 0.4m.

cooperative whole-body movement, is effective even in real environment.

4.7 Understanding the proposed method from motor learning point of view

In the proposed method, learning was focused on the CoM movements with several constraints. The list of the constraints are

1. The action space for learning consists of the velocity of the CoM.
2. Joint coordination are introduced by weighted pseudo inverse from CoM movement to joint space.
3. The ground reaction force always penetrates CoM.
4. The ZMP is restricted inside of the support polygon, which keeps the robot contacting with the ground.

In the ball-punching task, the above setting makes the complex motor learning task on a full-body humanoid robot feasible and tractable. As a result, we achieved learning within only a few thousand trials. Although

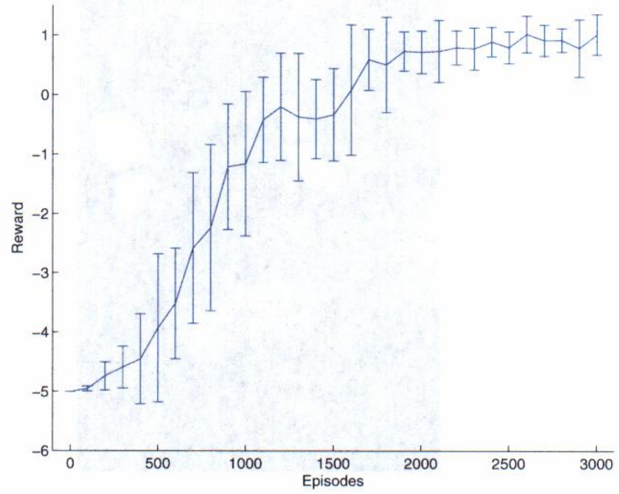


Figure 4.5: The acquired reward at each episodes. The learning curve was averaged by 5 experiments and smoothed out by taking a 50-moving average.

these constraints makes learning task more feasible and tractable, while it bounds the performance of the resulting movements. It is of course in trade-off relationship.

4.8 Summary and discussion

This chapter presented an approach for acquiring dynamic whole-body movements on humanoid robots focused on learning a control policy for the CoM. This approach allows us to execute learning in low-dimensional feature space, which is composed of CoM position and executing time variables, in acquiring a desired coordinated full-body movement. We applied the framework to the learning of a dynamic ball-punching motion on a HOAP2 model in numerical simulations. As a result, we demonstrated that it is possible to efficiently acquire dynamic punching motions through our learning approach. We then implemented the acquired CoM motion to a real robot and demonstrated the effectiveness of the acquired policy in the real environment.

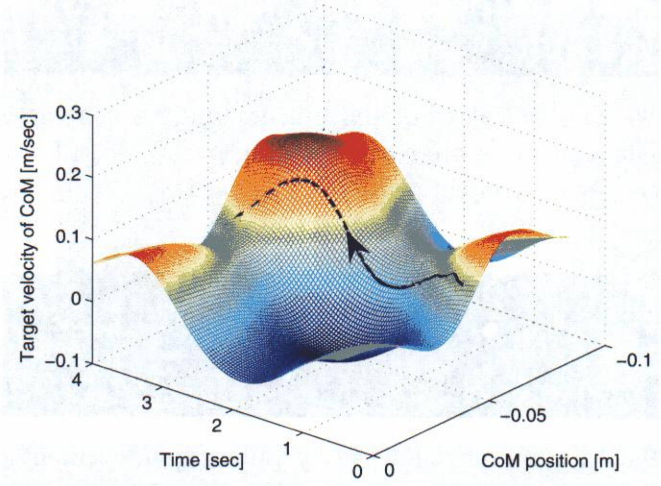


Figure 4.6: An acquired control policy for axis component of the CoM. The dash-line is the trajectory corresponding to the punching motion in Fig.4.7

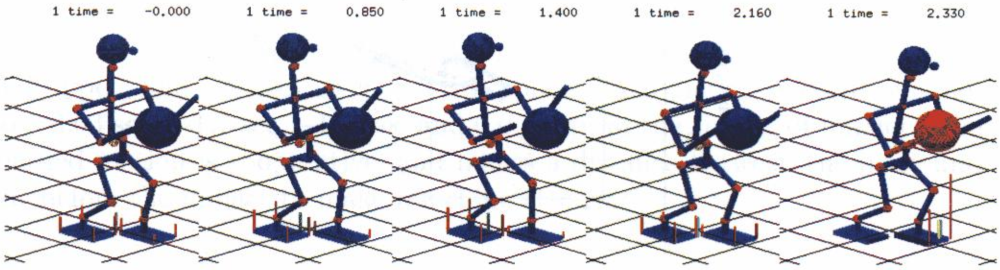


Figure 4.7: An acquired whole-body punching movement. The snapshots are corresponding in 1.832 sec, 2.868 sec, 3.492 sec, 3.705 sec and 3.858 sec, respectively. The red bar on the foot of the robot means the ground reaction force.

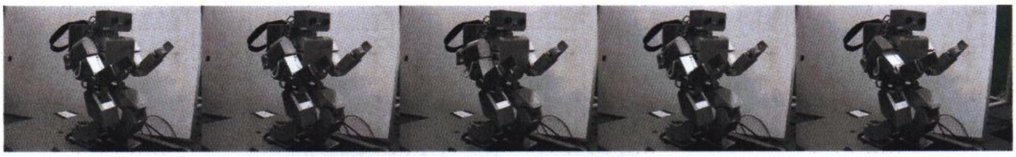


Figure 4.8: The initial whole-body punching movement on real hardware system. The snapshots are corresponding in 0.0 sec, 1.0 sec, 1.6 sec, 1.8 sec and 2.0 sec, respectively.

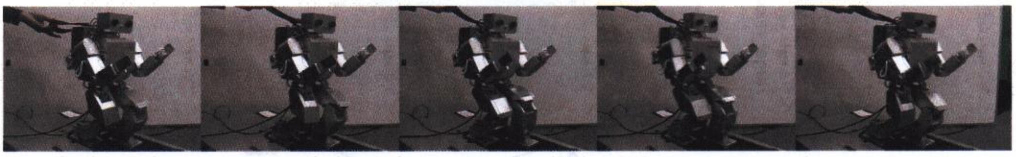


Figure 4.9: An acquired whole-body punching movement on real hardware system. The snapshots are corresponding in 0.0 sec, 1.0 sec, 1.5 sec, 1.8 sec and 2.0 sec, respectively.

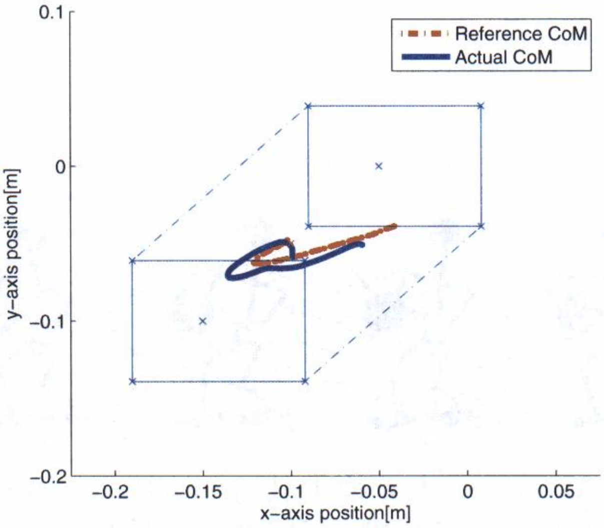
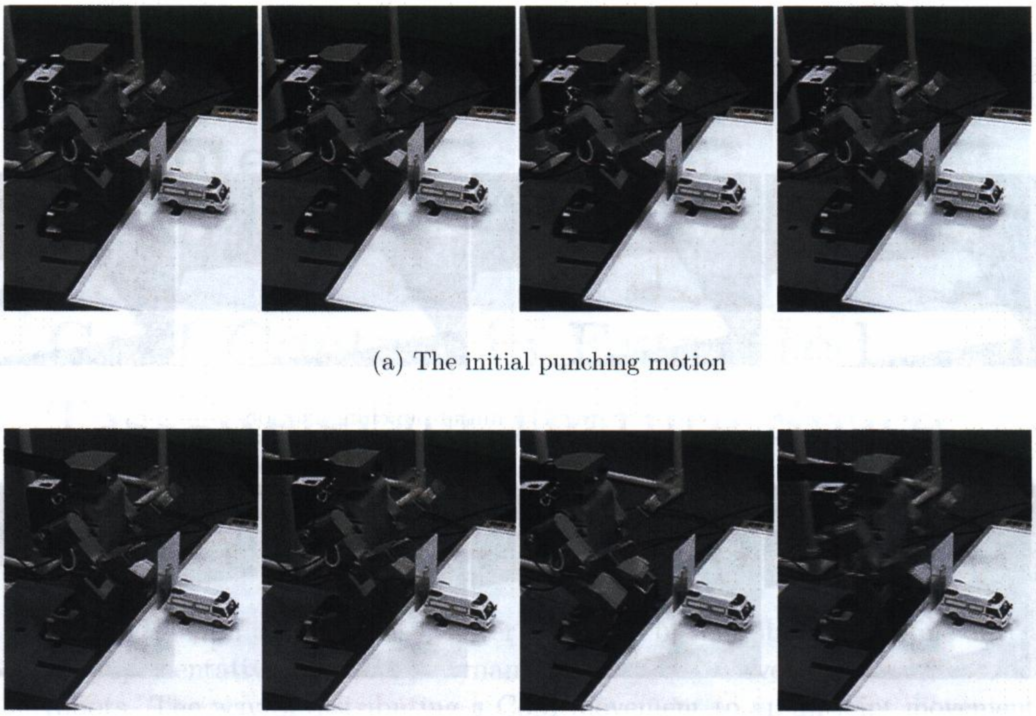


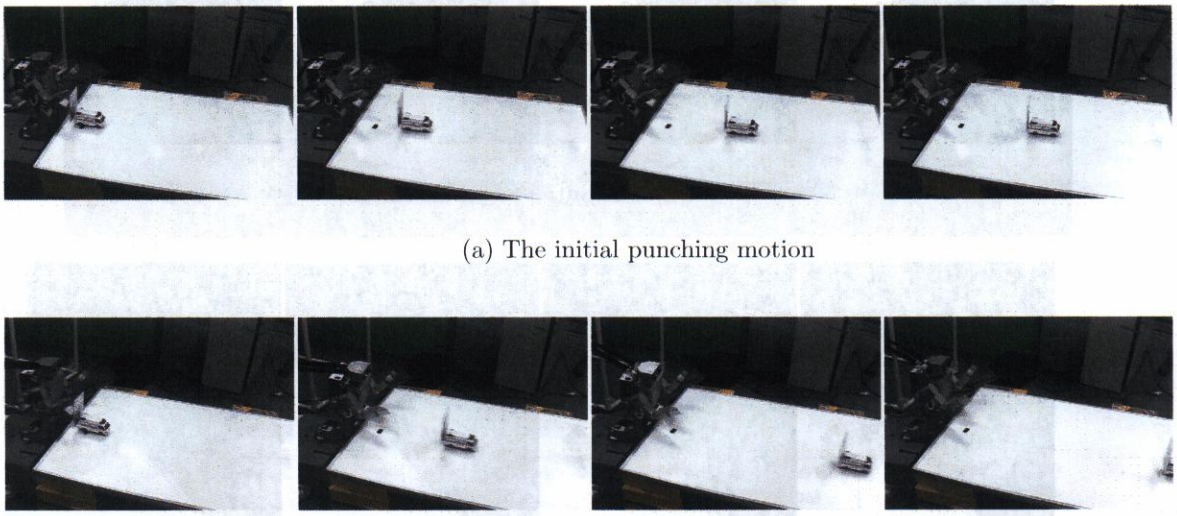
Figure 4.10: The trajectories of the reference and actual CoM during punching motion with the acquired control policy.



(a) The initial punching motion

(b) The learned punching motion

Figure 4.11: The sequential snapshots for punching motion with initial(upper) and learned(lower) control policy. Each picture is corresponding to -3.03 sec, -1.7 sec, -0.7 sec and 0.0 sec from the timing of impact. The learned motion is cooperative, and makes a distance between the body and car before the impact to make punching stronger.



(a) The initial punching motion

(b) The learned punching motion

Figure 4.12: The sequential snapshots for punching motion with initial(upper) and learned(lower) control policy. Each picture is corresponding to 0.0 sec, 0.67 sec, 1.67 sec and 2.0 sec from the timing of impact. From the movement of the car hit by punching, it is clear that the learned punching was significantly effective in term of the impact on the car. The speed of the car hit by initial and learned punching were 0.42 and 0.71 m/sec.

Chapter 5

CoM Control in Extracted Task-Relevant Feature Space

As described in the previous chapter, the CoM of the robot can be one of the key representative features for dynamic whole-body movements on humanoid robots. The way of distributing a CoM movement to an all-joint movement has multiple solutions, which is called a "redundancy" problem (Yoshikawa, 1990). Although several criteria have been utilized to yield uniqueness in resolving redundancy, such as the technique to minimum norm of all-joint velocity used in the previous chapter (see Section 4.3). In this chapter, we investigate human-like redundancy resolution for CoM movements via the utilization of human demonstrations. In our proposed method, we use a mapping from a reduced dimensional feature space to a task space. The reduced dimensional feature space is extracted from human demonstrations through a statistical method. Note that the "feature space" is, as captured, defined a low-dimensional space to achieve the task (as in task space control for an end-effector) such as the CoM of the robot, not as in the learning process discussed in previous chapters. We apply our proposed method to control CoM in order to extract low dimensional features space of a small humanoid robot model. Simulation results demonstrate the effectiveness of our proposed method for human-like redundancy resolution.

5.1 Introduction

A humanoid robot has a large number of DoFs in general. The task space control is one of most common frameworks to achieve a desired motion with such complex robots (Yoshikawa, 1990; Nakanishi, Cory, Mistry, Peters, & Schaal, 2005; Sentis & Khatib, 2005). One problem is that these systems may suffer from the redundancy problem, arised ill-posedness of dimension of task in operational space to joint space. In order to yield the uniqueness in redundancy resolutions, several criteria have been utilized, for example, the minimum norm of all-joint velocity. Since humanoid robots have a similar physical structure to humans, it is expected that humanoid robots will help us with many tasks in our normal living spaces without any additional environment-specific equipment. In such a case, it would be desirable for several reasons that the task space control could be achieved by taking a human-like solution in joint space among infinite number of candidates. Such reasons being, first, humanoids tend to be designed by considering human's configuration in joint limits and power distributions, as well as kinematics. Secondly, it is easy to predict humanoid's movements for humans, which is helpful and safe when they work together.

In the research field of physiology, there has been proposed several hypotheses to explain human motor control and learning. Here, we focus on the idea of muscle (joint) synergies (Bernstein et al., 1996; d'Avella et al., 2003; Tresch et al., 2006) in order to address a human-like redundancy resolution. According to this, for adult humans, the efficient motor learning skill can be explained so that several coordinations or synergies have been already explored and memorized through their experiences, and are appropriately utilized in order to achieve efficient learning even for a novel motor task. With this in mind, the redundancy problem can be much reduced to human-like solution which is naturally picked up among infinite number of candidate solutions.

In the computer graphics community, similar ideas have already been put into practice to reduce the search space much smaller, allowing efficient dynamics calculations. To extract such key basis, conventional studies use statistical methods for human demonstrations. Such basis is very similar to synergies as proposed in physiology.

In this chapter, by taking these ideas, we propose a novel task space control method by selecting human-like joint movements among infinite number of candidates. More precisely, we use data observed from human demonstrations to extract low dimensional basis for the target whole body movement, to perform the CoM control in a task-dependent low-dimensional feature space.

The organization of this chapter is as follows. In Section 5.2, we describe how we extract low-dimensional feature space via human's demonstration. The method to achieve the task space control from extracted low-dimensional features space is then proposed. We also suggest an approach to select task-dependent features based on its manipulability in this section. In Section 5.3, we consider the application of the proposed method for achieving a squatting motion on a whole-body humanoid robot in numerical simulations. Simulation results and discussion are described in Section 5.4 and 5.5, respectively.

5.2 Whole-body control in low-dimensional feature space

In this section, for simplicity, we only focus on the feature extraction with a linear model and a task space control method for end-effectors operating in the extracted low-dimensional feature space. Now, we assume that an all-joint trajectory can be represented by the following linear equation:

$$\mathbf{Q}_h \approx \mathbf{W}\mathbf{Y}_h, \tag{5.1}$$

where, $\mathbf{Q}_h = \begin{bmatrix}\mathbf{q}_h(1) & \mathbf{q}_h(2) & \cdots & \mathbf{q}_h(T)\end{bmatrix}$ is the observed all-joint data, $\mathbf{W} = \begin{bmatrix}\mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_M\end{bmatrix}$ is basis matrix and $\mathbf{Y}_h = \begin{bmatrix}\mathbf{y}_h(1) & \mathbf{y}_h(2) & \cdots & \mathbf{y}_h(T)\end{bmatrix}$ is the feature coefficient data, respectively, where, $\mathbf{q}, \mathbf{w} \in R^N$ and $\mathbf{y} \in R^M$. M is the dimension of the feature space. If we set $M < N$ and search $\mathbf{W}$ and $\mathbf{Y}_h$ simultaneously for minimizing an error function $\|\mathbf{Q}_h - \mathbf{W}\mathbf{Y}_h\|$, a dimensionality reduction can be achieved with a simple linear model. However, to select one solution from among the infinite number of combinations between $\mathbf{W}$ and $\mathbf{Y}_h$ requires some constraints. Principal Component Analysis (PCA) is one of most effective methods for achieving such dimensionality reduction (Jolliffe, 1986). This method is derived by introducing a constraint such that each $\mathbf{w}$ is orthonormal. In the neuroscience field, Non-negative Matrix Factorization (NMF) is often used to extract muscle synergies (Lee & Seung, 2000). The constraint in this method is simply that both $\mathbf{W}$ and

$\mathbf{Y}_h$ must be positive. One of the interesting properties is that the resulting basis tends to be sparse compared with other methods like PCA (Lee & Seung, 1999; d'Avella et al., 2003; Tresch et al., 2006). A constraint that $\mathbf{y}_h$ is sampled from a non-Gaussian distribution and each $\mathbf{y}_h$ is stochastically independent derives as, Independent Component Analysis (ICA) (Hyvarinen, Karhunen, & Oja, 2001) and it is often used with PCA (Tresch et al., 2006). Although all methods are based on a simple linear model, the characteristic of each method can be significantly different. By taking a rather simplified approach, for our purpose we focused on a PCA approach in this chapter.

5.2.1 Feature extraction by PCA

The objective function for PCA with a linear model can be represented as,

$$E(\mathbf{W})_{PCA} = \frac{1}{2T}\sum_{j=1}^{T}\left\|\mathbf{q}(j) - \sum_{i=1}^{M}(\mathbf{w}_i^T\mathbf{q}(j))\mathbf{w}_i\right\| \tag{5.2}$$

$$= \mathbf{tr}(\mathbf{C}_q) - \sum_{i=1}^{M}\mathbf{w}_i^T\mathbf{C}_q\mathbf{w}_i, \tag{5.3}$$

where, $\mathbf{C}_q = \frac{1}{T}\sum_{i=1}^{T}\mathbf{q}(i)\mathbf{q}(i)^T$, and the solution for PCA resulting from minimization of the objective function with respect to $\mathbf{w}$ is an eigenvector of $\mathbf{C}_q$ in the order of large eigenvalues $M(\leq N)$ (Hyvarinen et al., 2001). Same solutions can be derived from maximization of the variance of $\mathbf{y}_h$.

5.2.2 Task space control in the feature space

It is defined that $\mathbf{x}$ is the position of an arbitrary point in task space, $\mathbf{q}$ is a joint angle vector and the relationship between that is given as $\mathbf{x} = f(\mathbf{q})$. The derivative of the equation is given as follows:

$$\dot{\mathbf{x}} = \frac{\partial \mathbf{x}}{\partial \mathbf{q}}\frac{d\mathbf{q}}{dt} = J(\mathbf{q})\dot{\mathbf{q}}, \tag{5.4}$$

where, $J(\mathbf{q})$ is called Jacobian. We can calculate the desired angler velocity $\dot{\mathbf{q}}_d$ to achieve target velocity $\dot{\mathbf{x}}_d$ is given as

$$\dot{\mathbf{q}}_d = J(\mathbf{q})^{+}\dot{\mathbf{x}}_d + (\mathbf{I} - \mathbf{J}(\mathbf{q})^{+}\mathbf{J}(\mathbf{q}))\boldsymbol{\xi}, \tag{5.5}$$

where, $\boldsymbol{\xi}$ is an arbitrary vector and the mapping $\mathbf{J}(\mathbf{q})^{+} = \mathbf{J}(\mathbf{q})^{T}(\mathbf{J}(\mathbf{q})\mathbf{J}(\mathbf{q})^{T})^{-1}$ and $(\mathbf{I}-\mathbf{J}(\mathbf{q})^{+}\mathbf{J}(\mathbf{q}))$ are the pseudo inverse resolution and the null space projection, respectively (Yoshikawa, 1990).

Now, we consider to approximately represent a joint vector in low-dimensional feature space presented in Eq.(5.1). By assuming that basis matrix $\mathbf{W}$ for feature space is constant in time, the following equation can be obtained as $\dot{\mathbf{q}}_h \approx \mathbf{W}\dot{\mathbf{y}}_h$(if $N = M$, $\mathbf{q}_h = \mathbf{W}\mathbf{y}_h$, $\dot{\mathbf{q}}_h = \mathbf{W}\dot{\mathbf{y}}_h$).

By substituting the above in Eq.(5.4), the velocity $\dot{\mathbf{x}}$ is given as,

$$\dot{\mathbf{x}} = J(\mathbf{q})\dot{\mathbf{q}} \approx J(\mathbf{q})\mathbf{W}\dot{\mathbf{y}} = J_f(\mathbf{q})\dot{\mathbf{y}}, \tag{5.6}$$

where, $J_f(\mathbf{q}) = J(\mathbf{q})\mathbf{W}$ is named *feature-Jacobian* which relates $\dot{\mathbf{x}}$ to $\dot{\mathbf{y}}$. By using this, we can calculate the desired angler velocity $\dot{\mathbf{q}}_d$ to achieve $\dot{\mathbf{x}}_d$ as

$$\dot{\mathbf{q}}_d = \mathbf{W}\dot{\mathbf{y}}_d, \tag{5.7}$$

$$\dot{\mathbf{y}}_d = J_f(\mathbf{q})^{+}\dot{\mathbf{x}}_d + (\mathbf{I} - \mathbf{J}_f(\mathbf{q})^{+}\mathbf{J}_f(\mathbf{q}))\boldsymbol{\xi}_f. \tag{5.8}$$

Note that the dimension of $J_f(\mathbf{q})$ can be lower than $J(\mathbf{q})$. It means that the advantages of using this Jacobian are not only that the redundancy is reduced based on the observations, but also that calculation cost for each desired angler velocity $\dot{\mathbf{q}}_d$ is reduced. Moreover, it can be expected that human-like resolution is appropriately selected from among multiple solutions in joint space.

As already mentioned above, the above equations are precise if and only if $N = M$. For the cases of $M < N$, the accuracy of the manipulation by using the above equation is concerned with the corresponding eigenvalues of basis consisting feature space. Moreover, if the operating movement is greatly different from the movements captured from human, the accuracy can be worse. Thus, the manipulation by using Eqs.(5.7) and (5.8) is appropriate if the large principal components are used and performing a similar movement to captured data.

5.2.3 Evaluation of feature space by manipulability

While the humanoid robot has a generalized structure for achieving several tasks, however, the task space control with a feature-Jacobian presented in

Eqs.(5.7) and (5.8) could have a directivity depending on the characteristic of the feature space. In order to achieve the goal in this chapter, which is to reduce the redundancy in the task space control, we need a criterion for evaluating the extracted feature space in terms of the capability to execute the desired task space control.

The concept of the manipulability (Yoshikawa, 1990) seems to be suitable for this requirement. We consider to utilize the criterion to evaluate feature space for achieving the desired task space control. For Eq.(5.6), we consider the set of all end-effector velocities $\mathbf{v} = \dot{\mathbf{x}}$ realizable by constrained feature velocities such that the $||\dot{\mathbf{y}}|| \leq 1$. The set can be represented by the manipulability ellipsoid as

$$\mathbf{v}^T(\mathbf{J}_f^+)^T\mathbf{J}_f^+\mathbf{v} \leq 1, \tag{5.9}$$

where, $\mathbf{J}_f^+$ is the pseudo-inverse matrix of $\mathbf{J}_f$. By using the singular-value decomposition, we can write the above equation in a different form as

$$\mathbf{v}^T(\mathbf{J}_f^+)^T\mathbf{J}_f^+\mathbf{v} = \sum_{\sigma_i \neq 0} \frac{1}{\sigma_i^2}\tilde{v}_i^2 \leq 1, \tag{5.10}$$

where, $\tilde{\mathbf{v}} = U^T\mathbf{v}$, $\mathbf{U}\Sigma\mathbf{V}^T(= \mathbf{J}_f)$ is the singular-value decomposition of $\mathbf{J}_f$, and where, $\mathbf{U}$ and $\mathbf{V}$ are, respectively, $m \times m$ and $n \times n$ orthogonal matrices. Σ is an $m \times n$ matrix in which each diagonal element is called singular value of $\mathbf{J}_f$. Eq.(5.10) presents the manipulability ellipsoid, in which the length and the direction of each axis mean the capability of manipulating $\mathbf{x}$ in task space by using the feature Jacobian.

Note that the different features $\mathbf{W}$ used in $\mathbf{J}_f$ derive completely different manipulability ellipsoids. Hence, in order to achieve the desired movement in task space with reduced dimensional feature space, to analyze the characteristic of the ellipsoid by the manner presented above can be useful. In the later section, we use this criterion with the reconstruction error of PCA for selecting low dimensional and appropriate characteristic feature space for desired task space control.

5.3 Achievement of the squatting motion on a humanoid robot

In order to demonstrate the effectiveness of the proposed method, we apply it to a squatting motion on a Fujitsu HOAP2 humanoid robot model (in Fig.5.1) in numerical simulations.

5.3.1 Feature extraction of squatting motion from human demonstration

First of all, we extract feature space of the squatting motion from captured data through human demonstration. The data is captured by using a motion capture system. The markers are attached on each joint of the whole-body. By using HOAP2's kinematics model, the data in Cartesian is converted to joint angle data by taking the differences in the ratio of limbs between human and HOAP2 into account along with the method presented in (Riley, Ude, Wade, & Atkeson, 2003). Next, PCA is applied to the converted joint angle data in order to extract a feature space for human's squatting motion. The result is presented in Fig.5.2 as well as other movements. All of the movements captured have less than 10 dimensions to approximately explain the captured data. This is supported previous works by (Safonova et al., 2004).

5.3.2 Achievement of the squatting motion in feature space

We consider that the squatting motion can be interpreted in the CoM movement as just vertical movements, and horizontal movements are used to keep the ZMP within the supporting polygon (same as described in Section 4.3). In the double support phase, there is a redundancy problem, which means that it has multiple solutions in the configuration of the support polygon. To avoid such a problem, we additionally introduce orientation control of the feet as a task within task space control. Thus, we define the control variable $\mathbf{x}$ in task space as $\mathbf{x} = \begin{bmatrix}\mathbf{r}_{rCoM}^T & \mathbf{r}_{lCoM}^T & \phi_r^T & \phi_l^T\end{bmatrix}^T$, where, $\mathbf{r}_{r(l)CoM} \in \mathbf{R}^3$ is the CoM position from the tip of right (left) leg and $\phi_{r(l)} \in \mathbf{R}^3$ is Euler angles for right (left) foot. The velocity vector

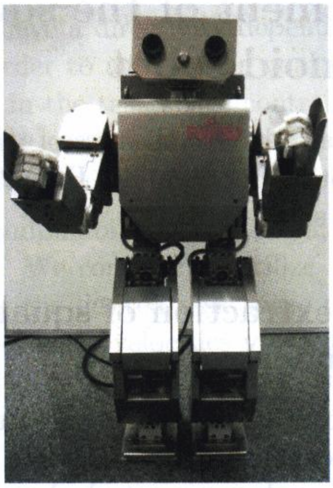


Figure 5.1: HOAP2.

$\dot{\mathbf{x}}$ corresponding to $\mathbf{x}$ has a relationship to $\dot{\mathbf{q}}$ as where $\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$, where $\mathbf{J}(\mathbf{q}) = \left[\mathbf{J}_{CoM}^T - \mathbf{J}_{rleg}^T \;\; \mathbf{J}_{CoM}^T - \mathbf{J}_{lleg}^T \;\; \mathbf{J}_{reuler}^T \;\; \mathbf{J}_{leuler}^T\right]^T$. If we determine the low-dimensional features $\mathbf{W}$, the target feature velocity $\dot{\mathbf{y}}_d$ is calculated by using equations with $\mathbf{J}(\mathbf{q})$ and $\mathbf{W}$ in order to achieve the target velocity $\dot{\mathbf{x}}_d = \mathbf{K}(\mathbf{x}_o - \mathbf{x})$, where, $\mathbf{x}_o$ is the target position of $\mathbf{x}$.

5.3.3 Manipulability analysis for feature spaces in the squatting motion

We analyze the manipulability ellipsoid for each low-dimensional feature space extracted by PCA presented in previous subsection to achieve the squatting motion. In order to investigate the ability to execute given tasks, in particular, we calculate the possible velocity in terms of given tasks with norm constraint of the joint velocity as

$$\dot{\mathbf{x}} = \mathbf{v} = U\tilde{\mathbf{v}}. \tag{5.11}$$

Each variables in $\mathbf{x}$ is plotted in Fig.5.5 corresponding to several feature spaces. The result suggests that extremely low dimensional feature space

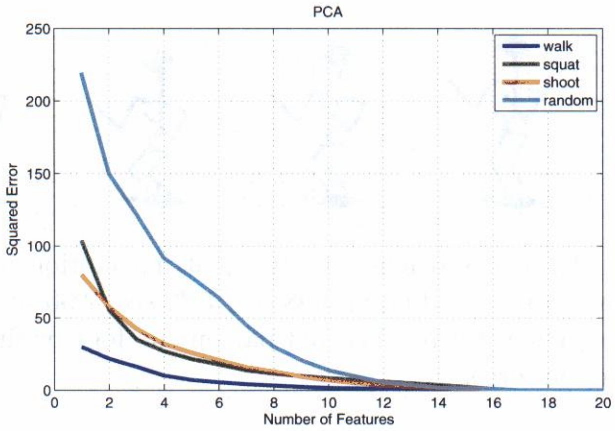


Figure 5.2: The result of PCA for human movements. The horizontal axis means the number of dimension of the features and the vertical axis means the squared error in Eq.(5.3).

does not have the ability to control CoM and orientation of each foot, which are required to perform the squatting motion in this setting.

5.4 Simulation result

Figure 5.5 presents possible velocities that corresponds to each feature space, which is composed of features in principal order. We can find that the manipulability in some direction starts to decrease around dimension 17. With this insight and the reconstruction error presented in Fig.5.2, we selected 18 dimensional feature space for the achievement of the squatting motion. By using 18 dimensional feature space, which is selected in ordering from the largest principal component, the squatting motion was achieved on the HOAP2 model in numerical simulation. The result is shown in Fig.5.4. As a comparison, the squatting motion operated in joint space with the minimum norm constraint is presented in Fig.5.3. The squatting motion was achieved without using the arms, while keeping body in upright. On the contrary, the motion operated in the feature space (Fig.5.4) does use arms and other joints as in the human's demonstration.

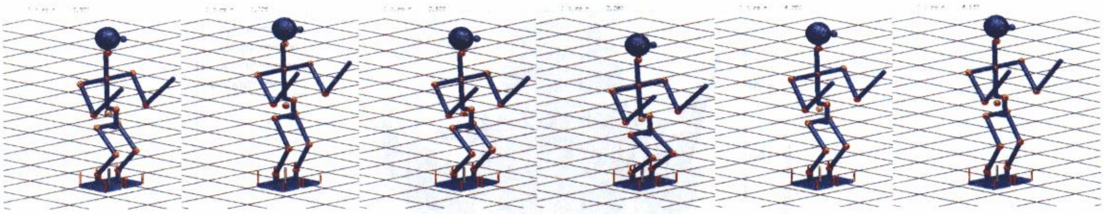


Figure 5.3: An achieved whole-body squatting motion in joint space. The snapshots are corresponding in 0.98 sec, 1.72 sec, 2.90 sec, 3.24 sec, 4.08 sec and 4.94 sec, respectively. The red bar on the foot of the robot means the ground reaction force.

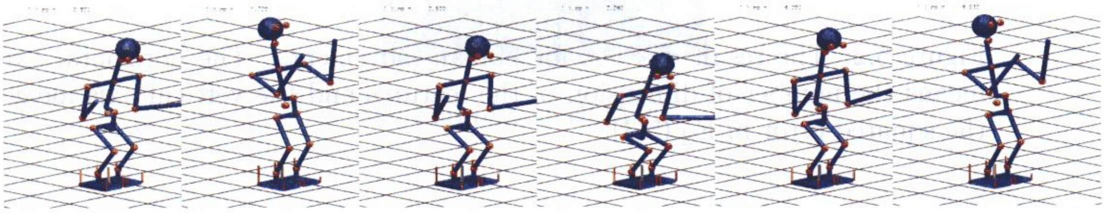


Figure 5.4: An achieved whole-body squatting motion on 18 principal component space.

5.5 Summary and Discussion

In this chapter, a novel redundancy resolution method to achieve CoM control operated in feature space extracted from human's demonstration was presented. As one of examples, the squatting motion was achieved by the proposed method on HOAP2 in numerical simulations. Even though, we only control CoM and Euler angles in task space with these features, the arm and hip movement was also achieved, which can be the characteristics of the observed squatting motion. In the previous chapter, the redundancy problem was solved by a weighted pseudo inverse calculation with manually tuned weight matrix. This framework can be used instead of the pseudo inverse calculation to achieve a desirable joint configuration and to avoid joint limits problems.

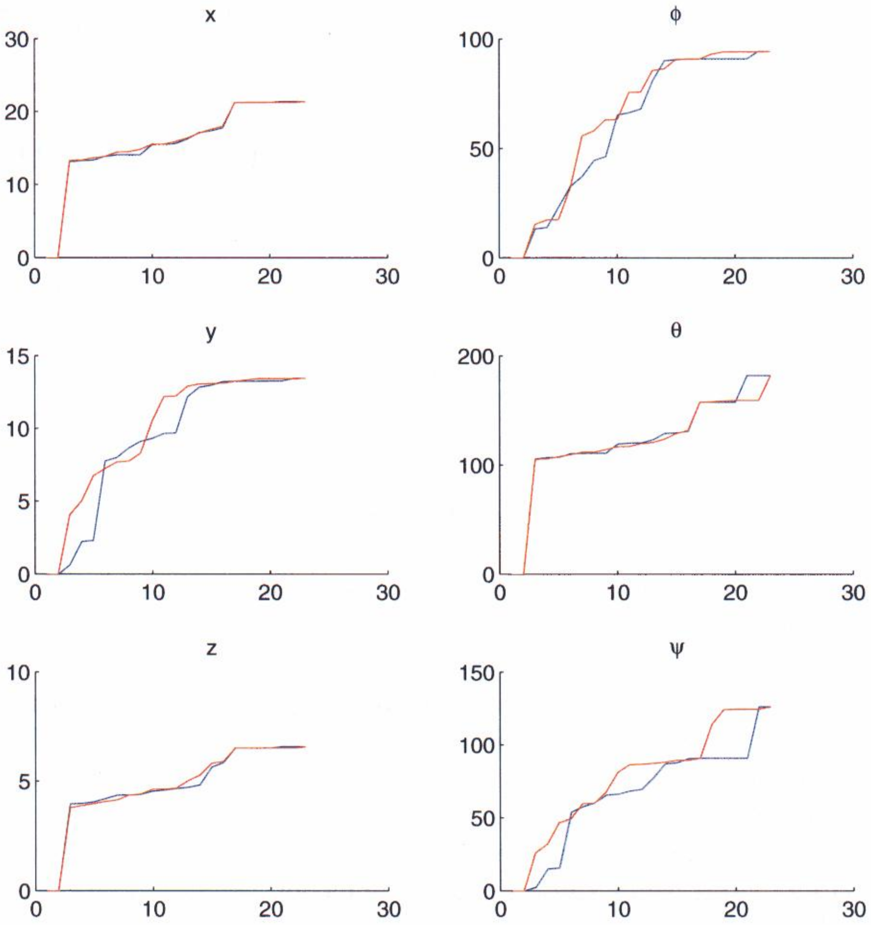


Figure 5.5: Manipulability in each dimension for the given task.

Chapter 6

Conclusion

In this chapter, we conclude this doctoral dissertation. In Section 6.1, we summarize this dissertation. In Section 6.2, discussion with related works is described. In Section 6.3, possible future extensions of our study and open issues are discussed.

6.1 Summary of this dissertation

In this doctoral dissertation, we proposed a paradigm, a motor learning framework for complex whole-body humanoid robots via RL. In order to avoid the *curse of dimensionality*, task-relevant features were utilized to construct low-dimensional space with capability for executing a target task. Then, learning is efficiently achieved in this low-dimensional feature space. This approach makes the RL on such a complex system tractable. The effectiveness of the paradigm was validated through an application of learning biped walking and a class of whole-body dynamic movements. In Chapter 3, we focused on bipedal locomotion with a 5-link biped robot. CPGs were utilized to introduce the natural joint coordination for biped walking and the characterized property of "entrainment" to synchronize the robot with its environment. The learning process was focused on sensory feedback from partial information about the robot, with lower dimensions, while effectively achieving robust walking behaviors. To demonstrate the effectiveness of the approach, the acquired controller in the simulation was successfully implemented in the real environment with a real robot. In Chapter 4, we turn from a biped walking for a 5-link robot to the dynamic full-body movements of a full-body humanoid robot. In so doing, we focused on the dynamically representative features, in particular, the CoM. The CoM is not only just dynamically representative, but also a low dimensional variable, which is also important for taking the balance of the robot explicitly into account. In this work, the CoM controller that utilizes the constraints of keeping ground contact conditions even during learning was introduced, thus, the learning was focused on the CoM movement, which surprisingly simplified the dynamics of the robot. A weighted pseudo inverse calculation distributed the CoM movement to each joint movement, which achieves a joint coordination as a result. In validation, as one of the dynamic full-body movements, we chose a ball-punching task since the task seems to require a momentum (strongly related to CoM movements) control rather than some of joints, which was one of examples as the target movements in this chapter, and applied the proposed framework. As a result, the (sub-) optimal punching motion was acquired within a few thousand trials, which is an extremely small number of trials, considered the original complexity of the robot and the task. We also investigated the performance of the acquired punching motion in a real environment by implementing the acquired policy on a real humanoid robot. In Chapter 5, we proposed a possible solution to an open problem appeared in Chapter 4. We proposed an observation-based redundancy resolution. We then applied the method to achieve whole-body squatting motion because the task can be easily realized by task space control framework, by controlling the CoM and orientation of each foot defined in task space in low-dimensional feature space, which consists of features extracted from human demonstration. The squatting motion with features in joint space observed in human demonstration was achieved in numerical simulation.

Although these results presented in this dissertation support the possibility of *Enhancing Humanoid learning abilities via scalable learning through task-relevant features*, however, we believe that these studies are just early steps towards granting *truly* motor learning abilities to humanoid robots in adapting to new environments as easily as humans and animals. In next section, we discuss contributions of our study with related works. We then propose possible directions for future works to make further steps towards it.

6.2 Discussion with related works

In this section, we discuss the contributions of our studies within the current research field and the relationship with other works addressing to the similar goal – the achievement of capability to adapt to the new environment with humanoid robots.

Learning biped walking behavior

The neural oscillator based approach in robotics has been explored for several tasks with the desirable property referred to as "entrainment" (Miyakoshi, Yamakita, & Furuta, 1994; Taga et al., 1991; Williamson, 1998; Kotosaka & Schaal, 2001; Fukuoka et al., 2003). Taga *et al.* firstly demonstrated the effectiveness of this approach to the robust biped locomotion control of a 2-dimensional simulated robot (Taga et al., 1991). Miyakoshi *et al.* (Miyakoshi, Taga, Kuniyoshi, & Nagakubo, 1998) extended the pioneering work from a 2 dimensional to a 3 dimensional model, while the neural oscillators are rather simplified. Fukuoka *et al.* applied the model to a real quadruped robot, and demonstrated its effectiveness even in a real rough terrain environment (Fukuoka et al., 2003). Even though, all of these pioneering works demonstrated the effectiveness of the CPG-based approach for addressing robust walking with real robots, it required tuning of a large number of the parameters to achieve a desirable controller. Our study exactly addressed this issue via RL. As presented above, we proposed efficient learning for partial parameters of the CPGs, which takes into account of the sensory feedback terms, by utilising a policy gradient method, and presented the effectiveness of the learning and acquired walking in a real environment. Sato *et al.* in (Sato et al., 2002; Nakamura, Sato, & Ishii, 2003) previously applied the RL to this problem, however, the utilized method was the value-function based learning while considering large dimensional state and action space, thus, it resulted in a large number of iterations. Moreover, the simulated robot model was unrealistic from a real robotics point of view. Recently, Mori *et al.* applied policy gradient learning on the model and reported the successive learning in numerical simulations (Mori, Nakamura, Sato, & Ishii, 2004).

In the above studies, the neural oscillator model was utilized as CPGs along with the pioneering work (Taga et al., 1991). The oscillator takes into account of both rhythm and amplitude by the coupled dynamics. As another approach, coupled phase oscillator models as a CPG rather than neural oscillators have been proposed for robust walking, in which the rhythm and amplitude of the oscillation can be separately taken into account by the dynamical equations (Tsuchiya, Aoi, & Tsujita, 2003; Nakanishi, Morimoto, Endo, Cheng, Schaal, & Kawato, 2004; Morimoto et al., 2006). Nakanishi *et al.* presented learning biped locomotion using dynamical movement primitives based on coupled phase oscillators (Nakanishi et al., 2004). In their method, nominal trajectories for each joint are learned from human demonstration by a function approximator. The oscillators, in which the input are composed of amplitude and phase within the dynamical systems, can be easily tuned to have desirable properties in terms of amplitude, phase or frequency. Using their phase resetting and frequency adaptation algorithms with rapid adaptation, robust walking even under disturbances in the real environment with a 5-link biped robot was achieved. The phase resetting for the coupled phase oscillator for biped locomotion was also presented in (Tsuchiya et al., 2003). They utilised a pre-designed nominal gait pattern based controller with phase resetting was able to achieve robust walking in real environment with a 3 dimensional full-body humanoid robot. Morimoto *et al.* utilized the inverted pendulum dynamics of the humanoid robot for phase detection of the coupled phase oscillator based controller, and demonstrated the achievement of robust walking with two different sized humanoid robots, a human and a small size robot. They also reported that phase resetting, while previous works could not achieve stepping with a simple nominal trajectory composed of sinusoidal curves, they proposed a phase modulation method which could achieves stable walking and stepping motions. It may be because the method presented in (Tsuchiya et al., 2003; Nakanishi et al., 2004) modulates only the phase at one particular timing, when the swing leg touches with the ground, whereas (Morimoto et al., 2006) provide continuous modulation.

Another approach, Tedrake *et al.* (Tedrake et al., 2004) presented an on-line learning of walking with a passive dynamic walker based biped robot which was a mechanically well-designed robot. The mechanical design makes the dynamics of the robot much more simple and resulted in quick on-line learning in real environment. However, the success of the quick on-line learning is based on the property of passive dynamic walking. Scaling up the framework to the whole-body humanoid robot is not tractable.

Even though the above methods have several differences, however, all of these methods require initial trajectories (or initial controller) and synergistic relationship among all joints, which makes the dynamics of the robot considerably less, thus making the learning task more tractable. We believe that, in general, on-line modulations such as phase resetting and frequency adaptation algorithms can achieve rapid adaptation to the environment, while it requires appropriate nominal trajectories. In contrast to that, RL approaches do not require highly tuned nominal trajectories, but require a large amount of the trials.

Learning complex whole body movements on a humanoid robot

One of next steps towards truly motor learning is how we achieve automatic synergistic relationship among all joints for several movements, such as: CPG arrangement of the design of coupling among oscillators for the biped walking, to reduce the complexity of the dynamics of the humanoid robot to a simple one. Much of the studies to date focused on biped walking have been able to achieve success with bare intuitions of biological knowledge of the target movement, they were not based on any principle nor objective that could be in fact be systematically automated. Thus, it is still an open problem as to how we could construct such a low-dimensional space to suitably achieve motor learning for several target movements. Chapter 4 has explored this issue by means of examining dynamically representable features, in particular, the CoM movements. The importance of the CoM movements have been well studied in the robotics research field (Nagasaka, 2000; Kagami et al., 2001; Sugihara & Nakamura, 2002; Kajita et al., 2003). The learning of CoM movements makes resolving the high dimensional problem a tractable one. The redundancy problem between the CoM and all-joints can be resolved by utilizing the human demonstration data in a manner that was proposed in Chapter 5. These results indicated a possibility of a learning approach which is not only for the biped walking, but also several whole-body movements on humanoid robots with task-dependent features. Moreover, such an operational space control has been recently focused on as a model for biological movement generation within redundant system (Scholz & Schoner, 1999; Schaal & Schweighofer, 2005; Mistry, Mohajerian, & Schaal, 2005).

6.3 Future works and other open issues

In chapters, we addressed motor learning for humanoid robots via RL within task-relevant features. One of nearest future works is to improve a policy by using a real humanoid robot. More efficient learning frameworks and algorithms are as important as methods to extract lower dimensional feature space. We start developing an efficient RL algorithm. Initial results are presented in Appendix A. This forms the immediate focus of our near future work.

Possible directions in further future works are:

- non-stationarity
- hierarchy
- modularity

We assumed in studies presented in this dissertation that reward function and task-relevant features are fixed, throughout the entire learning process. This assumption can be unrealistic in comparison to a human's motor learning process, since synergies observed in achieved motion may be completely different from the beginning. It would be one of our future works to investigate such a human-like motor learning process. One other aspect of the human's motor learning process can be its hierarchical and modular structure. In this dissertation, we focused on individual motor learning tasks. However, as further steps towards introducing humanoid robots into our living spaces to help us in our daily lives, we must consider how to memorize and generalize the acquired skills. Several computational models based on neuroscience researches have already been proposed, such as MOSAIC (Haruno, Wolpert, & Kawato, 2001) and some applications of such learning have also been explored in (Morimoto & Doya, 2001; Yoshimoto, Nishimura, Tokita, & Ishii, 2005). Another approach is taking a model of the human movement for efficient learning (Miyamoto, Schaal, Gandolfo, Gomi, Koike, Osu, Nakano, Wada, & Kawato, 1996). The development of a new framework focusing on real humanoid robots based on these past studies will be considered.

Appendix A

Novel policy gradient approach: Towards "*policy Newton method*"

In this chapter we propose a novel policy gradient type reinforcement learning method on the average reward manifold, in which a metric to measure the effect of change in policy parameters on the average reward is introduced. In our method, the derivative of the average reward with respect to the policy improvement can be fixed as a constant. Moreover, around a (sub-) optimal policy, the policy gradient method is equivalent to the Newton method. Simple simulation results with comparison to previously proposed natural policy gradient methods demonstrate the effectiveness of our policy gradient method.

A.1 The Policy Gradient On The Average Reward Manifold

Although the natural policy gradient presented in Chapter 2 does not affect the policy parameterization on its performance, however, it still affects the design of reward function. This is caused by the fact that the natural gradient considers the manifold of the state-action probabilistic distribution. Kakade pointed out in (Kakade, 2002) that the choice of the metric is not necessarily asymptotically efficient, *i.e.*, does not attain second order convergence. Bagnell *et al.* also mentioned again in (Bagnell & Schneider, 2003) that the metric is not unique to derive a natural policy gradient. The group then clarified the Kakade's metric as path distribution manifolds as presented in previous section.

In fact, in the natural gradient approach, "a small parameter change only affects on a fixed small change in the state-action distribution". But it does not mean that "a small parameter change only affects on a fixed small change in the performance". The natural policy gradient often achieve faster convergence than ordinal policy gradients in practical cases (Kakade, 2002; Bagnell & Schneider, 2003; Peters et al., 2003). However, the derivative of the average reward at every policy improvement can be different during learning, and it may depend on tasks (or design of reward function). It means that even though the natural policy gradient empirically achieves faster convergence than ordinal policy gradients, it does not still have any theoretical evidence for performance improvement in terms of the convergence speed. As presented in Chapter 2, the metric introduced in previously proposed natural policy gradient measures the effect in $p(s,a) = d^{\pi}(s)\pi(s,a;\boldsymbol{\theta})$ which is significantly related with the average reward. However, it does not yet measure the effect in the average reward $\eta(\boldsymbol{\theta}) = \int_s d^{\pi}(s) \int_u \pi(s,a)r(s,a)dads$. Obviously, if we can measure the effect of change in policy parameter to the average reward directly, it can be expected that the derivative of the change in the average reward at every policy improvement is constant, which results in the fast convergence rate independently on tasks.

We derive the metric which can measure the effect of change in policy parameter to the average reward directly. The average reward is not probabilistic variable, so no probability manifold. Instead of the way used in the previous section, we here simply define the following Euclidean distance in order to measure the effect of change in policy parameter to the average reward and use the first-order Taylor series expansion as:

$$
\begin{aligned}
|d\boldsymbol{\theta}|^2 &= (\eta(\boldsymbol{\theta}+d\boldsymbol{\theta}) - \eta(\boldsymbol{\theta}))^T(\eta(\boldsymbol{\theta}+d\boldsymbol{\theta}) - \eta(\boldsymbol{\theta})) \\
&\approx d\boldsymbol{\theta}^T \left\{ \frac{\partial \eta(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \frac{\partial \eta(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}^T \right\} d\boldsymbol{\theta} \\
&= d\boldsymbol{\theta}^T G(\boldsymbol{\theta}) d\boldsymbol{\theta}. \qquad \text{(A.1)}
\end{aligned}
$$

The policy gradient method with the above metric is given as

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \hat{g}(\boldsymbol{\theta}) \qquad \text{(A.2)}$$

where, $\hat{g}(\boldsymbol{\theta}) = G(\boldsymbol{\theta})^{-1}\frac{\partial\eta(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}$ is the policy gradient on the average reward manifold and it can be expected that the derivative of the change in the average reward at every policy improvement with the gradient is constant. Note that the matrix $G(\boldsymbol{\theta})$ is composed of only ordinal policy gradient $\frac{\partial\eta(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}$. It means that to achieve this novel policy gradient reinforcement learning does not require any additional estimation from ordinal policy gradient at all.

We can also explain the effectiveness of our proposed method in a different way. Further analysis to understand this new approach is given in (Amari, Park, & Fukumizu, 2000). Assuming (sub-) optimal average reward $\eta^*(\boldsymbol{\theta}^*)$ and the average reward for the current policy $\eta(\boldsymbol{\theta})$, we consider the error between them as $E(\boldsymbol{\theta}) = \frac{1}{2}|\eta^*(\boldsymbol{\theta}^*)-\eta(\boldsymbol{\theta})|^2$. By taking an second-order Taylor series expansion and simple analytical calculations, the optimal parameter update is

$$\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta} - \frac{\partial^2 E(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^2}^{-1}\frac{\partial E(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}, \tag{A.3}$$

where, $\frac{\partial E(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} = -\left(\eta^*(\boldsymbol{\theta}^*) - \eta(\boldsymbol{\theta})\right)\frac{\partial\eta(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}$ and $\frac{\partial^2 E(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^2} = \frac{\partial\eta(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\left[\frac{\partial\eta(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right]^T - \left(\eta^*(\boldsymbol{\theta}^*) - \eta(\boldsymbol{\theta})\right)\frac{\partial^2\eta(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^2}$. When $|\eta^*(\boldsymbol{\theta}^*)-\eta(\boldsymbol{\theta})| \approx 0$, the Hessian becomes $\frac{\partial^2 E(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^2} \approx G(\boldsymbol{\theta})$. Therefore, our proposed method can be equivalent (proportional) to the "Newton method" around a (sub-) optimal policy. This is first work to discuss "the second order convergence" in the context of policy gradient type reinforcement learning even around a (sub-) optimal policy. Note that the update rule Eq.(A.2) is also equivalent (proportional) to Gauss-Newton method which is a well-known algorithm in non-linear optimizations (Bertsekas, 1999).

However, $G(\boldsymbol{\theta})$ in Eq.(A.2) cannot be full-rank in general. As one of simplest solutions for calculating Eq.(A.2), we can replace $G(\boldsymbol{\theta})$ by $\hat{G}(\boldsymbol{\theta}) = G(\boldsymbol{\theta}) + \epsilon\mathbf{I}$ where ϵ is small value and $\mathbf{I}$ is the identity matrix. It can be interpreted as the weighted metric between the metric on the average reward manifold and the metric which appears in the ordinal policy gradient.

A.2 An another form of the policy gradient on the average reward manifold with minimum norm constraint

As described above, in Eq.(A.2), the matrix $G(\boldsymbol{\theta})$ is not invertible in general. Here, we derive an another form of the gradient by introducing the minimum norm constraint in the gradient, which does not require inverse matrix calculation, not as Eq.(A.2). We consider the first-order Taylor series expansion of the average reward at parameter $\boldsymbol{\theta}$ as:

$$\eta(\boldsymbol{\theta} + d\boldsymbol{\theta}) \approx \eta(\boldsymbol{\theta}) + \frac{\partial \eta(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}^T d\boldsymbol{\theta}. \tag{A.4}$$

From the above, the policy gradient with the constraint $d\boldsymbol{\theta}^T G(\theta) d\boldsymbol{\theta} = \epsilon$ can be also considered as $d\boldsymbol{\theta}$ which satisfies the following condition as

$$\frac{\partial \eta(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}^T d\boldsymbol{\theta} = \sqrt{\epsilon}. \tag{A.5}$$

Equation(A.2) can be derived from the above by taking $\hat{g} = G(\boldsymbol{\theta})^{-1} \frac{\partial \eta(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \propto \mathrm{argmin}_{d\boldsymbol{\theta}} || \frac{\partial \eta(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}^T d\boldsymbol{\theta} - \sqrt{\epsilon} ||^2$. However, it is obviously an ill-posed problem if the dimension of policy parameters is larger than one as most cases. In order to address this problem, we derive an another form of the policy gradient by introducing the minimum norm constraint in the gradient. It can be derived as $\tilde{g} \propto \mathrm{argmin}_{d\boldsymbol{\theta}} ||d\boldsymbol{\theta}^T d\boldsymbol{\theta}||$ s.t. $\frac{\partial \eta(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}^T d\boldsymbol{\theta} = \sqrt{\epsilon}$ which can be written in

$$\tilde{g} = \frac{1}{\gamma(\boldsymbol{\theta})} \frac{\partial \eta(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \tag{A.6}$$

where $\gamma(\boldsymbol{\theta}) = \left\{ \frac{\partial \eta(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}^T \frac{\partial \eta(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\}$ is scalar and its inverse is easily calculated in most cases. In calculation cost and matrix inversion issue's points of view, this policy gradient is more useful and practical than Eq.(A.2). In the following sections, we use Eq.(A.6) for calculation of the policy gradient on the average reward manifold.

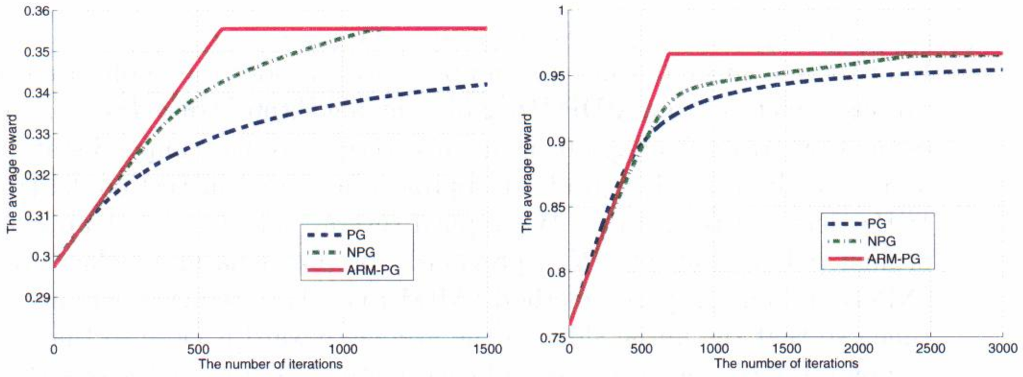
A.3 Applications to 3-state MDP

In order to confirm our discussion above, we consider the application of the proposed method to a MDP. To avoid any algorithmic disturbance, *e.g.*, an estimation error of the gradient, in this section, we use simple 3-state MDP and an analytical policy gradient approach presented in Section 2.4 (Baxter & Bartlett, 2001b, 2001a). We applied several policy gradient approaches, ordinal policy gradient (PG), previously proposed natural policy gradient (NPG) and our proposed method (ARM-PG). The step size parameter α is tuned so that the initial slope of the average reward is close so that we can compare the convergence speed of each method. Figure A.1 shows that PG and NPG get changes in the derivative of the average reward during learning and it results in slower convergence in both case. ARM-PG results in best performance among three methods in the sense of convergence speed in our simulations.

A.4 An algorithm for estimating the policy gradient on the average reward manifold in MDPs

In previous section, we calculated policy gradients by an analytical approach using models of the environment and reward function. In this section, we introduce an simple algorithm to estimate the policy gradient on the average reward manifold without using any models of the environment and reward function. According to (Baxter & Bartlett, 2001b; Sutton et al., 2000) and with Eq.(A.6), the gradient can be estimated by using Monte Carlo estimation as:

$$\tilde{g} = \frac{1}{\gamma(\boldsymbol{\theta})} \frac{\partial \eta(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \tag{A.7}$$



(a) Case 1

(b) Case 2

Figure A.1: Simulation results obtained by several policy gradient methods with models of the environment and reward function. The case 1 is obtained with the reward function $\mathbf{r} = [1.0, 0.0, 0.0]^T$. The case 2 is with the reward function $\mathbf{r} = [0.0, 1.0, 1.0]^T$. The dashed, dash-dot and solid lines are ordinal (PG), previously proposed natural (NPG), our proposed method (ARM-PG), respectively. The step size parameter α is tuned so that the initial convergence speed is same in order to demonstrate the fast convergence of the proposed method especially around a (sub-) optimal policy.

where,

$$\begin{aligned}
&\frac{\partial \eta(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = (1-\beta)\frac{\partial \eta_\beta(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \\
= \;& (1-\beta)\int d(s)\int \pi(s,a)\Big[\nabla \log d(s) \\
& + \frac{1}{1-\beta}\nabla \log \pi(s,a)\Big]\,(Q_\beta(s,a) - b(s))\, dads \\
= \;& \int d(s)\int \pi(s,a)\Big[\,(1-\beta)\,\nabla \log d(s) \\
& + \nabla \log \pi(s,a)\Big]\,(Q_\beta(s,a) - b(s))\, dads \\
= \;& \lim_{\beta\to 1}\int d(s)\int \pi(s,a)\,[\nabla \log \pi(s,a)]\,(Q_\beta(s,a) - b(s))\, dads \\
= \;& \lim_{T\to\infty}\lim_{\beta\to 1}\frac{1}{T}\sum_{t=0}^{T-1}\nabla \log \pi(s_t,a_t)\sum_{i=t+1}^{T}\beta^{i-t-1}(r(s_i,a_i)) - b(s_i) \\
= \;& \lim_{T\to\infty}\lim_{\beta\to 1}\frac{1}{T}\sum_{t=1}^{T}(r(s_t,a_t) - b(s_t))\sum_{i=0}^{t-1}\beta^{t-i-1}\nabla \log \pi(s_i,a_i), \quad \text{(A.8)}
\end{aligned}$$

$\eta_\beta(\boldsymbol{\theta}) = \lim_{T\to\infty} \boldsymbol{E}\left[\frac{1}{T}\sum_{t=0}^{T}\beta^t r(s_t, a_t)\right]$ and $Q_\beta^\pi(s,a) = \boldsymbol{E}\left[\sum_{k=0}^{\infty}\beta^k r(s_{t+k+1}, a_{t+k+1})\middle| s_t = s, a_t = a\right]$. $b(s)$ is called reward baseline which can be a variable for variance reduction of the estimator (Williams, 1992; Kimura & Kobayashi, 1998; Weaver & Tao, 2001; Greensmith et al., 2004). The proposed algorithm is described in Algorithm 1. If the policy parameter is updated by the ordinal policy gradient, *i.e.*, the line 11 is replaced by $\theta \leftarrow \theta + \alpha g_T$, the algorithm is exactly the MCG algorithm (Baxter & Bartlett, 2001b). However, such a small difference can drastically improve the convergence speed of the learning process as presented in next section.

Algorithm 1 An ARM-PG (Average Reward Manifold based Policy Gradient) algorithm

```
1.Given: an ergodic MDP, parameterized by $\theta$
2.Set $\theta \in R^m$, $\alpha = [0,1)$ and $\beta = [0,1)$
3.Until $\sqrt{g_T^T g_T} = 0$ (convergence) do
4.        Set $z = 0$, $g = 0$
5.        For $t = 0$ to $T - 1$ do
6.            state transition from $s_t$ with action $a_t$ to $s_{t+1}$
7.            $z_{t+1} = \beta z_t + \frac{\partial \log \pi(s_t,a_t)}{\partial\theta}$
8.            $g_{t+1} = g_t + \frac{1}{t+1}\left[\left(r(s_{t+1}, a_{t+1}) - b(s_{t+1})\right) z_{t+1} - g_t\right]$
9.        End for
10.       $\gamma = \left\{g_T^T g_T\right\}, \quad \tilde{g}_T = g_T/\gamma$
11.       $\theta \leftarrow \theta + \alpha\tilde{g}_T$
12.End until
```

A.5 Simulation results

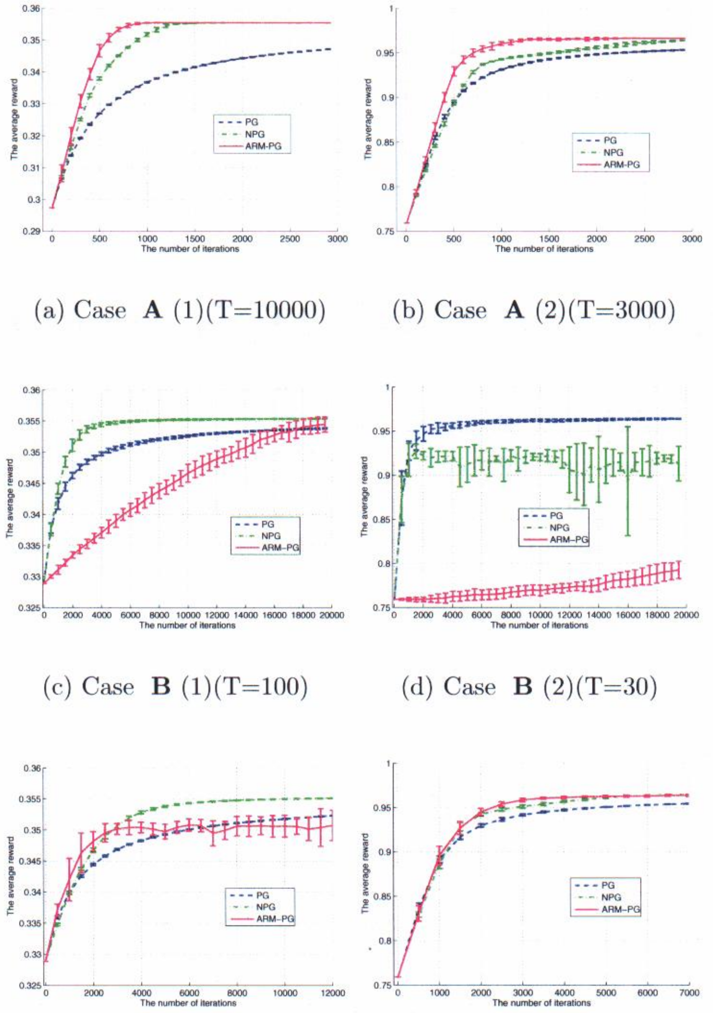
We investigate the effectiveness of an ARM-PG algorithm presented in previous section with comparison to other policy gradient methods. We again use the 3-state MDP. We selected six settings for simulations, which are composed of different settings in sample size, reward function and reward baseline. Case **A** supplies large enough samples for a gradient estimation, case **B** supplies only small samples as a more practical situation and case **C** supplies same samples with simultaneous estimation of a (sub-) optimal constant reward baseline (*i.e.* the average reward as a reward baseline)(Weaver & Tao, 2001). Use of such a baseline often reduces the variance of the estimated policy gradient.

Although the optimal baseline is presented by (Greensmith et al., 2004), the estimation requires more complex calculation than estimation of the average reward (sub-)optimal baseline. We implemented our proposed policy gradient algorithm (ARM-PG), MCG(PG)(Baxter & Bartlett, 2001b) and MCG with the following Fisher information matrix estimator $F(\boldsymbol{\theta}) \approx \frac{1}{T}\sum_{i=1}^{T}\left\{\frac{\partial \log \pi_i}{\partial \theta}\frac{\partial \log \pi_i}{\partial \theta}^T\right\}$ and multiplies its inverse with the estimated gradient to achieve the natural policy gradient (NPG) (Kakade, 2002) for all settings with two reward functions.

The result is presented in Fig.A.2. In case **A** with both reward function, ARM-PG resulted the best performance among all methods as well as the case with the analytically calculated gradients in Fig.A.1. However, in case **B** , ARM-PG presented significantly deteriorated performance compared with other two methods. For case **C** , the deterioration greatly changed for the better, in particular, our method presented the best performance in case **C** (2). These results suggest that the variance of the gradient estimator significantly affects on the performance of our method more than other two methods.

A.6 Discussions

In this chapter we pointed out that the previously proposed natural policy gradient method is not best choice in convergence speed. We proposed a novel policy gradient approach with the metric based on the average reward manifold. Moreover, we showed that our method is equivalent to the Newton method rather than gradient decent around an equilibrium point. Therefore, our study can be considered as an initial attempt to develop a reinforcement learning algorithm with a higher rate of convergence. Simple simulation results justified our approach. These results also suggest that even though our proposed method can significantly improve policy gradient with little computation cost, however, variance of the gradient estimator affects on its performance. This is one of our future works to be solved.



(a) Case **A** (1)(T=10000)

(b) Case **A** (2)(T=3000)

(c) Case **B** (1)(T=100)

(d) Case **B** (2)(T=30)

(e) Case **C** (1)(T=100)

(f) Case **C** (2)(T=30)

Figure A.2: Simulation results obtained by several policy gradient methods without any models of the environment and reward function. In each case, (1) is obtained with the reward function $\mathbf{r} = [1.0, 0.0, 0.0]^T$, and (2) is with the reward function $\mathbf{r} = [0.0, 1.0, 1.0]^T$. The dashed, dash-dot and solid lines are ordinal (PG), previously proposed natural (NPG), our proposed method (ARM-PG), respectively. The step size parameter α is tuned so that the initial convergence speed is same. All cases set $\beta = 0.9$ and each result was averaged over 10 runs.

Appendix B

Basics of Zero Moment Point

We define that $\boldsymbol{\tau}$ is moment at origin, and $\mathbf{p}$, $\mathbf{f}$ and $\boldsymbol{\tau}_p$ are the position of the Zero Moment Point (ZMP), the force and moment on the point, respectively as depicted in Fig.B.1. The relation between two points are obtained as

$$\boldsymbol{\tau} = \mathbf{p} \times \mathbf{f} + \boldsymbol{\tau}_p. \tag{B.1}$$

By substituting the following equation to the above equations

$$\dot{\mathbf{P}} = M\mathbf{g} + \mathbf{f}, \tag{B.2}$$

$$\dot{\mathbf{L}} = \mathbf{c} \times M\mathbf{g} + \tau, \tag{B.3}$$

Eq.(B.1) can be written in a different form as

$$\boldsymbol{\tau}_p = \dot{L} - \mathbf{c} \times M\mathbf{g} + \left(\dot{\mathbf{P}} - M\mathbf{g}\right) \times \mathbf{p}, \tag{B.4}$$

where, $\mathbf{P}$ and $\mathbf{L}$ are the momentum and angular momentum. From the definition of the ZMP so that horizontal moments of on the ZMP are zero, we can obtain the following ZMP-equations:

$$p_x = \frac{Mgx + p_z\dot{P}_x - \dot{L}_y}{Mg + \dot{P}_z}, \tag{B.5}$$

$$p_y = \frac{Mgy + p_z\dot{P}_y + \dot{L}_x}{Mg + \dot{P}_z}. \tag{B.6}$$

Moreover, assuming the point mass model as

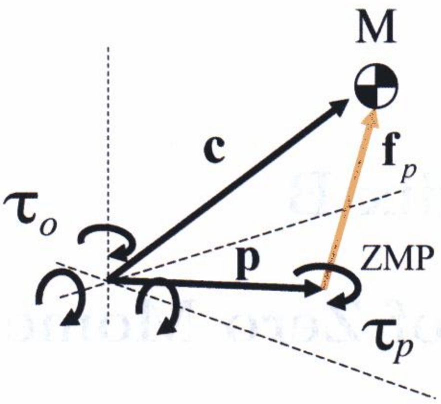


Figure B.1: ZMP

$$P = M\dot{\mathbf{c}}, \tag{B.7}$$
$$L = \mathbf{c} \times M\dot{\mathbf{c}}, \tag{B.8}$$

the position of the ZMP can be obtained by following equations:

$$p_x = x - \frac{(z - p_z)\ddot{x}}{\ddot{z} + g}, \tag{B.9}$$
$$p_y = y - \frac{(z - p_z)\ddot{y}}{\ddot{z} + g}. \tag{B.10}$$

From the analogy to the dynamics of the linearized-inverted pendulum, it is sometimes called as *the inverted pendulum model.* Here, we are assuming that the force acting on the point mass is only ground reaction force on ZMP. In this case, *i.e.*,

$$f_z = M\ddot{z} + Mg, \tag{B.11}$$
$$f_x = M\ddot{x}, \tag{B.12}$$

we can obtain the following equation:

$$\frac{x - p_x}{z - p_z} = \frac{M\ddot{x}}{M\ddot{z} + Mg}. \tag{B.13}$$

This means that in the case of the point mass model, the ground reaction force always penetrates from ZMP to CoM. This insight may be useful to intuitively understand the limitations of the ZMP-based approach.

Appendix C

The derivation of a policy gradient algorithm

A derivation of a policy gradient algorithm proposed by (Kimura & Kobayashi, 1998) is presented. First of all, the average reward J and discounted reward J_α are defined as follows:

$$J_\alpha = \int d^\pi(x) V_\alpha(x) dx = \int d^\pi(x) \int \pi(u|x) \left[Q_\alpha(x,u) - b(x) \right] du dx \tag{C.1}$$

$$J = \int d^\pi(x) \int \pi(u|x) r(x,u) du dx \tag{C.2}$$

where, $d^\pi(x)$ is the steady-state distribution, $\pi(u|x)$ is a stochastic policy. The policy gradient for the J_α with respect to the policy parameter θ is given as

$$\begin{aligned} \nabla J_\alpha &= \int \nabla d^\pi(x) \int \pi(u|x) \left[Q_\alpha(x,u) - b(x) \right] du dx \\ &+ \int d^\pi(x) \int \nabla \pi(u|x) \left[Q_\alpha(x,u) - b(x) \right] du dx \\ &+ \int d^\pi(x) \int \pi(u|x) \nabla Q_\alpha(x,u) du dx \end{aligned} \tag{C.3}$$

where,

$$\begin{aligned}
& \int d^{\pi}(x) \int \pi(u|x) \nabla Q_{\alpha}(x,u) du dx \\
= & \int d^{\pi}(x) \int \pi(u|x) \left[\nabla r(x,u) + \alpha \int p(x'|x,u) \nabla V_{\alpha}(x') dx' \right] du dx \\
= & \ \alpha \int d(x) \int \pi(u|x) \int p(x'|x,u) \nabla V_{\alpha}(x') dx' du dx \\
= & \ \alpha \int d(x') \nabla V_{\alpha}(x') dx' \\
= & \ \alpha \int d(x) \nabla V_{\alpha}(x) dx \\
= & \ \alpha \left\{ \nabla \int d(x) V_{\alpha}(x) dx - \int \nabla d(x) V_{\alpha}(x) dx \right\} \\
= & \ \alpha \nabla J_{\alpha} - \alpha \int \nabla d(x) \int \pi(u|x) Q_{\alpha}(x,u) du dx. \qquad \text{(C.4)}
\end{aligned}$$

Thus,

$$\begin{aligned}
\nabla J_{\alpha} &= \int \nabla d(x) \int \pi(u|x) Q_{\alpha}(x,u) du dx + \frac{1}{1-\alpha} \int d(x) \int \nabla \pi(u|x) Q_{\alpha}(x,u) du dx \\
&= \int d(x) \int \pi(u|x) \left[\nabla \log d(x) + \frac{1}{1-\alpha} \nabla \log \pi(u|x) \right] Q_{\alpha}(x,u) du dx \qquad \text{(C.5)}
\end{aligned}$$

In (Baxter & Bartlett, 1999), the following theorem was presented:

$$J_{\alpha} = \frac{J}{1-\alpha}.$$

We can easily derive the theorem as follows:

$$\begin{aligned}
J_\alpha &= \int d^\pi(x)V_\alpha(x)dx \\
&= \int d^\pi(x)\int \pi(u|x)\left[Q_\alpha(x,u)\right] dudx \\
&= \int d^\pi(x)\int \pi(u|x)\left[r(x,u)+\alpha\int p(x'|x,u)V_\alpha(x')dx'\right] dudx \\
&= \int d^\pi(x)\int \pi(u|x)r(x,u)dudx + \alpha\int d^\pi(x')V_\alpha(x')dx' \\
&= J+\alpha J_\alpha
\end{aligned} \tag{C.6}$$

Thus,

$$\nabla J_\alpha = \frac{\nabla J}{1-\alpha} \propto \nabla J. \tag{C.7}$$

With the above equation, we can derive the policy gradient for the average reward as:

$$\begin{aligned}
\nabla J &= (1-\alpha)\nabla J_\alpha \\
&= (1-\alpha)\int d(x)\int \pi(u|x)\left[\nabla\log d(x)+\frac{1}{1-\alpha}\nabla\log\pi(u|x)\right] Q_\alpha(x,u)dudx \\
&= \int d(x)\int \pi(u|x)\left[(1-\alpha)\nabla\log d(x)+\nabla\log\pi(u|x)\right] Q_\alpha(x,u)dudx \\
&= \lim_{\alpha\to 1}\int d(x)\int \pi(u|x)\left[\log\pi(u|x)\right] Q_\alpha(x,u)dudx \\
&= \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\log\pi_t(u|x)\sum_{s=t+1}^{T}\alpha^{s-t-1}(r_s(x,u)) \\
&= \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}(r_t(x,u))\sum_{s=0}^{t-1}\alpha^{t-s-1}\log\pi_s(u|x).
\end{aligned} \tag{C.8}$$

The Kimura's algorithm (Kimura & Kobayashi, 1998; Konda & Tsitsiklis,

2003) is explained by using the above policy gradient derivation as follows:

$$
\begin{aligned}
\nabla J &= (1-\beta)\nabla J_\beta && \text{(C.9)}\\
&= (1-\beta)\int d(x)\int \pi(u,x)[\nabla \log d(x) \\
&\quad +\frac{1}{1-\beta}\nabla \log \pi(u,x)]Q_\beta(x,u)dudx && \text{(C.10)}\\
&= \int d(x)\int \pi(u,x)[(1-\beta)\,\nabla \log d(x) && \text{(C.11)}\\
&\quad +\nabla \log \pi(u,x)]\left\{Q_\beta(x,u)-V_\beta(x)\right\}dudx \\
&= \lim_{\beta\to 1}\int d(x)\int \pi(u,x)\nabla \log \pi(u,x)\left\{Q_\beta(x,u)-V_\beta(x)\right\}dudx \\
&= \lim_{T\to\infty,\beta\to 1}\frac{1}{T}\sum_{t=0}^{T}\nabla \log \pi(u_t,x_t)\left\{\sum_{s=t}^{T}\beta^{s-t}(r(x_s,u_s)) - V_\beta(x_t)\right\} \\
&= \lim_{T\to\infty,\beta\to 1}\frac{1}{T}\sum_{t=0}^{T}\nabla \log \pi(u_t,x_t)\sum_{s=t}^{T}\beta^{s-t}\left\{r(x_s,u_s)+\beta V_\beta(x_{s+1})-V_\beta(x_s)\right\} \\
&= \lim_{T\to\infty,\beta\to 1}\frac{1}{T}\sum_{t=0}^{T}\nabla \log \pi(u_t,x_t)\sum_{s=t}^{T}\beta^{s-t}\delta(x_s,u_s) \\
&= \lim_{T\to\infty,\beta\to 1}\frac{1}{T}\sum_{t=0}^{T}\delta(x_t,u_t)\sum_{s=0}^{t}\beta^{t-s}\nabla \log \pi(u_s,x_s). && \text{(C.12)}
\end{aligned}
$$

Appendix D

Natural Gradient Algorithm for Inverse Kinematics

This chapter discusses Inverse Kinematics (IK) solutions with gradient decent methods for redundant robots like humanoids. From the information geometric point of view, we show that the direction of the gradient in the traditional gradient decent based IK solution is not in the steepest direction. Furthermore, we derive the natural gradient which gives the steepest direction for the IK problems. We also point out that the natural gradient based IK solution is equivalent to the Jacobian pseudoinverse based IK solution.

D.1 Introduction

The importance of the IK problem has been growing especially for recent redundant robots like humanoids. For such redundant situations, numerical approach is in general preferable because the relationship between a point in task space and joint angles has non-linearity (*e.g.* (Sugihara & Nakamura, 2002; Riley et al., 2003; Nakanishi et al., 2005; Matsubara, Morimoto, Nakanishi, Hyon, G.Hale, & Cheng, 2007)). Representative algorithms for the IK are Jacobian transpose algorithm and Jacobian pseudo inverse algorithm(L.Sciavicco & B.Siciliano, 1997; Yoshikawa, 1990). The Jacobian transpose algorithm can be interpreted as the gradient method for the solution of a system on nonlinear equations. However, Amari *et al.*(Amari, 1998) has pointed out that such a ordinal gradient does not present to the steepest direction because the parameter, in general, has Riemannian space rather than Euclidean. The natural gradient decent which is steepest gradient decent has then proposed, which explicitly consider the metric with the structure of the parameter space (Amari, 1998).

In this chapter, we firstly consider the metric in the joint space for the IK problem, and then analytically derive the resulting natural gradient. Furthermore, we discuss the equivalence between the natural gradient decent algorithm and Jacobian pseudo inverse algorithm in IK. To present the effectiveness of the derived natural gradient, we apply it with 3-link redundant manipulator in numerical simulations.

D.2 Gradient (Jacobian transpose) method for inverse kinematics

We define the position of the end effector from reference inertial frame $\mathbf{x}$ and velocity $\dot{\mathbf{x}}$. The relation between $\mathbf{x}$ and joint vector $\mathbf{q}$, and the velocities are presented as

$$\mathbf{x} = f(\mathbf{q}), \tag{D.1}$$

$$\dot{\mathbf{x}} = \frac{\partial \mathbf{x}}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} = J(\mathbf{q})\dot{\mathbf{q}}, \tag{D.2}$$

where, $J(\mathbf{q})$ is the Jacobian.

The objective function E for IK is given as

$$E(\mathbf{q}) = \left(\mathbf{x}_d - f(\mathbf{q})\right)^T \left(\mathbf{x}_d - f(\mathbf{q})\right), \tag{D.3}$$

$$\tag{D.4}$$

where, $\mathbf{x}_d$ is the target position.

By taking approximated Taylor series expansion using first-order term for $E(\mathbf{q})$, we can derive

$$E(\mathbf{q} + \eta\mathbf{a}) = E(\mathbf{q}) + \eta \left(\frac{\partial E(\mathbf{q})}{\partial \mathbf{q}}\right)^T \mathbf{a}, \tag{D.5}$$

where, η is a small value. The purpose is to obtain the update direction $\mathbf{a}$ for minimizing the $E(\mathbf{q} + \eta\mathbf{a})$. To avoid the infinitely large norm for $\mathbf{a}$, the following constraint is introduced:

$$\|\eta \mathbf{a}\|^2 = \mathbf{a}^T G \mathbf{a} = 1. \tag{D.6}$$

By the Lagrangian method, the optimal update direction can be obtained from $\frac{\partial}{\partial \mathbf{a}}\{E(\mathbf{q}+\eta\mathbf{a}) - \lambda(\mathbf{a}^T G\mathbf{a} - 1)\} = 0$ as (Amari, 1998)

$$\mathbf{a} \propto G^{-1} J(\mathbf{q})^T (\mathbf{x}_d - \mathbf{x}). \tag{D.7}$$

The ordinary gradient (Jacobian transpose) method takes the update-rule such that $\mathbf{q} \leftarrow \mathbf{q} + \eta \frac{\partial E(\mathbf{q})}{\partial \mathbf{q}} (\mathbf{x}_d - \mathbf{x})$(L.Sciavicco & B.Siciliano, 1997). It is updated in the steepest direction if and only if $G = I$. However, in general, G is task-dependent matrix. In the next chapter, we firstly consider the structure G for inverse problem, then we derive the natural gradient method based on it.

D.3 The steepest gradient for inverse kinematics

In this section, we derive the structure of Riemannian G for the IK. We derive it for deterministic case presented in Eq.(D.1) and stochastic case, respectively. For the deterministic case, the Euclid distance of end-effector is used for the metric in the joint space. For the stochastic case, Kullback Leibler divergence is used.

D.3.1 The natural gradient based on the Euclid distance of end-effector

By using the Taylor series expansion, we can derive a metric in Riemannian form as

$$\begin{aligned} ds^2 &= (f(\mathbf{q}+\eta\mathbf{a}) - f(\mathbf{q}))^T (f(\mathbf{q}+\eta\mathbf{a}) - f(\mathbf{q})) \\ &\approx \eta^2 \mathbf{a}^T J(\mathbf{q})^T J(\mathbf{q}) \mathbf{a}, \end{aligned} \tag{D.8}$$

where, $f(\mathbf{q}+\eta\mathbf{a}) \approx f(\mathbf{q}) + \eta J(\mathbf{q})\mathbf{a}$. It suggests that the IK problem has Riemannian which has the structure $G = J(\mathbf{q})^T J(\mathbf{q})$. Hence, from Eq.(D.7), the natural gradient with a constraint $ds^2 = 1$ is given as

$$\mathbf{a} \propto \left(J(\mathbf{q})^T J(\mathbf{q})\right)^{-1} J(\mathbf{q})^T (\mathbf{x}_d - \mathbf{x}). \tag{D.9}$$

D.3.2 The natural gradient based on the Kullback Leibler distance of end-effector

We assume the normal distribution noise ξ as the observation noise in the position of end-effector. In this stochastic case, the relation between a point of end-effector $\mathbf{x}$ and joint angle $\mathbf{q}$ can be presented as

$$\mathbf{x} = f(\mathbf{q}) + \xi, \tag{D.10}$$

$$\begin{aligned} p(\mathbf{x}|\mathbf{q}) =& \frac{1}{(2\pi)^{D/2}\|\Sigma\|^{\frac{1}{2}}} \\ & \cdot \exp\left\{-\frac{1}{2}\left(\mathbf{x} - f(\mathbf{q})\right)^T \Sigma^{-1}(\mathbf{x} - f(\mathbf{q}))\right\}, \end{aligned} \tag{D.11}$$

where, $\xi \sim N(0, \Sigma)$. We calculate the Kullback Leibler divergence $D(\mathbf{q}||\mathbf{q} + \eta\mathbf{a})$ between $p(\mathbf{x}|\mathbf{q})$ and $p(\mathbf{x}|\mathbf{q} + \eta\mathbf{a})$ as

$$D(\mathbf{q}||\mathbf{q} + \eta\mathbf{a}) = \int p(\mathbf{x}|\mathbf{q}) \log \frac{p(\mathbf{x}|\mathbf{q})}{p(\mathbf{x}|\mathbf{q} + \eta\mathbf{a})} d\mathbf{x}. \tag{D.12}$$

We again use the Taylor series expansion for $\log p(\mathbf{x}|\mathbf{q} + \eta\mathbf{a})$ at $\mathbf{q} + \eta\mathbf{a}$ using first- and second-order terms presented as $\log p(\mathbf{x}|\mathbf{q}+\eta\mathbf{a}) \approx \log p(\mathbf{x}|\mathbf{q}) + \eta \frac{\partial \log p(\mathbf{x}|\mathbf{q})}{\partial \mathbf{q}}\mathbf{a} + \frac{\eta^2}{2}\mathbf{a}^T \frac{\partial^2 \log p(\mathbf{x}|\mathbf{q})}{\partial^2 \mathbf{q}}\mathbf{a}$, which derives

$$\begin{aligned} & D(\mathbf{q}||\mathbf{q} + \eta\mathbf{a}) \\ \approx & \int p(\mathbf{x}|\mathbf{q}) \left(-\eta \frac{\partial \log p(\mathbf{x}|\mathbf{q})}{\partial \mathbf{q}}\mathbf{a} - \eta^2 \mathbf{a}^T \frac{\partial^2 \log p(\mathbf{x}|\mathbf{q})}{\partial^2 \mathbf{q}}\mathbf{a}\right) d\mathbf{x}. \end{aligned} \tag{D.13}$$

We can note that the first term is zero as

$$\int p(\mathbf{x}|\mathbf{q}) \frac{\partial \log p(\mathbf{x}|\mathbf{q})}{\partial \mathbf{q}}\mathbf{a}d\mathbf{x} = \int \frac{\partial p(\mathbf{x}|\mathbf{q})}{\partial \mathbf{q}}\mathbf{a}d\mathbf{x} = 0, \tag{D.14}$$

where, $\frac{\partial \log p(\mathbf{x}|\mathbf{q})}{\partial \mathbf{q}} = \frac{1}{p(\mathbf{x}|\mathbf{q})} \frac{\partial p(\mathbf{x}|\mathbf{q})}{\partial \mathbf{q}}$. The second term is

$$\begin{aligned} & - \int p(\mathbf{x}|\mathbf{q}) \frac{\partial^2 \log p(\mathbf{x}|\mathbf{q})}{\partial^2 \mathbf{q}} d\mathbf{x} \\ & = \int p(\mathbf{x}|\mathbf{q}) \frac{\partial p(\mathbf{x}|\mathbf{q})}{\partial \mathbf{q}} \frac{\partial p(\mathbf{x}|\mathbf{q})}{\partial \mathbf{q}}^T d\mathbf{x} = \mathbf{I}(\mathbf{q}), \end{aligned} \tag{D.15}$$

where, $\mathbf{I}(\mathbf{q})$ is called the Fisher information matrix (Amari, 1998). From the above equations, we can define the metric in Riemannian form for the inverse kinematic problem as

$$D(\mathbf{q}||\mathbf{q}+\eta\mathbf{a}) \approx \frac{\eta^2}{2}\mathbf{a}^T\mathbf{I}(\mathbf{q})\mathbf{a}. \tag{D.16}$$

Therefore, the natural gradient E with the constraint that $D(\mathbf{q}||\mathbf{q}+\eta\mathbf{a}) = 1$ is given as $\mathbf{a} \propto \mathbf{I}(\mathbf{q})^{-1}J(\mathbf{q})^T(\mathbf{x}_d - \mathbf{x})$.

Next, we calculate the Fisher information metric analytically for this problem. For the case of that the observation noise is Gaussian process, it can be easily derived as

$$\begin{aligned}\mathbf{I}(\mathbf{q}) &= \int p(\mathbf{x}|\mathbf{q}) \left\{ \frac{\partial \log p(\mathbf{x}|\mathbf{q})}{\partial \mathbf{q}} \frac{\partial \log p(\mathbf{x}|\mathbf{q})}{\partial \mathbf{q}}^T \right\} d\mathbf{x} \\ &= \int p(\mathbf{x}|\mathbf{q}) \left\{ J^T\Sigma^{-1}(\mathbf{x} - f(\mathbf{q})) \left(J^T\Sigma^{-1}(\mathbf{x} - f(\mathbf{q}))\right)^T \right\} d\mathbf{x} \\ &= J^T\Sigma^{-1} \left\{ \int p(\mathbf{x}|\mathbf{q})(\mathbf{x} - f(\mathbf{q}))(\mathbf{x} - f(\mathbf{q}))^T d\mathbf{x} \right\} \Sigma^{-1}J \\ &= J^T\Sigma^{-1}J \end{aligned} \tag{D.17}$$

where,

$$\frac{\partial \log p(\mathbf{x}|\mathbf{q})}{\partial \mathbf{q}} = J^T\Sigma^{-1}(\mathbf{x} - f(\mathbf{q})) \tag{D.18}$$

Thus, the natural gradient can be obtained the following equation.

$$\mathbf{a} \propto \left(J(\mathbf{q})^T\Sigma^{-1}J(\mathbf{q})\right)^{-1} J(\mathbf{q})^T(\mathbf{x}_d - \mathbf{x}) \tag{D.19}$$

Note that in the case of that $\Sigma^{-1} = \sigma^{-1}\mathbf{I}$, the gradient (D.19) is exactly same the deterministic case presented in Eq.(D.9). Moreover, the gradient direction is also equivalent to the update direction in the Jacobian pseudo inverse algorithm. In next section, it is briefly presented and discuss the point.

D.4 The relationship between the Jacobian pseudo inverse algorithm and natural gradient method

First, we approximately calculate a velocity of the end-effector by using $\mathbf{x}_d$ and $\mathbf{x}$ as

$$\dot{\mathbf{x}}_d \approx \frac{\mathbf{x}_d - \mathbf{x}}{\Delta t}, \tag{D.20}$$

then, based on Eq.(D.2), we define the objective function:

$$E = (\dot{\mathbf{x}}_d - J(\mathbf{q})\dot{\mathbf{q}})^T(\dot{\mathbf{x}}_d - J(\mathbf{q})\dot{\mathbf{q}}). \tag{D.21}$$

The objective is to minimize E with respect to $\dot{\mathbf{q}}$. By taking the derivative with respect to $\mathbf{q}$ as

$$\frac{\partial E(\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} = 2J(\mathbf{q})^T J(\mathbf{q})\dot{\mathbf{q}} - 2J(\mathbf{q})^T \mathbf{x}_d = 0, \tag{D.22}$$

the solution can be obtained as

$$\dot{\mathbf{q}}_d = J(\mathbf{q})^+ \dot{\mathbf{x}}_d, \tag{D.23}$$

where, $J(\mathbf{q})^+ = \left(J(\mathbf{q})^T J(\mathbf{q})\right)^{-1} J(\mathbf{q})^T$ is the pseudo inverse of $J(\mathbf{q})$ (Yoshikawa, 1990). By integrating $\dot{\mathbf{q}}_d$, we can update $\mathbf{q}$ in a direction to minimize the error E *i.e.*, $\mathbf{q} \leftarrow \mathbf{q} + \Delta t \cdot \dot{\mathbf{q}}_d = \mathbf{q} + \left(J(\mathbf{q})^T J(\mathbf{q})\right)^{-1} J(\mathbf{q})^T(\mathbf{x}_d - \mathbf{x})$.

Note that the update direction presented above is exactly equivalent to the direction in Eqs.(D.9) and (D.19). Thus, we can understand the two different type of methods which are Jacobian transpose algorithm and Jacobian pseudo inverse algorithm in the same framework as the gradient method.

It may be also important to note that if Eq.(D.2) is strictly satisfied for a redundant robot, another type of pseudo inverse $J(\mathbf{q})^+ = J(\mathbf{q}) \left(J(\mathbf{q})J(\mathbf{q})^T\right)^{-1}$ can be used. This is derived as a solution of minimization of the objective $||\dot{\mathbf{q}}||$ with constraint Eq.(D.2) (Yoshikawa, 1990).

D.5 Simulations: Inverse Kinematics for a redundant robot

In order to demonstrate the validity of discussion, here we apply the natural gradient method and ordinal gradient method for a 3-link redundant manipulator robot (see (Yoshikawa, 1990)). The results are presented in Fig.D.1 and Fig.D.1, respectively. The ordinal gradient method, in which the gradient is not steepest direction, makes curved trajectories and it may cause slow convergence. On the other hand, the natural gradient method makes straight line from the initial to the target position and it may be faster convergence than the curved one.

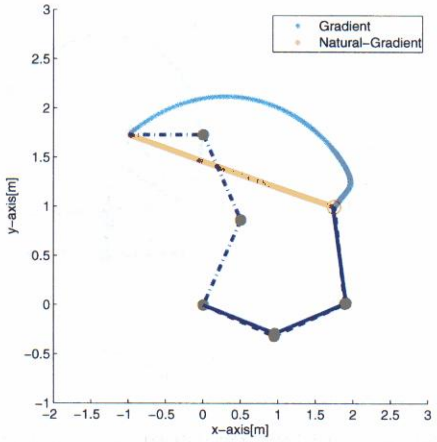


Figure D.1: Inverse kinematics problem with a three link manipulator. The blue dash-dot line means the initial posture of the manipulator ($\mathbf{q} = (\pi/3, \pi/3, \pi/3)^T$). Dashed and solid line are acquired postures by using gradient and natural gradient based methods, respectively. and the red circle means the goal state in the task space ($\mathbf{x} = (1.75, 1.0)^T$). Even though both gradient and natural gradient based method approach to the goal state, the natural gradient based method draws line trajectory which is the shortest path, but the gradient based method draws a curved trajectory.

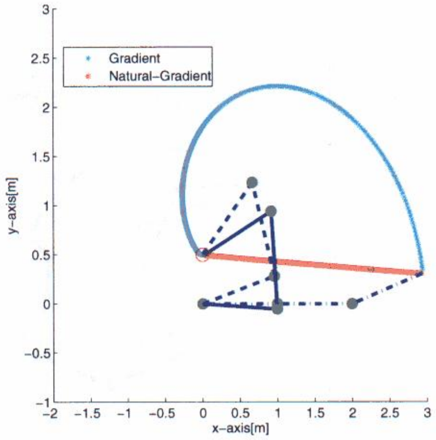


Figure D.2: Inverse kinematics problem with a three link manipulator. The blue line means the initial posture of the manipulator ($\mathbf{q} = (0.0, 0.0, \pi/10)^T$), and the red circle means the goal state in the task space ($\mathbf{x} = (0.0, 0.5)^T$).

Bibliography

Aberdeen, D., & Baxter, J. (2002). Scalable internal-state policy-gradient methods for POMDPs. In *Proceedings of the International Conference on Machine Learning*, pp. 3–10.

Aberdeen, D. A. (2003). Policy-gradient algorithms for partially observable markov decision processes. *Ph.D Thesis, Australian National University.*

Amari, S., Park, H., & Fukumizu, K. (2000). Adaptive Method of Realizing Natural Gradient Learning for Multilayer Perceptrons. *Neural Computation*, *12*(6), 1399–1409.

Amari, S. (1998). Natural Gradient Works Efficiently in Learning. *Neural Computation*, *10*(2), 251–276.

Bagnell, D., & Schneider, J. (2003). Covariant policy search. *Proceedings of the International Joint Conference on Artificail Intelligence*, *18*, 1019–1024.

Baxter, J., & Bartlett, P. L. (2001a). Experiments with infinite-horizon, policy-gradient estimation. *Journal of Artificial Intelligence Research*, *15*, 351–381.

Baxter, J., & Bartlett, P. L. (2001b). Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, *15*, 319–350.

Baxter, J., & Bartlett, P. L. (1999). Direct Gradient-Based Reinforcement Learning: I. Gradient Estimation Algorithms. *Technical report, Research School of Information Sciences and Engineering, Australian National University.*

Bellman, R. E. (2003). *Dynamic Programming.* Dover Pubns.

Bernstein, N. A., Latash, M. L., & Turvey, M. T. (1996). *Dexterity and Its Development.* Lawrence Erlbaum Associates.

Bersekas, D. P. (2000). *Dynamic Programming and Optimal Control, Vol.1.* Athena Scientific.

Bersekas, D. P. (2001). *Dynamic Programming and Optimal Control, Vol.2.* Athena Scientific.

Bertsekas, D. P. (1999). *Nonlinear Programming. Athena Scientific.*

Chalodhorn, R., Grimes, D., Maganis, G., Rao, R., & Asada, M. (2006). Learning humanoid motion dynamics through sensory-motor mapping in reduced dimensional spaces. In *IEEE International Conference on Robotics and Automation*, pp. 3693–3698.

d'Avella, A., Saltiel, P., & Bizzi, E. (2003). Combinations of muscle synergies in the construction of a natural motor behavior. *Nature Neuroscience*, *6*(3), 300–308.

Dejmal, I., & Zacksenhouse, M. (2006). Coordinative Structure of Manipulative Hand-Movements Facilitates Their Recognition. *IEEE Transactions on Biomedical Engineering*, *53*(12), 2455–2463.

Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural Computation*, *12*, 219–245.

Endo, G., Morimoto, J., Nakanishi, J., & Cheng, G. (2004). An Empirical Exploration of a Neural Oscillator for Biped Locomotion Control. In *IEEE Iternational Conference on Robotics and Automation*, pp. 3036–3042.

Fukuoka, Y., Kimura, H., & Cohen, A. H. (2003). Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain Based on Biological Concepts. *The International Journal of Robotics Reserch*, *22*(3–4), 187–202.

Greensmith, E., Bartlett, P. L., & Baxter, J. (2004). Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, *5*, 1471–1530.

Haruno, M., Wolpert, D. M., & Kawato, M. (2001). MOSAIC Model for Sensorimotor Learning and Control. *Neural Computation*, *13*, 2201–2220.

Hase, K., & Yamazaki, N. (1998). Computer Simulation of the Ontogeny of Biped Walking. *Anthropological Science, 106(4)*, 327–347.

Hirai, K., Hirose, M., Haikawa, Y., & Takenaka, T. (1998). The development of handa humanoid robot. In *IEEE International Conference on Robotics and Automation*, pp. 1321–1326.

Hyon, S., & Cheng, G. (2006). Passivity-based whole-body motion control for humanoids: Gravity compensation, balancing and walking. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 4915–4922.

Hyvarinen, A., Karhunen, J., & Oja, E. (2001). *Independent Component Analysis.* Wiley-Interscience.

Jolliffe, I. T. (1986). *Principal Component Analysis.* Springer.

Kagami, S., Kanehiro, F., Tamiya, Y., Inaba, M., & Inoue, H. (2001). AutoBalancer: An Online Dynamic Balance Compensation Scheme for Humanoid Robots. In Donald, B. R., Lynch, K., & Rus, D. (Eds.), *Algorithmic and Computational Robotics: New Directions*, pp. 329–340. A K Peters, Ltd.

Kagami, S., Kitagawa, T., Nishiwaki, K., sugihara, T., & Inaba, M. (2002). A fast dynamically equilibrated walking trajectory generation method of humanoid robot. *Autonomouns Robots*, *12*, 71–82.

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., & Hirukawa, H. (2003). Resolved momentum control: humanoid motion planning based on the linear and anguler momentum. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1644–1650.

Kakade, S. (2002). A natural policy gradient. *Advances in Neural Information Processing Systems*, *14*, 1531–1538.

Kimura, H., & Kobayashi, S. (1998). An Analysis of Actor/Critic Algorithms using Eligibility Traces: Reinforcement Learning with Imperfect Value Function. *International Conference on Machine Learning*, 278–286.

Kimura, H., Miyazaki, K., & Kobayashi, S. (1997). Reinforcement Learning in POMDPs with Function Approximation. In *Proceedings of the 14th International Conference on Machine Learning*, pp. 152–160.

Kimura, H., Yamashita, T., & Kobayashi, S. (2001). Reinforcement Learning of Walking Behavior for a Four-Legged Robot. In *Proceedings of the IEEE Conference on Decision and Control*, pp. 411–416.

Konda, V., & Tsitsiklis, J. (2003). On actor-critic algorithms. *Society for Industrial and Applied Mathematics*, *42*(4), 1143–1166.

Kotosaka, S., & Schaal, S. (2001). synchronized robot drumming by neural oscillator. *journal of the robotics society of japan*, *1*, 116–123.

Kuroki, Y., Ishida, T., Yamaguchi, J., Fujita, M., & Doi, T. (2001). A small biped entertainment robot. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 181–186.

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Letters to Nature*, *401*, 788–791.

Lee, D. D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pp. 556–562.

L.Sciavicco, & B.Siciliano (1997). *Modelling and Control of Robot Manipulators*. Springer.

Mataric, M. J. (1994). Reward functions for accelerated learning. In *Machine Learning: Proceedings of the Eleventh International Conference*, pp. 181–189.

Matsubara, T., Morimoto, J., Nakanishi, J., Hyon, S.-H., G.Hale, J., & Cheng, G. (2007). Learning to acquire whole-body CoM movements to achieve dynamic tasks. In *IEEE International Conference on Robotics and Automation*, pp. 2688–2693.

Matsubara, T., Morimoto, J., Nakanishi, J., Sato, M., & Doya, K. (2005). Learning sensory feedback to CPG for biped locomotion with policy gradient. In *IEEE International Conference on Robotics and Automation*, pp. 4175–4180.

Matsuoka, K. (1985). Sustained oscillatons generated by mutually inhibiting neurons with adaptation. *Biologial Cybernetics*, *52*, 367–376.

Mistry, M., Mohajerian, P., & Schaal, S. (2005). Arm movement experiments with joint space force fields using an exoskeleton robot. In *IEEE ninth international conference on rehabilitation robotics*, pp. 408–413.

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

Miyakoshi, S., Yamakita, M., & Furuta, K. (1994). Juggling Control Using Neural Oscillators. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 1186–1193.

Miyamoto, H., Schaal, S., Gandolfo, F., Gomi, H., Koike, Y., Osu, R., Nakano, E., Wada, Y., & Kawato, M. (1996). A kendama learning robot based on bi-directional theory. *Neural Networks*, *9*, 1281–1302.

Mori, T., Nakamura, Y., Sato, M., & Ishii, S. (2004). Reinforcement learning for a CPG-driven biped robot. In *Nineteenth National Conference on Artificial Intelligence (AAAI)*, pp. 623–630.

Morimoto, J., & Doya, K. (2001). Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning. *Robotics and Autonomous Systems*, *36*, 37–51.

Morimoto, J., Zeglin, G., & Atkeson, C. (2003). Minimax differential dynamic programming: application to a biped walking robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1927–1932.

Morimoto, J., Endo, G., Nakanishi, J., Hyon, S., Cheng, G., Bentivegna, D., & Atkeson, C. (2006). Modulation of simple sinusoidal patterns by a coupled oscillator model for biped walking. In *IEEE International Conference on Robotics and Automation*, pp. 1579–1584.

Murphy, K. (2000). A survey of POMDP solution techniques. *Technical Report, U.C. Berkeley.*

Nagasaka, K. (2000). The whole-body motion generation of humanoid robot using dynamics filter (in japanese). *Ph.D thesis, The University of Tokyo, Japan.*

Nakamura, Y., Sato, M., & Ishii, S. (2003). Reinforcement Learning for Biped Robot. In *Proceedings of the International Symposium on Adaptive Motion of Animals and Machines.*

Nakanishi, J., Cory, R., Mistry, M., Peters, J., & Schaal, S. (2005). Comparative experiments on task space control with redundancy resolution. In

Proceedings of the IEEE International Conference on Intelligent Robots and Systems, pp. 3522–3529.

Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., & Kawato, M. (2004). Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems*, *47*, 79–91.

Peters, J., Vijayakumar, S., & Schaal, S. (2003). Reinforcement learning for humanoid robotics. In *Humanoids2003, Third IEEE-RAS International Conference on Humanoid Robots*, pp. 1137–1144.

Riley, M., Ude, A., Wade, K., & Atkeson, C. G. (2003). Enabling real-time full-body imitation: a natsural way of transferring human movement to humanoids. In *IEEE International Conference on Robotics and Automation*, pp. 2368–2374.

Safonova, A., Hodgins, J. K., & Pollard, N. S. (2004). Synthesizing physically realistic human motion in low-dimensional behavior-specific spaces. In *ACM Trans.Graph*, pp. 514–521.

Sato, M., Nakamura, Y., & Ishii, S. (2002). Reinforcement Learning for Biped Locomotion. In *International Conference on Artificial Neural Networks*, pp. 777–782.

Schaal, S., & Schweighofer, N. (2005). Computational motor control in humans and robots. *Current Opinion Neurobiol*, *6*, 675–82.

Scholz, J., & Schoner, G. (1999). The uncontrolled manifold concept: identifying control variables for a functional task. *Experimental Brain Research*, *126*, 289–306.

Sentis, L., & Khatib, O. (2005). Control of free-floating humanoid robots through task prioritization. In *Proceedings of the IEEE International Conference in Robotics and Automation*, pp. 1718– 1723.

Singh, S., Jaakkola, T., & Jordan, M. (1994). Learning Without State-Estimation in Partially Observable Markovian Decision Processes. In *In Machine Learning: Proceedings of the Eleventh International Conference*, pp. 284–292.

Soechting, J. F., & Flanders, M. (1997). Flexibility and Repeatability of Finger Movements During Typing: Analysis of Multiple Degrees of Freedom. *Journal of Computational Neuroscience*, *4*(1), 29–46.

Strogatz, S. H. (1994). *Nonlinear Dynamics and Chaos.* Westview press.

Sugihara, T., & Nakamura, Y. (2002). Whole-body cooperative balancing of humanoid robot using COG jacobian. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2575–2580.

Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (2000). Policy Gradient Methods for Reinforcement Learning with Function Approximation. *Advances in Neural Information Processing Systems*, *12*, 1057–1063.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*, 9–44.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction.* MIT Press.

Taga, G., Yamaguchi, Y., & Shimizu, H. (1991). Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, *65*, 147–159.

Tedrake, R., Zhang, T. W., & Seung, H. S. (2004). Stochastic Policy Gradient Reinforcement Learning on a Simple 3D Biped. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 2849–2854.

Tesauro, G. (1994). TD-Gammon, a self teaching backgammon program, achieves master legel play. *Neural Computation*, *6*, 215–219.

Tresch, M. C., Cheung, V. C. K., & d'Avella, A. (2006). Matrix factorization algorithms for the identification of muscle synergies: Evaluation on simulated and experimental data sets. *Journal of Neurophysiology*, *95*, 2199–2212.

Tsitsiklis, J. N., & Roy, B. V. (1996). An analysis of temporal-difference learning with function approximation. Tech. rep. LIDS-P-2322.

Tsuchiya, K., Aoi, S., & Tsujita, K. (2003). Locomotion Control of a Biped Locomotion Robot using Nonlinear Oscillators. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 1745–1750.

Tuchiya, C., Kimura, H., & Kobayashi, S. (2004). Policy learning by GA using importance sampling. In *The 8th Conference on Intelligent Autonomous Systems*, pp. 281–290.

Vukobratović, M., & Borovac, B. (2004). Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics.*

Watkins, C., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3), 279–292.

Weaver, L., & Tao, N. (2001). The optimal reward baseline for gradient-based reinforcement learning. In *Uncertainty in Artificial Intelligence: Proceedings of the Seventheenth Conference*, pp. 538–545.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, *8*, 229–256.

Williamson, M. M. (1998). Neural control of rhythmic arm movements. *Neural Networks*, *11*(7-8), 1379–1394.

Yoshikawa, T. (1990). *Foundations of robotics: analysis and control.* MIT Press.

Yoshimoto, J., Nishimura, M., Tokita, Y., & Ishii, S. (2005). Acrobot control by learning the switching of multiple controllers. *Journal of Artificial Life and Robotics*, *9*, 67–71.