

)

# Progressive Learning and Its Application to Robotic Assembly

by

**Boo-Ho Yang**

B.S. in Applied Mathematics and Physics  
Kyoto University, Japan, 1988

M.S.M.E., Massachusetts Institute of Technology, 1990

Submitted to the Department of Mechanical Engineering  
in Partial Fulfillment of the Requirements for the Degree of

**DOCTOR OF PHILOSOPHY**

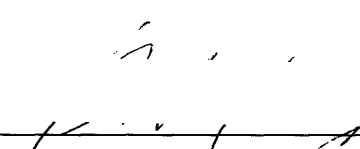
at the

**MASSACHUSETTS INSTITUTE OF TECHNOLOGY**

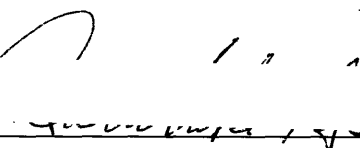
February 1995

©Massachusetts Institute of Technology 1995  
All rights reserved

Signature of Author \_\_\_\_\_

  
Department of Mechanical Engineering  
February, 1995

Certified by \_\_\_\_\_

  
Haruhiko Asada  
Professor of Mechanical Engineering  
Thesis Supervisor

Accepted by \_\_\_\_\_

  
Ain A. Sonin  
Chairman, Department Committee on Graduate Students  
MASSACHUSETTS INSTITUTE  
OF TECHNOLOGY

SEP 21 1995 Barker Eng

# **Progressive Learning and Its Application to Robotic Assembly**

Boo-Ho Yang

Submitted to the Department of Mechanical Engineering  
on February 10, 1995  
in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy

## **Abstract**

Human learning is by far effective than machine learning in many ways. Human uses a curriculum, an organized set of materials to be learned in a particular sequence. Inspired by this human learning, we have developed a new method of machine learning by designing a series of input tasks, referred to as "curriculum," so that machine learning can be performed smoothly, quickly and stably without incurring fatal mistakes. The new method termed "Progressive Learning" uses scheduled excitation inputs that allow the system to learn quasi-static, slow modes in the beginning, followed by the learning of faster modes. We first present a theory of progressive learning by formulating a gradient based, model reference adaptive control problem. It is well known that in a model reference adaptive control system an excitation at a high frequency causes instability to the system when the relative order of the plant is high and the SPR(strictly positive real) condition is not met. To derive a stability analysis for progressive learning, we apply a method of averaging analysis to describe the behavior of the adaptive system in the frequency domain. Based on this analysis, we prove that the stable convergence of control parameters is guaranteed if the system is excited gradually through the reference input in accordance to the progress of the adaptation. A numerical example is provided to verify the above analysis. The concept of progressive learning is next applied to robotic assembly to explore the possibility of progressive learning. A high speed insertion task is used as an example, where an impedance control law is learned with the excitation scheduling method. In this method, learning starts with a slow, quasi-static motion and goes to a fast, dynamic motion. During the learning process, the stiffness terms of the impedance controller are learned first, followed by the damping terms and finally by the inertial terms. Consequently, this progressive learning method enables the learning of high-speed dynamic control laws without instability and fatal damage due to high speed collisions. The mechanism of progressive learning is also discussed in detail and verified through simulation experiments.

Thesis Supervisor: Haruhiko Asada

Title: Professor of Mechanical Engineering

Dedicated to my parents

## Acknowledgments

---

Life at MIT is a hell, but the hell could be endurable and, sometimes, even enjoyable. To me, the six years I have spent at MIT is the most valuable period of time in my life, and that was possible only with those to whom I owe many thanks.

Among them, I would like to express my sincerest thanks to my thesis supervisor, Professor Haruhiko Asada. Throughout the years, he has given me inspiring advice and constant encouragement. I benefited greatly from his keen insight and constant efforts to seek innovative research directions. I also wish to express my hearty thanks to Mrs. Kumiko Asada for being so considerate and supportive over the years.

My thesis committee members, Professor Anuradha M. Annaswamy, Professor Michael I. Jordan, and Dr. J. Kenneth Salisbury have also greatly contributed to the direction and quality of this thesis work. Especially, I would like to thank Professor Anuradha M. Annaswamy for her helpful advice and discussions on adaptive control. Without her help, this thesis could not be completed.

Special thanks go to Leslie M. Regan for her administrative help and support.

I am grateful to all of my colleagues and ex-colleagues in the Intelligent Machines Laboratory. Especially, I would like to thank Sheng Liu, Xiangdon He, Sean Li, Sooyong Lee, Susan Ipri, Anton Pil, Ming Zhou, Kevin Brown, Cliff Federspiel, Brennan MaCarragher, Danny Braunstein, Mark West, Jahng Park, Toshihiko Koyama and Masakazu Nakashima. The work and life at MIT would not have been so valuable and enjoyable without the friendship and cheer from the lab group. I also wish to express my special thanks to Dr. Kazuhiro Kosuge, Kenichiro Shimokura, Kenji Okamoto, Noriaki Yoshida, Noriyuki Fujiwara, Akihira Nishikawa, Atsushi Toizumi, Yoshihiro Sasage and their families for their tender care and needed support.

I wish to thank the close friends that I made at MIT, David Inouye, Lucy Jen, Masahiko Ikuta, Toshiya Sugimoto, Kenji Shimada, Takashi Maekawa, and many others, for being available whenever I need their support most. Especially, David and Ikuta, I will not forget your constant supply of coffee, food and drink over the years. Some people say that the "help" deteriorated my productivity, though.

I would like to express my sincere appreciation to Mr. Poo Doo-Ok for his moral and financial support when I needed it most.

And last, I would like to express my deepest thanks to my parents for their dedicated love and support.

# Contents

---

<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	Background . . . . .	9
1.2	Objectives . . . . .	11
1.3	Outline of the Thesis . . . . .	12
<b>2</b>	<b>The Basic Concept of Progressive Learning</b>	<b>14</b>
<b>3</b>	<b>Theory of Progressive Learning</b>	<b>17</b>
3.1	Introduction . . . . .	17
3.2	Statement of the Problem . . . . .	19
3.3	Stability Analysis using Averaging . . . . .	25
3.4	Input Design in Frequency Domain . . . . .	29
3.4.1	Frequency Range for Stability . . . . .	29
3.4.2	Transfer Function Matching in Frequency Domain . . . . .	31
3.4.3	Progressive Excitation . . . . .	32
3.5	Simulation . . . . .	41
3.6	Conclusions . . . . .	48
<b>4</b>	<b>Progressive Learning for Robotic Assembly: Application to a Non-linear Process</b>	<b>49</b>
4.1	Introduction . . . . .	49
4.2	High Speed Insertion Task . . . . .	50
4.2.1	Formulation of the Problem . . . . .	50

4.2.2	The Progressive Learning Approach: Accommodating the Task Complexity Level by Motion Speed Scheduling . . . . .	53
4.3	Mechanisms of Progressive Learning . . . . .	55
4.3.1	Gradient Following . . . . .	56
4.3.2	Local, Progressive Learning of Internal Model . . . . .	59
4.4	Implementation and Simulation . . . . .	63
4.4.1	Implementation . . . . .	64
4.4.2	Simulation Results . . . . .	65
4.4.3	Verifying the Features of Progressive Learning . . . . .	71
4.5	Strategies for Increasing the Motion Speed . . . . .	74
4.6	Conclusion . . . . .	79
<b>5</b>	<b>Conclusions</b>	<b>80</b>
5.1	Thesis Summary . . . . .	80
5.2	Contributions . . . . .	82
5.2.1	Theory of Progressive Learning . . . . .	82
5.2.2	Application to Robotic Assembly . . . . .	82
	<b>Appendix A Derivation of the Sensitivity Vector</b>	<b>84</b>
	<b>Appendix B Derivation of the Covariance Matrix</b>	<b>86</b>
	<b>References</b>	<b>89</b>

# List of Figures

---

- 1.1 Conceptual diagram of progressive learning system . . . . . 10
- 2.1 Progressive learning system . . . . . 16
- 3.1 Model Reference Adaptive Controller . . . . . 21
- 3.2 Frequency range of stability for a given  $\theta$  . . . . . 31
- 3.3 Frequency domain matching with partially exciting reference input . . . 33
- 3.4 Progressive expansion of the stability range . . . . . 40
- 3.5 Initial phase shifts of  $\Phi_\theta(s)$  and  $\Phi_m(s)$  . . . . . 43
- 3.6 Results of the non-progressive excitation . . . . . 44
- 3.7 Output error and parameter curves with progressive excitation . . . . . 46
- 3.8 Progressive changes of phase angle curves . . . . . 47
- 4.1 Schematic diagram of impedance control . . . . . 52
- 4.2 Schematic diagram of Gradient Following Approaches . . . . . 57
- 4.3 Model Based Reinforcement Learning . . . . . 61
- 4.4 Task environment of the ball insertion . . . . . 65
- 4.5 Learning histories of reinforcement  $r$  and stiffness  $K$ , damping  $D$  and  
inertia matrices  $M$  over 300 iterations . . . . . 66
- 4.6 Task performances before learning (a) and after 100 learning trials (b) . 69
- 4.7 Task performance at high speed before learning inertia  $M$  (a) and after  
(b) . . . . . 70
- 4.8 Comparison between progressive and non-progressive learning methods 71

4.9	Comparison of the magnitudes of errors and reaction force between progressive learning and non-progressive learning . . . . .	73
4.10	Comparison of $\phi$ between progressive learning (a) and non-progressive learning (b) . . . . .	75
4.11	Distribution of sample data in $\Delta x - \Delta v_x$ plane . . . . .	76
4.12	Learning histories of reinforcement $r$ and stiffness $K$ , damping $D$ and inertia matrices $M$ over 300 iterations with a smaller motion speed increment. Broken lines shows the results of the previous progressive learning simulation. . . . .	77
4.13	Learning histories of reinforcement $r$ and stiffness $K$ , damping $D$ and inertia matrices $M$ with the new strategy of motion speed scheduling .	78



# Chapter 1

## Introduction

---

### 1.1 Background

When people learn new tasks, they are slow and meticulous at the beginning but speed up the operations as they gain experience and skills. We know that we had better attempt simplified tasks at the beginning in order to become familiar with the tasks and then, based on the knowledge and experience gained, execute the full-scale tasks. Carrying out complex tasks in haste may incur costly failures and damage, or lead to confusion in which case the learner can gain neither useful information nor valuable experience. Such confusing results and failures have no pedagogical value for learning the task. There is an old saying that, we learn more from failure than success, but this is true only when the learner has enough competence to interpret the results and correlate these with possible causes. Depending on the competence and amount of knowledge the learner possesses at the beginning, judicious choices are required to determine the level of task complexity appropriate for initial attempts.

In this thesis, we will explore a new approach to learning, inspired by this human learning behavior. The level of task complexity will be increased gradually as the learner gains knowledge and improves task performance. To this end, we need to organize the learning process by evaluating the learner and determining the level of task complexity appropriate for the learner, so that the learner can attain useful knowledge with a minimum of confusion and failure. Figure 1.1 shows a conceptual diagram of this learning system consisting of a learning organizer, called a “pedagogue”, and a learner. The learner, who has its own learning mechanism, is provided by the

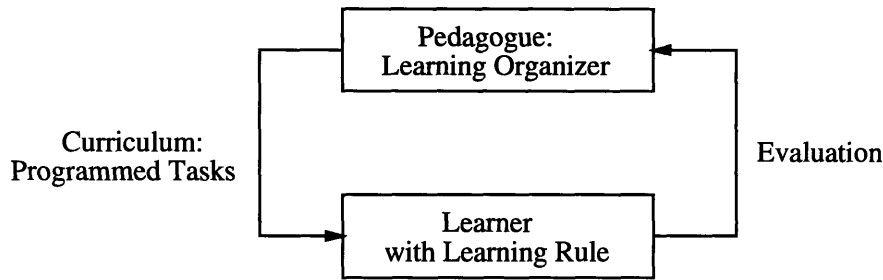


Figure 1.1: Conceptual diagram of progressive learning system

pedagogue with a series of programmed tasks, called a “curriculum”. Designing this learning system is thus two-fold: the design of the learning rule for the learner and the design of tasks, i.e. the curriculum.

This task design problem is relatively new to the learning/adaptive control community, but related issues have been addressed in different areas. In experimental psychology, the idea of designing approximated tasks to facilitate learning, referred to as “shaping”, has been used for animal training for decades ([Honig and Staddon, 1977] for example). In the animal training, however, the design of tasks has to rely on a trainer’s intuitive understanding of the way that the animal behavior is generated, and no general formulation of shaping the tasks has been addressed in the psychology literature. Recently, the concept of shaping has been applied to artificial neural networks, in which successive presentations of sample data to a network have been studied. [Allen, 1989], for example, trained recurrent networks by presenting short sequences first, followed by longer sequences over time. [Gullapalli and Barto, 1990] applied the concept of shaping to accelerate learning for a key-pressing task. In this application, the key-pressing task is divided into a sequence of subtasks such as raising, positioning and pressing the fingertip, and the control action is evaluated and

reinforced separately at each subtask.

In adaptive control, on the other hand, the design of reference signals has been a central issue for the stable convergence of system parameters. In particular, it has been revealed that stability conditions depend not only on dynamic properties of the plant but also on the excitation level of the reference signal. For example, [Riedle and Kokotovic, 1985] obtained a sharp local stability-instability boundary in terms of the frequency content of the reference signals. They call this “signal dependent positivity condition.” [Åström, 1984] examined the instability mechanism of the well-known counter-example [Rohrs, et al., 1982] in the frequency domain. [Sastry and Bodson, 1989] provided an extensive analysis for the relationship between the frequency contents of the reference input and the convergence rate of adaptive systems. One interpretation of these stability analyses is that an adaptive system may be able to avoid instability if the excitation level of the reference input is maintained sufficiently low even in a case where instability would occur otherwise. However, the system would fail to be fully excited for such a low level input and, as a result, the system parameters cannot converge to their true values. Therefore, it is quite important to develop a strategy that allows the system to be fully excited yet in a stable manner.

## 1.2 Objectives

The objective of this thesis is to explore a novel learning method in which the level of task complexity advances progressively in accordance with the learner’s competence and level of accomplishment. In the learning method, termed “Progressive Learning,” we design a series of tasks with different complexity levels appropriate for the learner. The idea is to integrate the task assignment scheduling with the design of the total control algorithm. It is expected that by integrating these we will be able to

avoid instability and divergence of learning, expedite the learning process, and maintain a desired task performance level having a minimum chance of failure and damage to the system. To prove the above arguments, the theory underpinning progressive learning is derived by formulating a model reference adaptive control problem. A thorough analysis of the behavior of progressive learning is presented.

### **1.3 Outline of the Thesis**

This thesis is composed of two main parts: theory and application of progressive learning, presented in Chapter 3 and Chapter 4 respectively. Although these parts considerably share the same concept of progressive learning, the developments in each of the chapters are fairly self-contained.

In Chapter 2, we first present the basic concept and the definition of progressive learning.

In Chapter 3, we present a theory of progressive learning by formulating a gradient based, model reference adaptive control problem. A stability analysis for progressive learning is derived by applying a method of averaging analysis. In the analysis, the behavior of the adaptive system is described in the frequency domain. Based on this analysis, we prove that the stable convergence of control parameters is guaranteed if the system is excited gradually through the reference input in accordance to the progress of the adaptation. A numerical example is provided to verify the above analysis.

In Chapter 4, the concept of progressive learning is applied to robotic assembly to explore the possibility of progressive learning. A high speed insertion task is used as an example, where an impedance control law is learned with the excitation scheduling method. In this method, learning starts with a slow, quasi-static motion and goes to a fast, dynamic motion. During the learning process, the stiffness terms of the

impedance controller are learned first, followed by the damping terms and finally by the inertial terms. Consequently, this progressive learning method enables the learning of high-speed dynamic control laws without instability and fatal damage due to high speed collisions. The mechanism of progressive learning is also discussed in detail and verified through simulation experiments. Different strategies for varying motion speeds to expedite the learning process are also addressed at the end.

Conclusions are given in Chapter 5.

## Chapter 2

# The Basic Concept of Progressive Learning

---

To apply learning control to practical processes, stability must be guaranteed. Learning algorithms that cannot be guaranteed to converge are not acceptable or feasible for practical use. Moreover, in most applications, a certain minimum level of task performance must be accomplished at all times, even at an early stage of the learning process. Once the system is engaged in an actual task, it must not fail in performing the task, nor yield poor outputs. For example, industrial robots in a factory production line must always be able to perform a given task within a tolerance error. Learning algorithms, although guaranteed to converge in theory, may not be applicable to practical tasks if task performance during the learning process is not satisfactory. Particularly difficult is the early stage of learning when the system does not have enough data or exact knowledge about the task.

Humans perform unfamiliar tasks slowly and meticulously when their knowledge is limited and stringent task specifications must be met. By reducing speed, for example, they make the task tractable and executable despite limited knowledge and skills. The required level of task performance is compared with their competence to perform the task, and, if difficult to execute, the task complexity is lowered by reducing execution speed, relaxing some conditions, or limiting the scope of the task. As humans gain experience and become familiar with the new task, they increase the task execution speed, or attempt to deal with tougher conditions and a broader range of situations. People continue to learn the task by progressively increasing the level of task complexity thereby improving the task performance ability. Judicious judgments must be made in these steps, and an effective strategy must be set up to learn the

task while executing it satisfactorily.

Assigning an appropriate task level that matches the learner's competence is an important feature in human learning. In the early stage of learning, we should first lower the task complexity in order to avoid failure and unsatisfactory performance, and then increase the complexity level in accordance with the progress in learning and the improvement in task performance. If this strategy is successfully applied to machine learning, we will be able to overcome the difficulties of traditional learning methods as described earlier. In order to learn a task while executing the task by satisfying minimum task requirements even in an early stage, learning and task execution must be coordinated. In the traditional framework of learning control, the major research interest is focused on the development of learning rules and their convergence conditions. In the proposed approach, we do not address a learning rule alone, but we integrate it with the synthesis of learning schedule in which the task complexity level is varied depending on the learning progress. This new learning method, which we refer to as "Progressive Learning", is defined as follows:

Progressive Learning is a learning method in which the level of task complexity is gradually increased in accordance with the progress of learning so that minimum task performance requirements can be met throughout the learning process and that the learning process may not diverge as the level of task complexity increases.

Progressive learning is a dynamic process, since task assignments vary dynamically during the learning process. As shown in Figure 2, the system consists of a learner, a task process or a plant, a performance evaluator, and a learning scheduler. The learning scheduler determines the task complexity level appropriate for the learner on the basis of the task performance evaluation. In consequence, there are two feedback

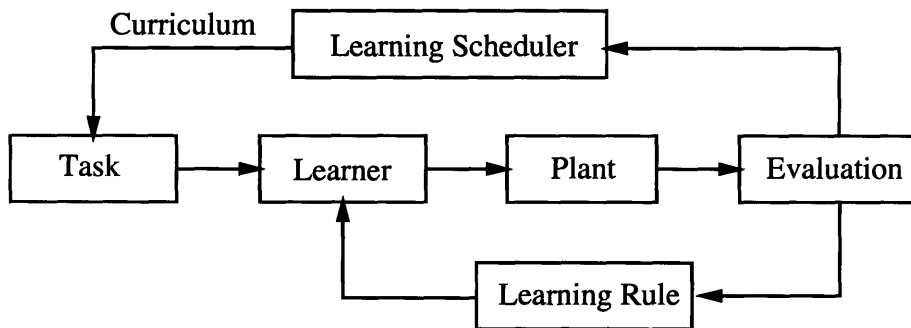


Figure 2.1: Progressive learning system

loops involved in the progressive learning system: the task assignment loop and the standard learning control loop. In designing this progressive learning system, we need to elaborate both the schedule of the task assignments and the learning rule in an integrated and cohesive manner. There is a significant synergism between the two that improves stability and convergence speed along with practical aspects, as will be explored later.



## Chapter 3

# Theory of Progressive Learning

---

### 3.1 Introduction

Since its introduction to model reference adaptive control by [Parks, 1966], the concept of SPR (Strictly Positive Real) has been playing the major role in developing various stable algorithms for adaptive systems, e.g. [Narendra, et al., 1980] [Anderson, 1986]. The applicability of the SPR approach, however, has been limited to systems with relative degree less than two. For example, as often seen in mechanical positioning systems, non-collocated sensor/actuator systems inherently have a high relative order, and, therefore, it has been difficult to develop a stable learning technique for the non-collocated systems. Although a variety of techniques have been presented in order to overcome the limitation to low relative degree systems [Narendra and Annaswamy, 1989] [Kokotović, et al., 1992], it is still the major obstacle for generalizing the SPR condition to a broader class of adaptive systems.

On the other hand, the stability analysis using the averaging techniques has revealed that the stability conditions depend not only on the positive realness of the system but also on the frequency contents of the system's internal signals. For example, [Riedle and Kokotovic, 1985] obtained a sharp local stability-instability boundary in the frequency domain by linearizing the system. They call this "signal dependent positivity condition." [Åström, 1984] applied the averaging theory to examine the instability mechanism of the well-known counter-example by [Rohrs, et al., 1982] in the frequency domain. [Sastry and Bodson, 1989] provided an extensive analysis for the relationship between the frequency contents of the reference input and the conver-

gence rate of adaptive systems. One interpretation of these stability analyses in the frequency domain is that an adaptive system can avoid instability even with relative order more than one if the frequencies of the reference input are maintained sufficiently low. However, the absence of high frequency contents in the reference input fails to satisfy the persistent excitation conditions and, as a result, the system parameters cannot converge to their true values. Therefore, it is quite important to provide the reference input that meets both the signal dependent positivity condition and the persistent excitation condition so that stable, efficient convergence can be guaranteed even for a system with a high relative order.

In this chapter, we present a new input design method for stable adaptive control of complex systems with high relative orders based on the concept of progressive learning. The key idea is to gradually excite the system by providing a particular sequence of reference inputs that consists of all low frequency contents in the beginning and that consists of full frequency contents at the end. The progressive learning method allows the system to learn parameters recursively and progressively, starting with the ones associated with low frequencies and moving up to the ones with a full spectrum. For each step of learning, the frequency contents of the reference input are carefully selected from a limited range of frequency so that the signal dependent positivity condition is met in order to avoid instability. Since the persistent excitation conditions are not met, the system parameters may converge to certain values different from the true ones. However, the stability analysis in the frequency domain to be provided in this paper will show that the partial convergence of the parameters broadens the frequency range in which stability is guaranteed. As a result, the system can be excited at higher frequencies than that of the previous learning phase. After repeating this procedure of designing the reference input, the frequency range for the positivity

condition reaches the whole frequency range and, as a result, the last reference input in the sequence excites the system persistently for the parameter convergence to the true values while avoiding the instability.

In this chapter, we present this progressive learning approach in the context of model reference adaptive control. We first obtain stability conditions that relate frequency contents of the reference input to the stability property of the system. Based on this stability analysis, we prove the main theorem: the existence of a sequence of reference inputs that achieve the progressive convergence of the control parameters for the adaptive control system. A system with relative degree of 3 is used as an exemplary case study and all the arguments and analyses are verified through simulation.

## 3.2 Statement of the Problem

In this section, we consider a model reference adaptive control(MRAC) scheme of the type treated in standard textbooks (e.g., [Narendra and Annaswamy, 1989]). The plant to be controlled is linear and time-invariant with input  $u \in \mathbb{R}$  and output  $y_p \in \mathbb{R}$  which are related by

$$y_p = W_p(s)u \quad (3.1)$$

where  $W_p(s) = k_p(Z_p(s)/R_p(s))$  is the transfer function of the plant. The reference model to be followed is linear and time-invariant with input  $r \in \mathbb{R}$  and output  $y_m \in \mathbb{R}$  which are related by

$$y_m = W_m(s)r \quad (3.2)$$

where  $W_m(s) = k_m(Z_m(s)/R_m(s))$  is the transfer function of the reference model. The objective of control is to find a differentiator-free control law  $u(t)$  such that the output error

$$e_1 = y_p - y_m \quad (3.3)$$

converges to zero asymptotically for arbitrary initial conditions and arbitrary piecewise continuous, uniformly bounded reference signals  $r(t)$ .

To meet the control objective, we make the following standard assumptions concerning the plant  $W_p(s)$  and the reference model  $W_m(s)$ :

- (A1)  $R_p(s)$  is a monic polynomial of known degree  $n$ ,
- (A2)  $Z_p(s)$  is a monic Hurwitz polynomial of known degree  $m < n$ ,
- (A3) The sign of  $k_p$  is known,
- (A4)  $Z_m(s)$  and  $R_m(s)$  are monic Hurwitz polynomials of degree  $m$  and  $n$  respectively.

We also add an assumption on the reference input  $r$  as

- (A5)  $r$  has an autocovariance.

In what follows,  $s$  denotes either the Laplace variable or the differential operator.

### A. Control Structure

The control scheme proposed by [Narendra and Annaswamy, 1989] is shown in Figure 3.1. The controller is described completely by the following differential equations and definitions:

$$\dot{w}_1 = \Lambda w_1 + l u \quad (3.4)$$

$$\dot{w}_2 = \Lambda w_2 + l y_p \quad (3.5)$$

$$w \stackrel{\text{def}}{=} [r, w_1^T, y_p, w_2^T]^T \quad (3.6)$$

$$\theta \stackrel{\text{def}}{=} [k, \theta_1^T, \theta_0, \theta_2^T]^T \quad (3.7)$$

$$u = \theta^T w \quad (3.8)$$

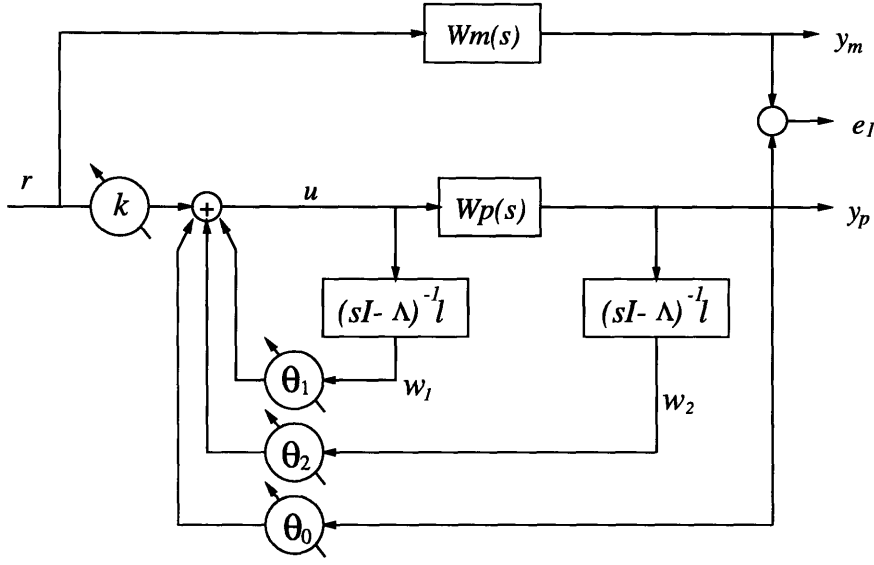


Figure 3.1: Model Reference Adaptive Controller

where  $\theta_1, \theta_2, w_1, w_2 \in \mathbb{R}^{n-1}$ ,  $k, \theta_0 \in \mathbb{R}$ , and  $(\Lambda, l)$  is an asymptotically stable system in controllable canonical form with

$$\lambda(s) \stackrel{\text{def}}{=} \det(sI - \Lambda) = \lambda_0(s)Z_m(s) \quad (3.9)$$

for some monic Hurwitz polynomial  $\lambda_0$  of degree  $n - m - 1$ .

Assuming that the control parameters are constant, the transfer functions of the feedforward and the feedback controllers can be expressed respectively

$$\frac{\lambda(s)}{\lambda(s) - C(s)} \quad \text{and} \quad \frac{D(s)}{\lambda(s)}$$

where

$$\frac{C(s)}{\lambda(s)} = \theta_1^T (sI - \Lambda)^{-1} l, \quad (3.10)$$

$$\frac{D(s)}{\lambda(s)} = \theta_0 + \theta_2^T (sI - \Lambda)^{-1} l, \quad (3.11)$$

and the overall transfer function of the plant together with the controller can be

expressed as

$$W_\theta(s) = \frac{kk_p Z_p(s)\lambda(s)}{(\lambda(s) - C(s))R_p(s) - k_p Z_p(s)D(s)}. \quad (3.12)$$

From this transfer function, the closed-loop characteristic function can be given as

$$\Phi_\theta(s) = (\lambda(s) - C(s))R_p(s) - k_p Z_p(s)D(s). \quad (3.13)$$

The transfer function from the reference input  $r$  to the regressor vector  $w$  with a constant parameter vector  $\theta$  is also derived as

$$H_{wr}(s, \theta) = \begin{bmatrix} 1 \\ (sI - \Lambda)^{-1}lW_p^{-1}W_\theta \\ W_\theta \\ (sI - \Lambda)^{-1}lW_\theta \end{bmatrix}. \quad (3.14)$$

### B. Nominal Representation of Reference Model

It is well known that under the above assumptions and control structure there exists a unique constant vector  $\theta^*$  such that the closed-loop transfer function  $W_{\theta^*}(s)$  matches  $W_m(s)$  exactly. Namely, we can express the reference model as the plant  $W_p(s)$  with the same controller at  $\theta = \theta^*$ . In this representation, the regressor vector  $w_m$  is given by

$$w_m = [r, w_{m1}^T, y_m, w_{m2}^T]^T. \quad (3.15)$$

Let  $\Phi_m(s)$  be the model characteristic function, that is, the closed-loop characteristic function when  $\theta = \theta^*$ , and it can be derived that

$$\Phi_m(s) = Z_p(s)\lambda_0(s)R_m(s). \quad (3.16)$$

Important to note is that for a given constant parameter vector  $\theta$  the closed-loop transfer function is expressed as follows:

$$W_\theta(s) = \frac{kk_p Z_p \lambda_0 Z_m}{kk_p \Phi_\theta} = \frac{\Phi_m Z_m}{\Phi_\theta R_m} = \frac{k}{k^*} \frac{\Phi_m(s)}{\Phi_\theta(s)} W_m(s) \quad (3.17)$$

where  $k^* = k_m/k_p$ , the nominal value of  $k$ . Also note that  $w_m$  is the output of a stable linear time invariant system driven by  $r(t)$  and its transfer function is

$$H_{w_m r} = H_{wr}(s, \theta^*) = \begin{bmatrix} 1 \\ (sI - \Lambda)^{-1} l W_p^{-1} W_m \\ W_m \\ (sI - \Lambda)^{-1} l W_m \end{bmatrix}. \quad (3.18)$$

### C. Output Error Dynamics

Let us define the parameter error vector as

$$\phi \stackrel{\text{def}}{=} \theta - \theta^*. \quad (3.19)$$

From the above equations, the dynamics of the output error  $e_1$  can be easily derived as

$$e_1 = \frac{1}{k^*} W_m(s) \phi^T w. \quad (3.20)$$

### D. Adaptation Rule

The objective of adaptation is to make the parameter error as well as the output error asymptotically converge to zero. For the above formulation, a so-called SPR rule such as

$$\dot{\phi}(t) = \dot{\theta}(t) = -e_1(t)w(t) \quad (3.21)$$

guarantees the overall stability of the adaptive system with persistently exciting signals, provided that  $W_m(s)$  is strictly positive real (SPR) (e.g., [Narendra and Annaswamy, 1989]). However, this adaptation rule cannot guarantee the stability for plants with high relative degree. It is known that instability may occur with a SPR rule if the regressor vector  $w(t)$  is excited at a high frequency (e.g., [Kokotović, et al., 1985]).

In this paper, we use the gradient descent rule, often referred to as the MIT rule, for adaptation. The idea of MIT rule is to reduce  $e_1^2$  by adjusting  $\theta$  along the direction

of steepest descent. Namely, the MIT rule can be expressed as

$$\dot{\phi}(t) = -\frac{1}{2}\alpha \frac{\partial e_1^{2T}}{\partial \phi} = -\alpha e_1 \frac{\partial e_1^T}{\partial \phi} \quad (3.22)$$

where  $(\partial e_1/\partial \phi)^T$  is the sensitivity vector denoted by  $\psi(t)$ , and can be derived as

$$\psi(t) = \frac{\partial e_1^T}{\partial \phi} \quad (3.23)$$

$$= \bar{W}_\theta(s)w \quad (3.24)$$

where  $\bar{W}_\theta = W_\theta/k$ , that is,

$$\bar{W}_\theta(s) = \frac{k_p Z_p(s)\lambda(s)}{(\lambda(s) - C(s))R_p(s) - k_p Z_p(s)D(s)}. \quad (3.25)$$

The derivation of the above equation is provided in Appendix A.

It has been empirically and analytically shown that the closed-loop stability of the MIT rule depends on the adaptation gain and the magnitude of the reference signal. It has also been shown that the MIT rule cause instability depending on the initial values of the control parameters. In other words, the MIT rule may cause instability even for a simple plant for which the stability can be guaranteed with the SPR rule, although a complete stability analysis has not been available yet. The objective of this paper is to show that the adaptive system can be stabilized even with the MIT rule if the system is excited progressively by changing the frequency content of the reference input according to the progress of the adaptation. In the following sections, we first derive a stability condition in the frequency domain for the MIT rule and prove that the stability of the adaptive system depends on the frequency content of the reference signal as well as the values of the control parameters. Based on the stability analysis, we prove that there always exists a sequence of reference inputs that guarantee the stability for a plant with a high relative order.



### 3.3 Stability Analysis using Averaging

Averaging is an asymptotic method that allows the analysis of dynamic behavior of a nonautonomous (time varying) system through an autonomous (time invariant) system obtained by time-averaging of the original system. The averaging method was originally proposed by [Bogoliuboff and Mitropolskii, 1961], and further developed by [Sethna, 1973] and [Hale, 1980], to name a few. Averaging methods were then successfully developed for the stability analysis of adaptive systems by [Åström, 1984], [Riedle and Kokotovic, 1985] and [Anderson, 1986]. An extensive review and useful averaging theorems for adaptive systems are found in [Sastry and Bodson, 1989]. In [Sastry and Bodson, 1989], the characterization of the asymptotic stability of the adaptive systems was addressed through averaging analysis for systems with two time scales.

The objective of this section is to examine the convergence of control parameters for the adaptation rule proposed in the previous section. In this section, we first apply the two-time scale averaging analysis given in [Sastry and Bodson, 1989] to the proposed adaptive system and next derive a stability condition in terms of frequencies of the reference input.

The dynamics equation of the control parameters is given from eqs.(3.20), (3.22) and (3.24) as follows:

$$\dot{\phi}(t) = -\frac{\alpha}{k^*} W_m \phi^T w \bar{W}_\theta w. \quad (3.26)$$

To apply the averaging method, we need to treat the above equation as a slow adaptation process. For the purpose, we introduce an additional assumption:

- (A6) the adaptation gain  $\alpha$  is sufficiently small, that is, the variations of  $\phi$  are slow compared with those of  $e_1$ .

With this assumption, we can separate the slow time scale of the control parameters from the fast time scale of the other signals. By applying the averaging method given in [Sastry and Bodson, 1989], we can approximate the original system in eq.(3.26) by using an averaged system as

$$\dot{\phi}_{av} = -\frac{\alpha}{k^*} \left[ \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t_0}^{t_0+T} W_m \phi_{av}^T w \bar{W}_\theta w dt \right] \quad (3.27)$$

$$= -\frac{\alpha}{k^*} \left[ \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t_0}^{t_0+T} \bar{W}_\theta w W_m w^T dt \right] \phi_{av}. \quad (3.28)$$

Defining  $w_f \stackrel{\text{def}}{=} W_m w$  and  $w_\theta \stackrel{\text{def}}{=} \bar{W}_\theta w$ , and assuming the cross correlation between these two exists, we obtain

$$\dot{\phi}_{av} = -\frac{\alpha}{k^*} \left[ \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t_0}^{t_0+T} w_f w_\theta^T dt \right] \phi_{av} \quad (3.29)$$

$$= -\frac{\alpha}{k^*} R_{w_f w_\theta}(0) \phi_{av}. \quad (3.30)$$

The averaging theorem in [Sastry and Bodson, 1989] proved that assuming the cross correlation matrix  $R_{w_f w_\theta}(0)$  exists and  $\alpha$  is sufficiently small the original system is exponentially stable if the averaged system is exponentially stable. Therefore, in order to derive a stability condition for the original system, we need only to derive a stability condition of the averaged system given in eq.(3.30).

First, let us express the cross correlation matrix  $R_{w_f w_\theta}(0)$  in terms of the frequency content of the reference input. Defining  $S_r(d\omega)$  be the spectral measure of the reference input, we can express  $R_{w_f w_\theta}(0)$  as

$$R_{w_f w_\theta}(0) = \frac{1}{2\pi k^*} \int \left| \frac{\Phi_m(j\omega)}{\Phi_\theta(j\omega)} \right|^2 |W_m(j\omega)|^2 \frac{\Phi_m(j\omega)}{\Phi_\theta(j\omega)} [I + G_\theta] H_{w_m r}^*(j\omega) H_{w_m r}^T(j\omega) [I + G_\theta^T] S_r(d\omega) \quad (3.31)$$

where

$$G_\theta = \begin{bmatrix} 0 & \bar{\phi}^T/k^* \\ 0 & \phi_k/k^* I \end{bmatrix} \in \mathbb{R}^{2n \times 2n} \quad (3.32)$$

and

$$\bar{\phi} = \begin{bmatrix} \theta_1 - \theta_1^* \\ \theta_0 - \theta_0^* \\ \theta_2 - \theta_2^* \end{bmatrix} \quad (3.33)$$

$$\phi_k = k - k^*. \quad (3.34)$$

See Appendix A for the derivation of the above expression.

For simplicity of expression, we assume the following concerning the reference input:

(A7) The reference input  $r$  is a summation of sinusoidal signals with  $N$  distinct frequencies such as

$$r = \sum_{i=1}^N R_i \sin(\omega_i t), \quad R_i > 0 \text{ for all } i. \quad (3.35)$$

$R_{w_f w_\theta}(0)$  then can be expressed as

$$R_{w_f w_\theta}(0) = \frac{1}{k^*} \sum_{i=1}^N R_i^2 \left| \frac{\Phi_m(j\omega_i)}{\Phi_\theta(j\omega_i)} \right|^2 |W_m(j\omega_i)|^2 \frac{\Phi_m(j\omega_i)}{\Phi_\theta(j\omega_i)} [I + G_\theta] H_{w_m r}^*(j\omega_i) H_{w_m r}^T(j\omega_i) [I + G_\theta^T]. \quad (3.36)$$

The following lemma can be easily proved.

**Lemma 3.1** *The averaged system in eq.(3.30) is exponentially stable if the real parts of all the eigenvalues of  $R_{w_f w_\theta}(0)$  are positive.*

Proof A natural Lyapunov function is given as

$$V(\phi_{av}) = |\phi_{av}|^2 \quad (3.37)$$

and, from eq.(3.30),

$$\dot{V}(\phi_{av}) = -\alpha \phi_{av}^T [R_{w_f w_\theta}(0) + R_{w_f w_\theta}(0)^T] \phi_{av}. \quad (3.38)$$

If the real parts of all the eigenvalues of  $R_{w_f w_\theta}(0)$  are positive, the matrix in the parentheses above is symmetric positive definite. Therefore, letting  $\lambda_{\min}$  be the smallest eigenvalues of  $R_{w_f w_\theta}(0)$  within the stability range of  $\phi_{av}$ , we get

$$-\dot{V}(\phi_{av}) \geq \alpha \lambda_{\min} V(\phi_{av}). \quad (3.39)$$

Namely, the parameter error converges exponentially to zero with the rate of  $\alpha \lambda_{\min}$ .

□

Since  $\phi^T G_\theta = 0$  for all  $\phi$ , the derivative of the Lyapunov function given in eq.(3.38) can be rewritten as

$$\dot{V}(\phi_{av}) = -\alpha \phi_{av}^T [R_\theta + R_\theta^T] \phi_{av} \quad (3.40)$$

where

$$R_\theta = \frac{1}{k^*} \sum_{i=1}^N R_i^2 \left| \frac{\Phi_m(j\omega_i)}{\Phi_\theta(j\omega_i)} \right|^2 |W_m(j\omega_i)|^2 \frac{\Phi_m(j\omega_i)}{\Phi_\theta(j\omega_i)} H_{w_{mr}}^*(j\omega_i) H_{w_{mr}}^T(j\omega_i). \quad (3.41)$$

Assume that the matrices  $H_{w_{mr}}^*(j\omega_i) H_{w_{mr}}^T(j\omega_i)$  are linearly independent for given  $\omega_i$ ,  $i = 1, 2, \dots, N$ . Since  $R_i^2 \left| \frac{\Phi_m(j\omega_i)}{\Phi_\theta(j\omega_i)} \right|^2 |W_m(j\omega_i)|^2$  is strictly positive given any  $\phi_{av}$  bounded, the above lemma and the averaging theorem automatically prove the following stability condition:

**Theorem 3.1 (Stability Conditions)** *Suppose all the assumptions from (A1) to (A8) are satisfied. Then, the original system given in eq.(3.26) is exponentially stable, if*

$$\operatorname{Re} \left\{ \frac{\Phi_m(j\omega_i)}{\Phi_\theta(j\omega_i)} \right\} > 0 \quad \text{for all } \omega_i, \quad (3.42)$$

or

$$|\arg\{\Phi_m(j\omega)\} - \arg\{\Phi_\theta(j\omega)\}| < \frac{\pi}{2} \quad \text{for all } \omega_i, \quad (3.43)$$

Note that this condition is only a sufficient condition and is very conservative. However, since our objective in this chapter is to prove that there exist a sequence of reference inputs that guarantee the stable convergence of the control parameters, we consider that the derivation of the necessary condition is not needed.

### 3.4 Input Design in Frequency Domain

In the previous section, we proved that an adaptive system is stable if the system is excited within a particular frequency range that satisfies the stability condition given in Theorem 3.1. Since the model characteristic function  $\Phi_m(s)$  is given, the frequency range for the conservative stability depends only on the control parameter vector. In other words, the frequency stability range becomes wider as the control parameter vector  $\theta$  approaches the optimal parameter vector  $\theta^*$ , and once the parameter vector reaches the neighbor of the optimal one the stability range covers the whole frequency range. This argument agrees with our intuitive argument of progressive learning. Namely, learning is difficult and may cause instability in the beginning, but if learning starts at a low task complexity level (low frequency excitation in this case) and the complexity level is gradually increased according to the progress of learning, the learner can find the optimal solution progressively. The objective of this section is to formulate and prove the above argument based on the stability condition derived in the previous section. A numerical example to verify the analysis follows at the end.

#### 3.4.1 Frequency Range for Stability

We first define the frequency range for stability based on Theorem 3.1 as follows. For a given parameter vector  $\theta$ , let  $\Theta_\theta$  be the frequency range for stability such that

$$|\arg\{\Phi_m(j\omega)\} - \arg\{\Phi_\theta(j\omega)\}| < \frac{\pi}{2} \quad \text{for all } \omega \in \Theta_\theta \quad (3.44)$$

With this definition, we can state that the overall stability is guaranteed if we design the reference input as follows:

$$r = \sum_{i=1}^N R_i \sin(\omega_i t), \quad \omega_i \in \Theta_\theta, \quad i = 1, \dots, N \quad (3.45)$$

Let us assume

(B1) the initial parameter set  $\theta^{(0)}$  is always given so that the phase angle of  $\Phi_\theta(s)$  at low frequencies are almost zero as shown in Figure 3.2.

Since a reference model is always stable and it becomes a DC at low frequencies, the phase angle difference between the reference model and the initial closed-loop system is small at the low frequencies. Namely, there exists a frequency  $\omega^{(0)}$  such that

$$\Omega_{\theta^{(0)}} \stackrel{\text{def}}{=} \{\omega | 0 \leq \omega \leq \omega^{(0)}\} \in \Theta_{\theta^{(0)}} \quad (3.46)$$

where  $\Omega_{\theta^{(0)}}$  is the largest continuous subset of  $\Theta_{\theta^{(0)}}$  that includes zero.

If the plant  $W_p = Z_p/R_p$  does not have an integrator, the zero initialization ( $\theta^{(0)} = 0$ , no feedback loop) gives

$$\Phi_{\theta^0}(s) = \lambda(s)R_p(s) \quad (3.47)$$

and, therefore, suffices to hold the assumption. Even if the plant has an integrator, it is easy to show that small values for  $\theta$  and the resultant feedback loop satisfies the above assumption. Note that the stability condition given in eq.(3.43) is also satisfied at high frequencies. For example, if both the characteristic functions are Hurwitz, the phase angles of the both functions reach  $90 \times (2n - 1)$  degrees with a high frequency. However, those frequencies are beyond the bandwidth of the reference model and there are virtually no information acquired by exciting the system at these frequencies. Therefore, we do not include the high frequency range in the definition of the frequency range for stability.

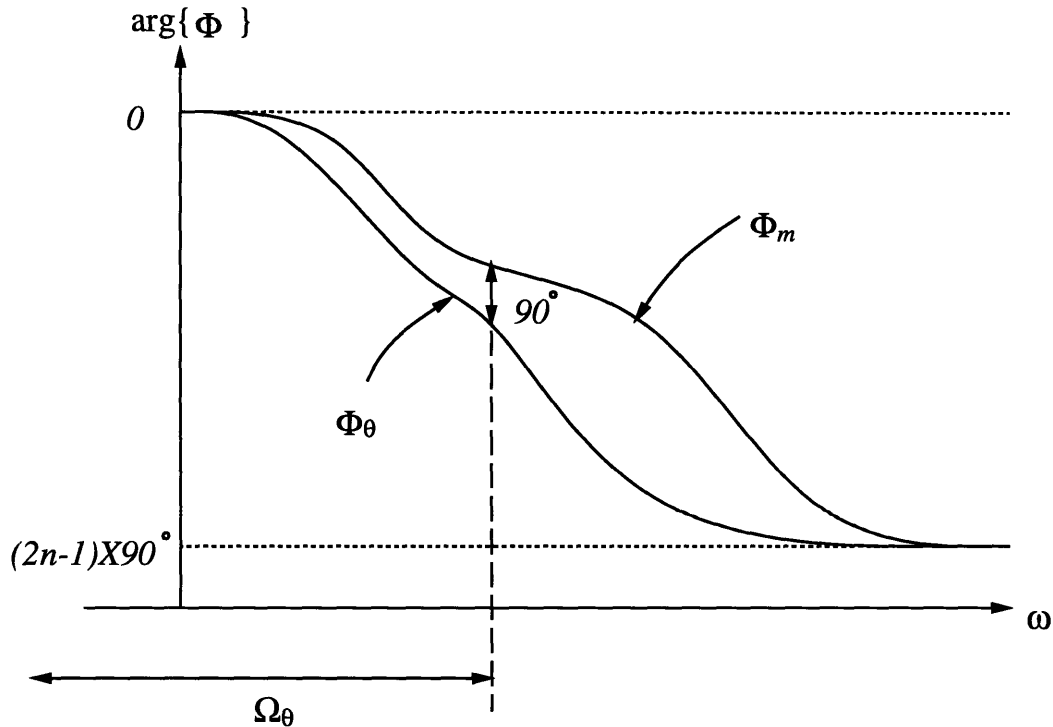


Figure 3.2: Frequency range of stability for a given  $\theta$

### 3.4.2 Transfer Function Matching in Frequency Domain

In this section, an important concept of “transfer function matching in frequency domain” is introduced to explain the behavior of an adaptive control system. Suppose that the reference input consists of summation of sinusoidal signals with  $N$  distinct frequencies such as following:

$$r = \sum_{i=1}^N R_i \sin(\omega_i t) \quad (3.48)$$

Assuming that the output error tends to zero with a stability-proven adaptive law, [Boyd and Sastry, 1986] proved that the control parameters  $\theta$  converge to a point where

$$\begin{aligned} W_\theta(j\omega_i) &= W_m(j\omega_i) \\ W_\theta(-j\omega_i) &= W_m(-j\omega_i) \end{aligned} \quad i = 1, 2, \dots, N. \quad (3.49)$$

Or, since

$$W_\theta(s) = \frac{\Phi_m(s)}{\Phi_\theta(s)} W_m(s), \quad (3.50)$$

the above equations can be rewritten using the characteristic equations as:

$$\begin{aligned} \Phi_\theta(j\omega_i) &= \Phi_m(j\omega_i) \\ \Phi_\theta(-j\omega_i) &= \Phi_m(-j\omega_i) \end{aligned} \quad i = 1, 2, \dots, N. \quad (3.51)$$

In other words, the control parameters converge to a solution of the above simultaneous equations in terms of  $\theta$ . Important to note is that the solution is not necessarily the same as the optimal control parameter vector  $\theta^*$ . For example, if the number of the distinct frequencies of the input signals,  $N$ , is not larger than  $(2n-1)/2$ , where  $2n-1$  is the number of the control parameters, then the above simultaneous equations become underdetermined and the control parameters converge to one of the infinite number of solutions. The persistent excitation condition is thus translated to the number of the distinct frequencies in the reference input.

And more importantly, even if the number of distinct frequencies is sufficiently large, the control parameters  $\theta$  may fail to converge to the desired values  $\theta^*$  if the frequency range of the reference input does not cover the whole frequency range (bandwidth) of the reference model. Namely, if the reference input excites only the slow, quasi-static modes of the reference model, the closed-loop characteristic function  $\Phi_\theta(j\omega)$  converges to match the model characteristic function only in the slow modes, and the lack in matching occurs over the frequency range of the faster modes as shown in Figure 3.3.

### 3.4.3 Progressive Excitation

The idea of progressive learning is that the system is excited in low frequencies in the beginning to avoid the instability and the stable excitation level is increased



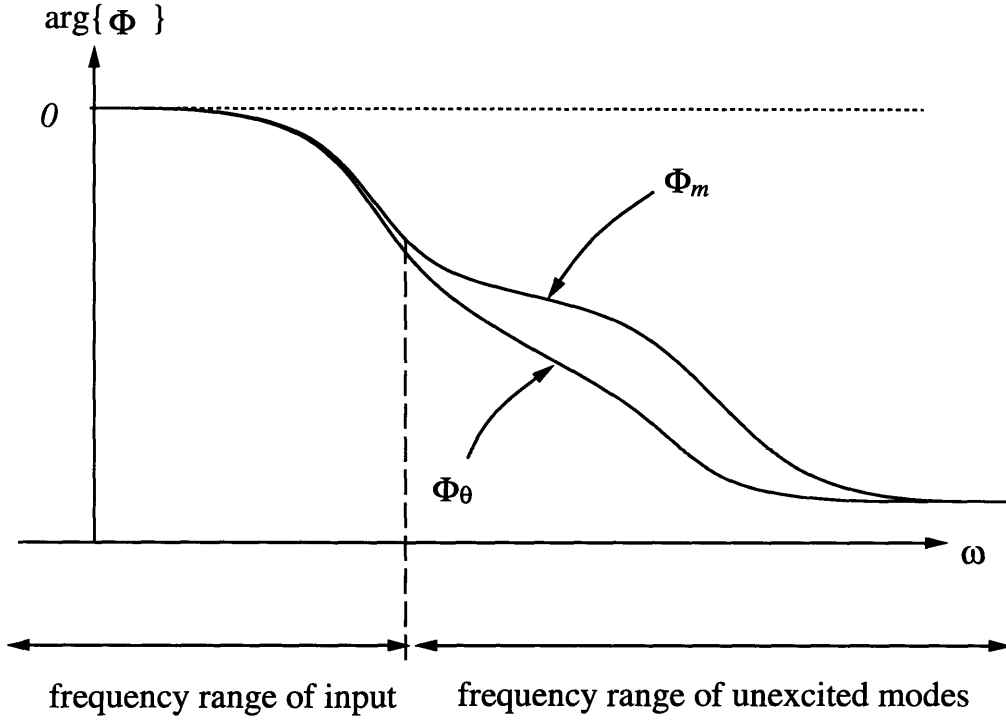


Figure 3.3: Frequency domain matching with partially exciting reference input

gradually according to the progress of learning. In this section, we formulate and prove the following statement:

*For given plant, reference model, adaptation rule, and initial parameter values, there always exist a sequence of excitation frequencies such that the system is maintained stable and the control parameters converge to the optimal ones and the output error converges to zero.*

The following proposition is important to prove the above statement.

**Proposition 3.1** *Suppose  $\theta$  is given such that*

$$\Phi_{\theta}(s) = s^h \prod_{i=1}^m (s^2 + 2\zeta_i \omega_i s + \omega_i^2) \prod_{j=1}^k (s + \omega'_j) \quad (3.52)$$

and

$$0 < \zeta_i < 1 \quad \text{for } i = 1, 2, \dots, m, \quad (3.53)$$

$$\omega_i \neq 0 \quad \text{for } i = 1, 2, \dots, m, \quad (3.54)$$

$$\omega'_j \neq 0 \quad \text{for } j = 1, 2, \dots, k. \quad (3.55)$$

Then, there exists  $M > 0$  such that

$$\left| \frac{d}{d\omega} \arg\{\Phi_\theta(j\omega)\} \right| < M \quad \text{all } \omega > 0 \quad (3.56)$$

Proof From eq.(3.52), the phase angle of  $\Phi_\theta(j\omega)$  for  $\omega > 0$  is expressed as

$$\arg \Phi_\theta(j\omega) = \frac{h\pi}{2} + \sum_{i=1}^m \tan^{-1}\left(\frac{2\zeta_i\omega_i\omega}{\omega_i^2 - \omega^2}\right) + \sum_{j=1}^k \tan^{-1}\left(\frac{\omega}{\omega'_j}\right) \quad (3.57)$$

and its derivative in terms of  $\omega$  for  $\omega > 0$  is expressed as

$$\frac{d}{d\omega} \arg \Phi_\theta(j\omega) = \sum_{i=1}^m \frac{2\zeta_i\omega_i(\omega_i^2 + \omega^2)}{(\omega_i^2 - \omega^2)^2 + (2\zeta_i\omega_i\omega)^2} + \sum_{j=1}^k \frac{\omega'_j}{\omega_j'^2 + \omega^2}. \quad (3.58)$$

For  $i = 1, \dots, m$ , let us define  $f_i(j\omega)$  as

$$f_i(j\omega) = \frac{2\zeta_i\omega_i(\omega_i^2 + \omega^2)}{(\omega_i^2 - \omega^2)^2 + (2\zeta_i\omega_i\omega)^2} \quad (3.59)$$

It is easy to derive that  $|f_i(j\omega)|$  becomes the largest at  $\omega = |\omega_i|\sqrt{-1 + 2\sqrt{1 - \zeta_i^2}}$  if  $\zeta_i < \sqrt{3}/2$ , or at  $\omega \rightarrow 0$  otherwise. Namely,

$$\sup_{\omega} |f_i(j\omega)| = \begin{cases} \frac{\zeta_i}{2|\omega_i|\sqrt{1-\zeta_i}(1-\sqrt{1-\zeta_i})} & \text{if } 0 < \zeta_i < \frac{\sqrt{3}}{2} \\ 2\zeta_i/\omega_i & \text{otherwise} \end{cases} \quad (3.60)$$

Since the first supremum in the above equation becomes much larger than the second one as  $\zeta_i$  goes to small, we assume that all  $\zeta_i$ s are smaller than  $\sqrt{3}/2$  for the calculation of the upperbound of  $|f_i(j\omega)|$  without losing generality.

Let  $\omega_{\min}$  be the minimum value of  $|\omega_i|$  and  $\zeta_{\min}$  be the minimum value of  $\zeta_i$  for  $i = 1, \dots, m$ . Then, the upperbound of the first summation of eq.(3.58) is given by

$$\left| \sum_{i=1}^m \frac{2\zeta_i \omega_i (\omega_i^2 + \omega^2)}{(\omega_i^2 - \omega^2)^2 + (2\zeta_i \omega_i \omega)^2} \right| \leq \frac{m\zeta_{\min}}{2\omega_{\min} \sqrt{1 - \zeta_{\min}} (1 - \sqrt{1 - \zeta_{\min}})} \quad (3.61)$$

The upperbound of the second summation of eq.(3.58) is easily derived as:

$$\left| \sum_{j=1}^k \frac{\omega'_j}{\omega_j'^2 + \omega^2} \right| < \frac{k}{\omega'_{\min}} \quad (3.62)$$

where  $\omega'_{\min}$  is the smallest value of  $\omega'_j$ . Letting  $M$  be

$$M = \frac{m\zeta_{\min}}{2\omega_{\min} \sqrt{1 - \zeta_{\min}} (1 - \sqrt{1 - \zeta_{\min}})} + \frac{k}{\omega'_{\min}}, \quad (3.63)$$

we can obtain

$$\left| \frac{d}{d\omega} \arg \Phi_{\theta}(j\omega) \right| < M \quad \text{for all } \omega > 0 \quad (3.64)$$

□

With the above proposition, we next prove the following lemma.

**Lemma 3.2** *Suppose that assumptions (A1) to (A5) are satisfied. For a given  $\omega_l > 0$  and  $\theta$ , assume that*

$$\Phi_{\theta}(j\omega_l) = \Phi_m(j\omega_l). \quad (3.65)$$

*Then, there exists  $\varepsilon > 0$  such that*

$$|\arg\{\Phi_m(j\omega)\} - \arg\{\Phi_{\theta}(j\omega)\}| < \frac{\pi}{2} \quad \text{for all } \omega \in \{\omega \mid |\omega - \omega_l| \leq \varepsilon\} \quad (3.66)$$

*Proof* Similarly to the proof of Proposition 3.1, we first express  $\Phi_{\theta}(s)$  as

$$\Phi_{\theta}(s) = s^h \prod_{i=1}^m (s^2 + 2\zeta_i \omega_i s + \omega_i^2) \prod_{j=1}^k (s + \omega'_j) \quad (3.67)$$

where

$$0 \leq \zeta_i < 1 \quad \text{for } i = 1, 2, \dots, m, \quad (3.68)$$

$$\omega_i \neq 0 \quad \text{for } i = 1, 2, \dots, m, \quad (3.69)$$

$$\omega'_j \neq 0 \quad \text{for } j = 1, 2, \dots, k \quad (3.70)$$

We also define

$$\Gamma_\theta(j\omega) \stackrel{\text{def}}{=} \arg\{\Phi_m(j\omega)\} - \arg\{\Phi_\theta(j\omega)\}. \quad (3.71)$$

We consider the following two cases:

*Case 1.*  $\zeta_i > 0$  for all  $i = 1, 2, \dots, m$ . From the above proposition, there exists  $M > 0$  such that

$$\left| \frac{d}{d\omega} \arg \Phi_\theta(j\omega) \right| < M \quad \text{for all } \omega > 0 \quad (3.72)$$

From the assumption (A4),  $\Phi_m(s)$  is strictly Hurwitz and, therefore, its derivative is also bounded by a finite positive value  $M_m$  as

$$\left| \frac{d}{d\omega} \arg \Phi_m(j\omega) \right| < M_m \quad \text{for all } \omega > 0 \quad (3.73)$$

From the above inequalities and eq.(3.71), we obtain

$$\left| \frac{d}{d\omega} \Gamma_\theta(j\omega) \right| < M + M_m \quad \text{for all } \omega > 0 \quad (3.74)$$

Since  $\Gamma_\theta(j\omega_l) = 0$  from eq.(3.65), we get

$$|\Gamma_\theta(j\omega)| < (M + M_m)|\omega - \omega_l| \quad \text{for all } \omega > 0 \quad (3.75)$$

Therefore, by choosing

$$\varepsilon = \frac{\pi/2}{M + M_m}, \quad (3.76)$$

we proved that

$$|\Gamma_\theta(j\omega)| < \frac{\pi}{2} \quad \text{for all } \omega \in \{\omega \mid |\omega - \omega_l| \leq \varepsilon\} \quad (3.77)$$

*Case 2.*  $\zeta_i = 0$  for some  $is$ . We first express  $\Phi_\theta(s)$  as

$$\Phi_\theta(s) = s^h \prod_{t=1}^p (s^2 + \omega_t''^2) \prod_{i=1}^q (s^2 + 2\zeta_i \omega_i s + \omega_i^2) \prod_{j=1}^k (s + \omega_j') \quad (3.78)$$

where

$$0 \leq \zeta_i < 1 \quad \text{for } i = 1, 2, \dots, m, \quad (3.79)$$

$$\omega_i \neq 0 \quad \text{for } i = 1, 2, \dots, q, \quad (3.80)$$

$$\omega'_j \neq 0 \quad \text{for } j = 1, 2, \dots, k \quad (3.81)$$

$$\omega''_t \neq 0 \quad \text{for } t = 1, 2, \dots, p \quad (3.82)$$

Since  $\Phi_\theta(s)$  has imaginary roots, its derivative in terms of  $\omega$  cannot be bounded any more. In what follows, we consider only in the neighbor of  $\omega_l$  and prove that the derivative can be bounded in the small range.

From eq.(3.78), we get

$$\arg \Phi_\theta(j\omega) = \frac{h\pi}{2} + \sum_{i=1}^q \tan^{-1}\left(\frac{2\zeta_i\omega_i\omega}{\omega_i^2 - \omega^2}\right) + \sum_{j=1}^k \tan^{-1}\left(\frac{\omega}{\omega'_j}\right) + \sum_{t=1}^p \arg\{\omega_t''^2 - \omega^2\} \quad \text{for } \omega > 0 \quad (3.83)$$

The derivative of all the terms in the above equation are bounded except the last terms  $\arg\{\omega_t''^2 - \omega^2\}$ . And also, we get

$$\frac{d}{d\omega} \arg\{\omega_t''^2 - \omega^2\} = \begin{cases} 0 & \text{if } 0 < \omega \neq \omega_t'' \\ \infty & \text{if } \omega \rightarrow \omega_t'' \end{cases} . \quad (3.84)$$

Therefore, to prove the above lemma, it is sufficient to consider a case in which  $\omega_l$  coincides with  $\omega_t''$  or  $\omega_l$  is located in the neighbor of  $\omega_t''$ .

Without losing generality, we assume that  $\Phi_\theta(s)$  has a repeated imaginary roots at  $\omega_l + \varepsilon_0$  and  $\Phi_\theta(s)$  in eq.(3.78) can be rewritten as

$$\Phi_\theta(s) = s^h (s^2 + (\omega_l + \varepsilon_0)^2)^d \prod_{t=1}^{p-d} (s^2 + \omega_t''^2) \prod_{i=1}^q (s^2 + 2\zeta_i\omega_i s + \omega_i^2) \prod_{j=1}^k (s + \omega'_j) \quad (3.85)$$

where

$$0 \leq |\varepsilon_0| < |\omega_t'' - \omega_l| \quad \text{for } t = 1, 2, \dots, p - d. \quad (3.86)$$

From the above equation, we get

$$|\Phi_\theta(j\omega_l)| = |\varepsilon_0^2 + 2\varepsilon_0\omega_l|^d \sigma_p \quad (3.87)$$

where

$$\sigma_p \stackrel{\text{def}}{=} \omega_l^h \prod_{t=1}^{p-d} |-\omega_l^2 + \omega_t''^2| \prod_{i=1}^q \sqrt{(\omega_i^2 - \omega_l^2)^2 + (2\zeta_i \omega_i \omega_l)^2} \prod_{j=1}^k \sqrt{\omega_l^2 + \omega_j'^2} \quad (3.88)$$

From eq.(3.86), we get

$$\sigma_p > 0 \quad (3.89)$$

From eq.(3.65), we also get

$$|\Phi_\theta(j\omega_l)| = |\Phi_m(j\omega_l)|. \quad (3.90)$$

Since the reference model is strictly stable,  $\Phi_m(s)$  is strictly Hurwitz and there exists  $\sigma_m > 0$  such that

$$|\Phi_m(j\omega_l)| > \sigma_m \quad \text{for all } \omega. \quad (3.91)$$

From eqs.(3.87) and (3.91), we get

$$|\varepsilon_0^2 + 2\varepsilon_0\omega_l|^d > \frac{\sigma_m}{\sigma_p} \quad (3.92)$$

Let us Define  $\varepsilon_1$  and  $\varepsilon_2$  as

$$\varepsilon_1 = \omega_l - \sqrt{\omega_l^2 - \left(\frac{\sigma_m}{\sigma_p}\right)^{\frac{1}{d}}} \quad (3.93)$$

$$\varepsilon_2 = -\omega_l + \sqrt{\omega_l^2 + \left(\frac{\sigma_m}{\sigma_p}\right)^{\frac{1}{d}}} \quad (3.94)$$

where  $\varepsilon_1 > 0$  and  $\varepsilon_2 > 0$  since  $d$  is bounded such as  $0 \leq 2d < 2n - 1$ . Then, the above inequality in eq.(3.92) implies

$$\varepsilon_0 < -\varepsilon_1 \quad \text{or} \quad \varepsilon_2 < -\varepsilon_0 \quad (3.95)$$

Namely,  $\Phi_\theta(s)$  cannot have imaginary roots in  $T_l \stackrel{\text{def}}{=} \{\omega | \omega_l - \varepsilon_1 \leq \omega \leq \omega_l + \varepsilon_2\}$ . In other words, according to the proposition, the derivative of  $\Phi_\theta(j\omega)$  is bounded in the

neighbor of  $\omega_l$ . Let  $M'$  be the supremum of the derivative of  $\Phi_\theta(j\omega)$  in  $T_l$  and  $M_m$  be the supremum of the derivative of  $\Phi_m(j\omega)$ . Then,

$$\left| \frac{d}{d\omega} \Gamma_\theta(j\omega) \right| < M' + M_m \quad \text{for all } \omega \in T_l \quad (3.96)$$

Since  $\Gamma_\theta(j\omega_l) = 0$  from the assumption, we get

$$|\Gamma_\theta(j\omega)| < (M' + M_m)|\omega - \omega_l| \quad \text{for all } \omega \in T_l \quad (3.97)$$

Let  $\varepsilon_3$  be

$$\varepsilon_3 = \frac{\pi/2}{M' + M_m}. \quad (3.98)$$

Then, by choosing  $\varepsilon = \min\{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$ , we guarantee

$$|\Gamma_\theta(j\omega)| < \frac{\pi}{2} \quad \text{for all } \omega \in \{\omega \mid |\omega - \omega_l| \leq \varepsilon\} \quad (3.99)$$

Therefore, the lemma was proved.  $\square$

With the above lemma, we can prove the existence of a sequence of excitations for progressive learning. Let us first introduce the following assumptions.

(B2) For a given  $\theta^{(h-1)}$ , there exists  $\omega^{(h)}$  such that

$$\Omega_{\theta^{(h-1)}} = \{\omega \mid 0 \leq \omega \leq \omega^{(h)}\} \quad (3.100)$$

(B3) The reference signal  $r$  is designed as

$$r = \sum_{i=1}^N R_i \sin(\omega_i t) \quad (3.101)$$

where all  $R_i > 0$ ,  $N$  is sufficiently large, and  $\omega_i$ s include  $\omega^{(h)}$  and are uniformly distributed in  $\Omega_{\theta^{(h-1)}}$ .

Following is the main theoretical result that supports the above argument illustrated by Figure 3.4.

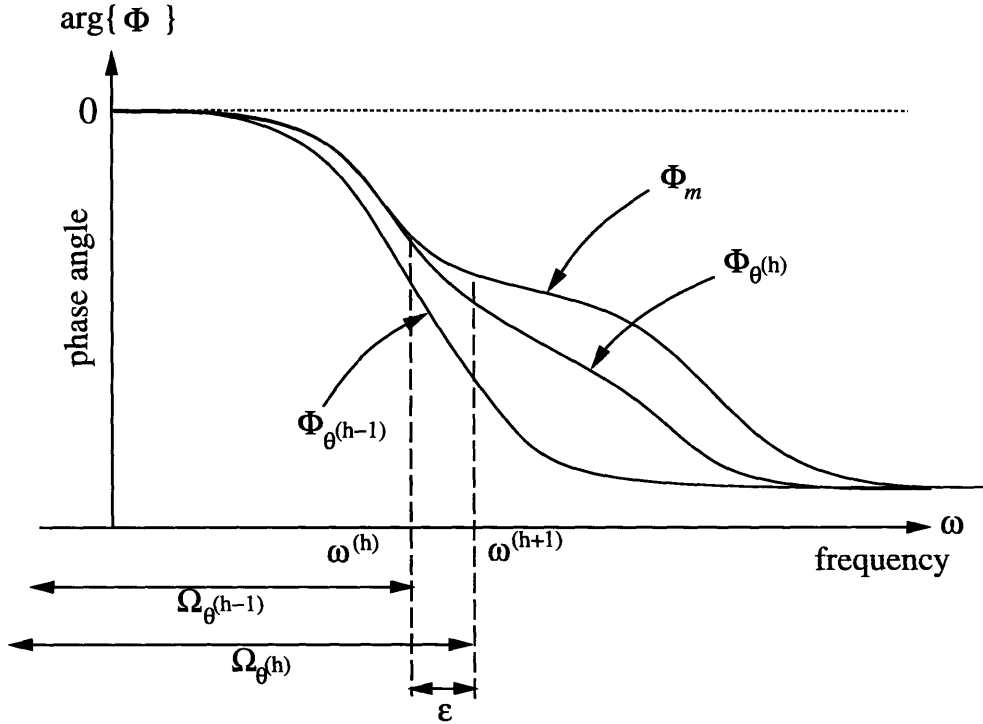


Figure 3.4: Progressive expansion of the stability range

**Theorem 3.2 (Progressive Excitation Theorem)** *Assume all the assumptions from (A1) to (A8) and from (B2) to (B3) are satisfied. Also suppose that the parameter vector converges from  $\theta^{(h-1)}$  to  $\theta^{(h)}$  by a stable adaptation law with the above reference input. Then, there always exists  $\varepsilon > 0$  such that*

$$\omega^{(h+1)} = \omega^{(h)} + \varepsilon, \quad \text{and} \quad (3.102)$$

$$|\arg\{\Phi_m(j\omega)\} - \arg\{\Phi_{\theta^{(h)}}(j\omega)\}| < \frac{\pi}{2} \quad \text{for all } \omega \in \{\omega \mid 0 \leq \omega \leq \omega_{\theta^{(h)}}\} \quad (3.103)$$

*Proof* First, from the frequency domain matching theorem by [Boyd and Sastry, 1986] and the condition (B3), it is obvious that

$$|\arg\{\Phi_m(j\omega)\} - \arg\{\Phi_{\theta^{(h)}}(j\omega)\}| < \frac{\pi}{2} \quad \text{for all } \omega \in \{\omega \mid 0 \leq \omega \leq \omega_{\theta^{(h)}}\} \quad (3.104)$$



In other words, by exciting the system with a large number of frequencies within the previous frequency range of stability  $\Omega_{\theta^{(h-1)}}$ , the stability range is at least maintained even after the change of the control parameters due to adaptation.

Next, from (B3), the system is excited at  $\omega^{(h)}$  and the frequency domain matching occurs at the frequency as

$$\Phi_{\theta^{(h-1)}}(j\omega^{(h)}) = \Phi_m(j\omega^{(h)}). \quad (3.105)$$

Then, from Lemma 4.1, there exists  $\varepsilon_0 > 0$  such that

$$|\arg\{\Phi_m(j\omega)\} - \arg\{\Phi_{\theta^{(h-1)}}(j\omega)\}| < \frac{\pi}{2} \quad \text{for all } \omega \in \{\omega \mid |\omega - \omega^{(h)}| \leq \varepsilon_0\} \quad (3.106)$$

From eqs.(3.104) and (3.106), by choosing  $\varepsilon = \varepsilon_0$ , we proved the theorem.  $\square$

From Theorem 3.2 and the assumption (B1), we can automatically prove the existence of a sequence of excitation frequencies for stable convergence.

### 3.5 Simulation

In this section, the arguments about stability presented in the previous sections is shown to be valid by simulation. A plant with relative degree 3 is used as an example and it is demonstrated that the stable parameter convergence can be achieved by a sequence of progressively exciting reference signals even with a gradient descent adaptation.

#### A. Plant and Controller

The transfer function of the plant and the reference model are chosen to be

$$W_p(s) = \frac{Z_p(s)}{R_p(s)} = \frac{1}{(s+1)(s^2+0.4s+1.04)} \quad (3.107)$$

$$W_m(s) = \frac{Z_m(s)}{R_m(s)} = \frac{1}{(s+2)(s^2+6s+45)} \quad (3.108)$$

respectively. The fixed control parameters are

$$\Lambda = \begin{bmatrix} 0 & 1 \\ -2 & -1.5 \end{bmatrix}, \quad l = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \text{or} \quad \lambda(s) = s^2 + 1.5s + 2. \quad (3.109)$$

The nominal characteristic polynomial is given by

$$\Phi_m(s) = R_m(s)\lambda(s) = s^5 + 9.5s^4 + 70s^3 + 183.5s^2 + 192s + 90 \quad (3.110)$$

Six control parameters  $k$ ,  $\theta_1 = [\theta_{11}, \theta_{12}]^T$ ,  $\theta_0$ , and  $\theta_2 = [\theta_{21}, \theta_{22}]^T$  are adjusted using the gradient descent method. In this simulation, we assume that the plant parameters are known for the calculation of the gradient. The true values of the control parameters are

$$\begin{aligned} k^* &= 1 \\ \theta_1^* &= [-56.22, -6.6]^T \\ \theta_0^* &= -90.688, \\ \theta_2^* &= [60.1968, 34.8528]^T. \end{aligned}$$

### *B. Initialization and Instability Mechanism*

The feedforward gain  $k$  was initialized by 0.001 and the other control parameters were initialized by zeros. As a result, the initial closed-loop characteristic polynomial is

$$\Phi_{\theta(0)}(s) = R_p(s)\lambda(s) = s^5 + 2.9s^4 + 4.54s^3 + 4.6s^2 + 3s + 1.04 \quad (3.111)$$

Figure 3.5 shows the phase angle curves of both nominal and initial characteristic polynomials. As seen in the figure, the difference of phase angles between the two exceeds 90 degrees with frequencies higher than 1 rad/sec.

According to the stability analysis presented in Section 3, if the closed-loop system is excited at those high frequencies, it may cause instability. To verify the argument,

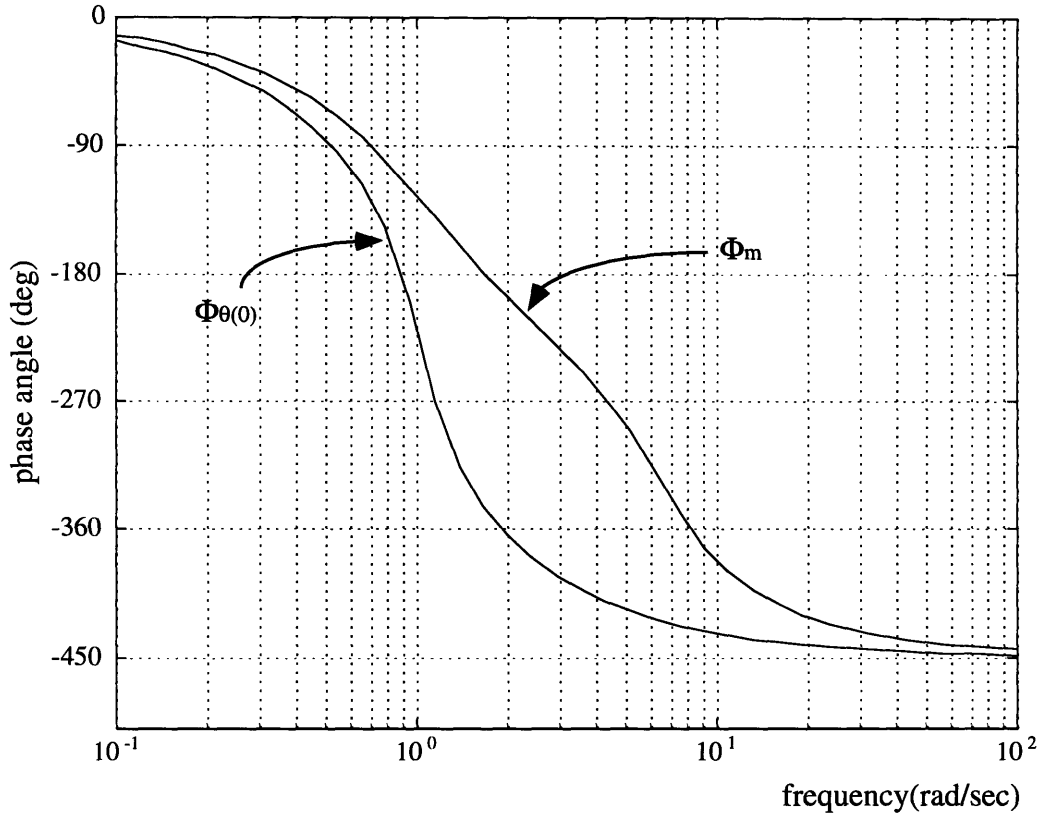


Figure 3.5: Initial phase shifts of  $\Phi_{\theta}(s)$  and  $\Phi_m(s)$

we first designed a reference signal as

$$r(t) = 200 \sin(0.2t) + 400 \sin(1.7t) + 500 \sin(2.5t) + 1050 \sin(3.2t), \quad (3.112)$$

and used the gradient descent rule given in Eq.(MITrule) to adjust the control parameters for 500 seconds. The results obtained are shown in Figures 3.6 (a)-(e), which show the response of the output error  $y_p(t) - y_m(t)$  as well as the adaptation curves of the control parameters. As shown in Figures 3.6-(b), (d) and (e), the control parameters  $k$ ,  $\theta_0$  and  $\theta_{21}$  moved in the direction which is opposite to the desired. As a result, the closed loop system became unstable around 400th second as shown in Figures 3.6-(a). This simulation results demonstrate that, if the frequency content of a reference input includes high frequencies for which a large difference in phase shift between the reference model and the closed-loop plant is generated, the adaptation

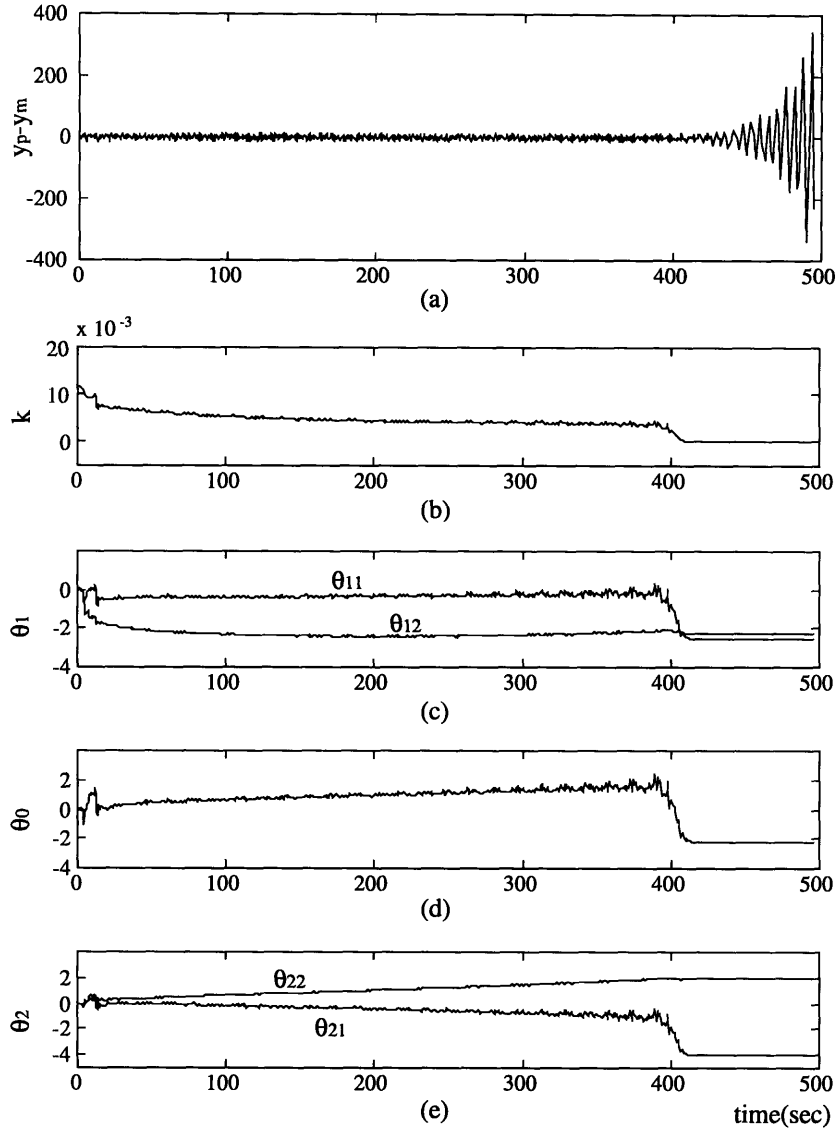


Figure 3.6: Results of the non-progressive excitation

causes instability.

### C. Progressive Excitations

The results of this subsection demonstrate that the instability can be avoided by progressively raising the frequencies of the sinusoidal reference input. In the light of Theorem 3.1, the stable parameter convergence is guaranteed if all the frequencies of

the reference input are chosen to be within the frequency range of stability, for which the phase angle differences between the reference model and the closed-loop plant are less than 90 degrees. In Figure 3.5, it is seen that the phase angle difference exceeds 90 degree at frequency around 0.9 rad/sec. Therefore we chose the sinusoidal reference input for the first excitation as

$$r(t) = 200 \sin(0.06t) + 400 \sin(0.15t) + 300 \sin(0.5t) + 250 \sin(0.8t), \quad (3.113)$$

and continued the adaptation for 300 seconds. Figure 3.7 shows the result of the first excitation. As seen in the figure, the control parameters changed slightly due to the adaptation. However, the output error decreased significantly as shown in Figure 3.7-(a). This result implies that the feedback controller with the very small gains suffices to track a trajectory with the low frequencies such as in eq.(3.113). Furthermore, the slight changes of the control parameters generated the significant shift of the phase angle curve of the closed-loop characteristic polynomial to  $\Phi_{\theta(1)}$  as shown in Figure 3.8, and as a result, the frequency range of stability was largely increased. This results complied with the analysis of progressive excitation given in the previous section.

Secondly, we raised the frequencies of the sinusoidal reference input as

$$r(t) = 200 \sin(0.06t) + 400 \sin(0.35t) + 300 \sin(0.6t) + 250 \sin(0.95t), \quad (3.114)$$

based on the new frequency range of stability obtained by the first excitation, and continued the adaptation for the next 300 seconds. The results of the second excitation are shown in Figures 3.7 and 3.8. Similarly to the first excitation, the second excitation did not generate significant changes in the control parameters and, therefore, the those values still remained very small. However, the output error decreased to a small value and the phase angle curve of the closed-loop polynomial shifted significantly to  $\Phi_{\theta(2)}$ . This simulation results show that only the low frequency modes of the system were

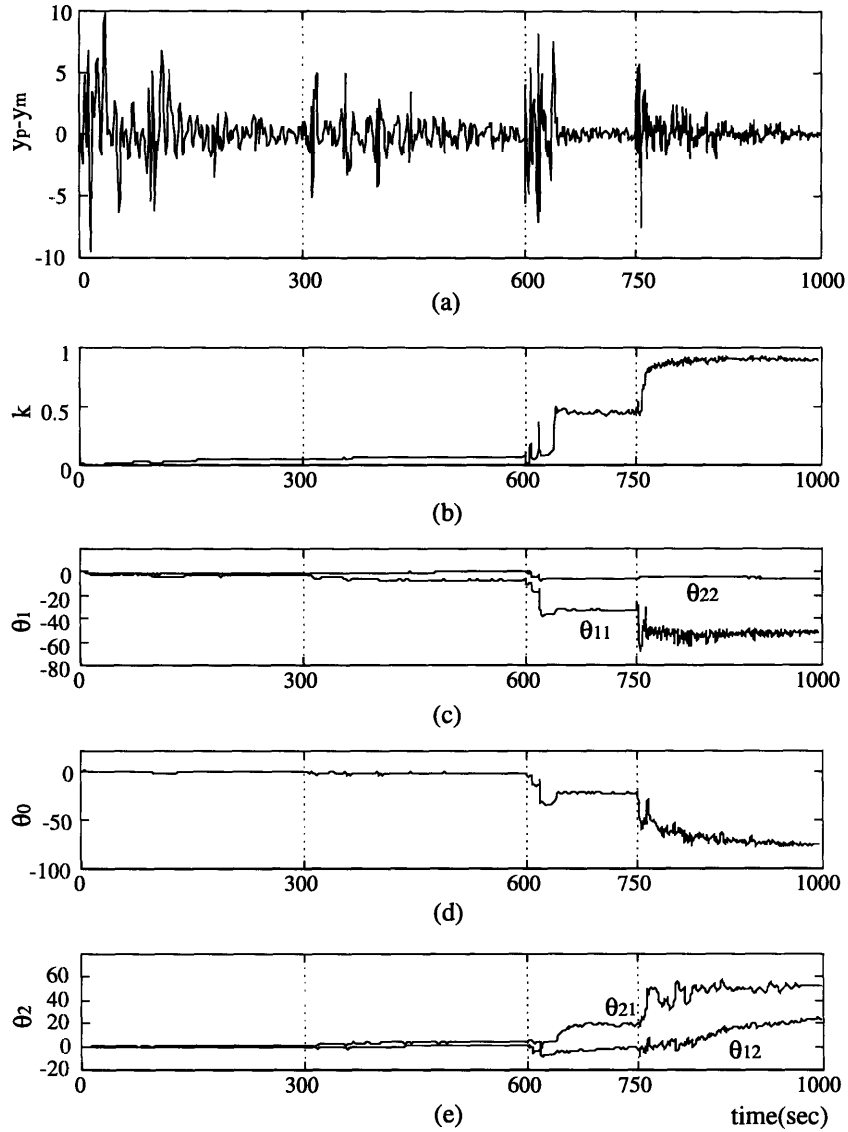


Figure 3.7: Output error and parameter curves with progressive excitation

excited by the sinusoidal reference input with the low frequencies given in eq.(3.114).

Based on the results given above, we increased the frequencies of the reference input again as

$$r(t) = 200 \sin(0.06t) + 400 \sin(0.45t) + 300 \sin(0.7t) + 250 \sin(1.1t), \quad (3.115)$$

and continued the adaptation for the next 150 seconds. Unlike the previous excitations,

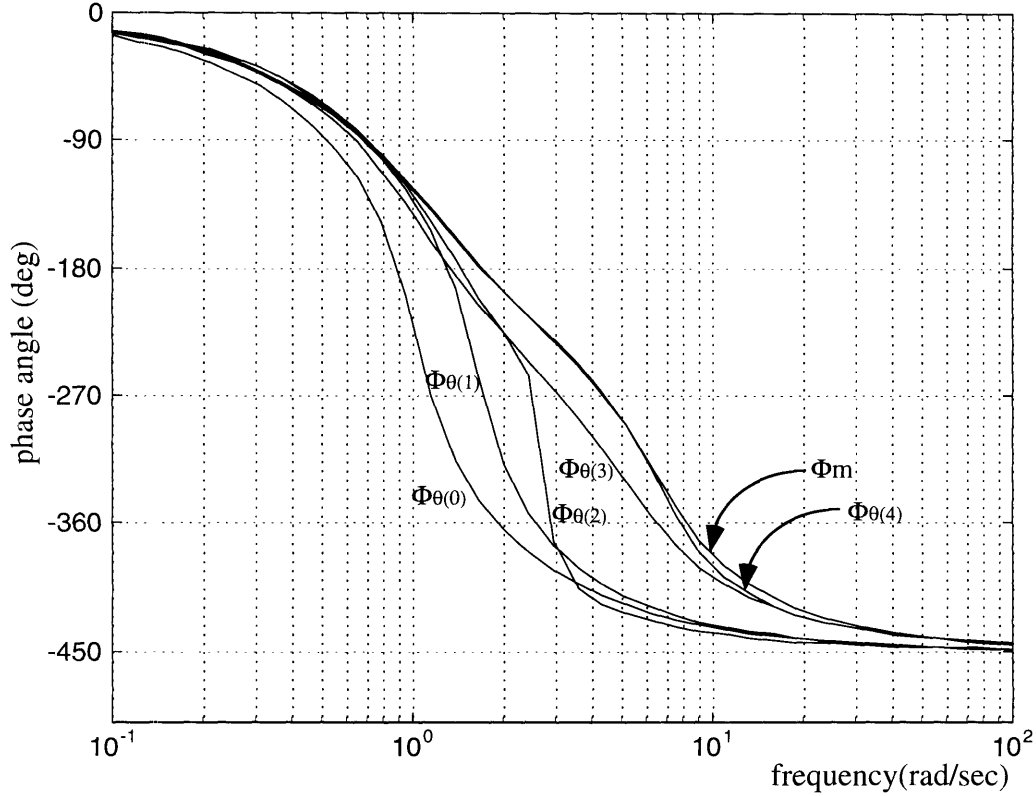


Figure 3.8: Progressive changes of phase angle curves

this reference input excited the more dynamic modes of the system and, as a result, abrupt changes in the control parameters occurred as shown in Figures 3.7 (b)-(e). Important to remark is that the excitation at the more dynamic modes drove the closed-loop system to match with the reference model in a wider frequency range and, as a result, the phase angle difference between  $\Phi_m$  and the resultant characteristic polynomial  $\Phi_{\theta(3)}$  became less than 90 degrees for all the frequency range as shown in Figure 3.8.

Finally, we excited the system in the full range by choosing the same reference input as given by eq.(3.112), that is,

$$r(t) = 200 \sin(0.2t) + 400 \sin(1.7t) + 500 \sin(2.5t) + 1050 \sin(3.2t). \quad (3.116)$$

As shown in Figures 3.7 and 3.8, the control parameters finally converged to the desired

values and the matching in the frequency domain occurred in all the frequency range.

### **3.6 Conclusions**

In this chapter, we developed a new input design method for stable adaptive systems based on the concept of progressive learning. The main idea of the new method is to excite the system gradually in accordance with the progress of the adaptation. By applying an averaging technique, we developed stability conditions of the progressive learning system in terms of the frequency contents of the reference input, and based on the analysis, we proved the existence of a sequence of reference inputs that achieve the progressive convergence of the control parameters for the adaptive control system. A system with relative degree of 3 was used as an example and all the arguments and analyses were verified through simulation.



## Progressive Learning for Robotic Assembly: Application to a Nonlinear Process

---

### 4.1 Introduction

In this chapter, the progressive learning method will be explored in the context of high-speed robotic assembly, and a specific learning algorithm will be developed for robot impedance control. High speed assembly is an important issue in industry as well as a challenging problem for academia. As motion speed increases, the assembly process becomes more dynamic and intricate. Unlike quasi-static assembly, which is performed using compliance or stiffness control, high-speed assembly needs dynamic control laws, e.g. full impedance control including damping and inertia terms. Such dynamic control laws contain a number of parameters to be tuned to a specific task process, which is often not exactly known. It is a difficult job to find the optimal values in a large parameter space where task process conditions are not exactly known. Learning control is a rational choice for coping with process uncertainties, but it must be feasible and effective for dealing with a large control parameter space. It should also be noticed that a failure in high speed assembly may incur serious damage to the robot and the environment. Even for the purpose of learning, fatal mistakes and failures must be avoided at all times.

Our solution to this problem is to use the progressive learning method. The idea is to increase motion speed gradually as learning proceeds. We start with a slow speed to learn a quasi-static compliance law, that is the stiffness term in the impedance control law. After the quasi-static law has been learned, motion speed is increased

slightly. Then the learning procedure is repeated for different motion speeds and the control law is relearned or re-tuned to the higher speed so that the robot can perform the task effectively at these higher speeds. At higher speeds, the inertial terms as well as some damping terms become more prominent than the stiffness terms. Therefore, these dynamic terms must be learned for higher speeds. Since we increase the speed gradually, the robot does not have to learn all the control parameters at the same time, but simply needs to refine the control law within a limited range of the large parameter space. Since the stiffness terms and some of the dynamic terms have been learned well for slower speeds, only the inertia and some damping terms need to be learned for the higher motion speed. Namely, the learning parameter space is “gradually excited” as the motion speed increases. This simplifies the learning problem significantly. Also important to note is that, since the robot has been trained successfully for a slower speed, the robot will not make fatal mistakes or fail totally, which could lead to serious damage. This makes the learning operations feasible.

In what follows, we will develop an effective progressive learning algorithm for a high-speed insertion task, and demonstrate the feasibility and effectiveness for the simple assembly problem. Particularly interesting is the problem of obtaining a schedule for varying motion speeds so that the learning of a dynamic control law can be performed quickly and safely.

## **4.2 High Speed Insertion Task**

### **4.2.1 Formulation of the Problem**

In this section, the concept of progressive learning is reduced to a concrete algorithm for high-speed robotic assembly. As shown in Figure 4.1-(a), the task is simply to insert a ball into a chamfered hole in an  $x - y$  plane. The controller is

given a nominal trajectory  $\mathbf{x}_d(t) = (x_d(t), y_d(t))^T$ . However, due to the uncertainty inherent in the assembly process, the hole is not precisely aligned with the trajectory and the ball often collides with a chamfer surface. Compliance control is necessary to cope with the geometric uncertainty of the assembly process, but is not sufficient for high speed insertion. For example, when the ball approaches a chamfer at high speed and collides with the surface, the quasi-static controller may not be able to prevent the ball from bouncing on the chamfer surface. When bouncing, the ball cannot be guided along the chamfer surface correctly and this insertion operation may result in failure, as addressed by [Asada and Kakumoto, 1990]. To avoid bouncing as well as to guide the ball correctly despite high speed, we need a dynamic control law such as full impedance control including damping and inertial terms along with the compliance or stiffness term. Such dynamic control laws contain a number of parameters to be tuned to a specific task process. It is a difficult job to find the optimal values in a large parameter space, particularly when all the parameters must be learned on-line in real time. It should be noted that a failure in high speed assembly may incur serious damage to the robot as well as to the parts and the environment. Even for the purpose of learning, fatal mistakes must be avoided at all times. Therefore, we intend to apply progressive learning to cope with these difficulties.

As shown in Figure 4.1-(b), a ball is held with an appropriate impedance. We begin by formulating the impedance control law in accordance with [Hogan, 1985]. The motion of the ball of mass  $m_0$  is governed by the equation of motion given by

$$\mathbf{f} + \mathbf{p} = m_0 \ddot{\mathbf{x}} \quad (4.1)$$

where  $\mathbf{x} = (x, y)^T$  is the position of the ball with an inertial reference,  $\mathbf{p} = (p_x, p_y)^T$  is the contact force acting on the ball, and  $\mathbf{f} = (f_x, f_y)^T$  is the actuator's force to be controlled. The objective of impedance control is to emulate a desired mechanical

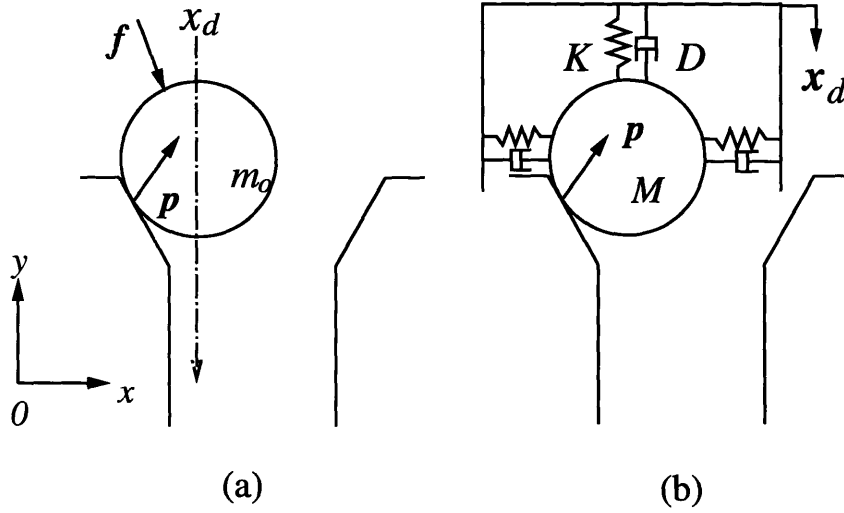


Figure 4.1: Schematic diagram of impedance control

impedance by controlling actuator force  $\mathbf{f}$ . The desired dynamics of the system shown in Figure 4.1-(b) is given by

$$\mathbf{p} = M\ddot{\mathbf{x}} + D(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) + K(\mathbf{x} - \mathbf{x}_d) \quad (4.2)$$

where  $\mathbf{x}_d = (x_d, y_d)^T$  is the nominal trajectory, and  $M$ ,  $D$  and  $K$  are the desired inertia, damping and stiffness matrices respectively. The external force  $\mathbf{p}$  is measured by a force sensor attached to the end effector. From eqs.(4.1) and (4.2), we can derive the impedance control law given by

$$\mathbf{f} = m_0 M^{-1} D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + m_0 M^{-1} K(\mathbf{x}_d - \mathbf{x}) + (m_0 M^{-1} - I)\mathbf{p} \quad (4.3)$$

To formulate a learning algorithm, we need a means for evaluating control performance. In accordance with [Yang and Asada, 1995], we will define a performance index function, referred to as a reinforcement function. In robotic assembly, the objective of control is to mate a part with an uncertain fixture while minimizing the interference and conflict between the part and the fixture. In the ball insertion task,

the robot is required to insert a ball into a hole with a minimum reaction force from the walls. At the same time, the controller must follow the nominal trajectory at least until the ball makes contact with a chamfer or a wall since the nominal hole position is the best initially available conjecture for the real position. Namely, the controller is required to follow the nominal trajectory closely while producing the smallest possible reaction force. Based on these considerations, we define the reinforcement function  $r$ , a performance index for the controller, as follows.

$$r = -[\rho_1 \|\mathbf{x}_d - \mathbf{x}\|^2 + \rho_2 \|\dot{\mathbf{x}}_d - \dot{\mathbf{x}}\|^2 + \rho_3 \|\mathbf{p}\|^2] \quad (4.4)$$

where  $\rho_1$ ,  $\rho_2$  and  $\rho_3$  are weighting factors of the individual terms. The problem is to learn the impedance parameters,  $K$ ,  $D$  and  $M$ , in eq.(4.3) so that the above reinforcement can be maximized. Our approach to this learning problem is to increase the reference motion speed,  $\dot{\mathbf{x}}_d$ , progressively and repeat a learning procedure for different motion speeds so that the impedance matrices can be learned in sequence.

#### 4.2.2 The Progressive Learning Approach: Accommodating the Task Complexity Level by Motion Speed Scheduling

The objective of motion speed scheduling is twofold: one is to prevent the learning process from diverging and the other is to maintain minimum task performance and avoid fatal mistakes and damage. As will be shown later, when a robot attempts to learn all the parameters simultaneously, the process tends to diverge or, even if it converges, searching for the optimal parameters can be a lengthy process. Furthermore, unless appropriate initial parameters are assigned, the control system may become unstable and even dangerous at high speed operations, and this may result in serious damage to the system. Therefore, learning must be initiated at low speed and the motion speed must be increased gradually as the robot gains control

knowledge.

### A. Slow Speed Motion

In progressive learning, we start with a slow speed mainly to excite the quasi-static terms in the impedance control law given in eq.(4.3). A gradient-following learning algorithm is applied to the reinforcement function given in eq.(4.4) to learn the control parameters. In this slow speed learning, only stiffness or compliance terms can gain the most information from the learning process. Since the damping and inertia terms are not fully excited at this slow speed, meaningful values cannot be acquired for those terms.

To execute this learning, initial values for impedance parameters  $K$ ,  $D$  and  $M$  must be provided. These, however, need not to be accurate; one can use some positive matrices so that the robot can track the nominal trajectory  $\mathbf{x}_d$  stably to perform a task at low speed. As learning proceeds, the stiffness parameter  $K$  will be updated toward the optimal stiffness, which maximizes the reinforcement function while changes in  $D$  and  $M$  remain small.

### B. Medium Speed Motion

After the learning curve of stiffness  $K$  has reached a plateau or the reinforcement function has reached a certain threshold level, we increase the motion speed  $\dot{\mathbf{x}}_d$  to a medium speed. As a result, the assembly process becomes more dynamic and the damping term  $D$  in eq.(4.3) now becomes highly excited. We can use the same learning algorithm as in low speed learning to learn the new parameters. The optimal stiffness that was learned at the slow speed learning must be used as the initial values for  $K$  in this phase of learning. The damping and inertial terms in the previous phase are also used as the initial values for  $D$  and  $M$ . In the beginning of learning, the increased speed temporarily deteriorates the performance of the controller, but the

controller adapts itself to the higher speed operation by quickly learning the damping matrix  $D$ . Important to note here is that we can make a smooth transition from the slow motion operation to the medium speed operation by succeeding the previously learned parameters as the initial values for the next learning procedure. In other words, if we were to choose initial values randomly, the controller's performance would be extremely poor at the beginning and, as a result, would cause some damage to the task environment. With progressive increase of motion speed, we can avoid these problems and find the optimal controller faster and more effectively.

### C. High Speed Motion

After learning the optimal damping has been completed, we further increase the motion speed to the highest. This increase of motion speed excites the system in a broader range, and makes the dynamic characteristics more prominent. The inertia term becomes dominant in this phase. As at the medium speed, the impedance matrices acquired in the previous phase are used for the initial impedance matrices. By doing so, we can avoid the instability and damage to the system that could otherwise be caused. After temporary deterioration of performance caused by the increase of the motion speed, the controller will smoothly reach the optimal impedance matrices for the high speed motion.

## 4.3 Mechanisms of Progressive Learning

In this section, we manifest the mechanisms of progressive learning in detail and address the issues of why this learning method works for impedance learning and why stable and smooth convergence can be obtained without fatal mistakes and damage to the system. We analyze the behavior of the progressive learning process for a class of learning algorithms appropriate for the progressive learning of impedance parameters.

### 4.3.1 Gradient Following

In the previous section, the learning problem was formulated as the reinforcement learning of impedance parameters. Most reinforcement learning algorithms are based on a gradient-following method, where the parameters to be learned are updated based on the gradient of a given performance index (reinforcement). In this subsection, we focus on the basic properties of the gradient following learning, and discuss how these properties mesh with the concept of progressive learning, that is, gradual excitation using the scheduled motion speed. Based on this section's analysis, we will develop a specific reinforcement algorithm in the following section.

Figure 4.2 shows the schematic diagram of a gradient following approach to learning a feedback control law. Let  $\mathbf{w} = (w_1, \dots, w_n)^T$  be a set of controller's parameters,  $\mathbf{u}$  be control inputs and  $r$  be a performance index (reinforcement) and  $\mathbf{x}$  be feedback signals. Then, the general rule of gradient following can be expressed as:

$$\Delta \mathbf{w} = \alpha \left( \frac{\partial r}{\partial \mathbf{w}} \right)^T = \alpha \left( \frac{\partial \mathbf{u}}{\partial \mathbf{w}} \right)^T \left( \frac{\partial r}{\partial \mathbf{u}} \right)^T \quad (4.5)$$

where  $\alpha$  is a learning rate.

In general, the effect of control actions on the task process is not exactly known and, therefore, the derivative  $(\partial r / \partial \mathbf{u})^T$  in the above expression is not available. To overcome this problem, various reinforcement algorithms for estimating the derivative have been developed. In this subsection, however, we simply assume that the derivative is known to the system, and the estimation problem of the derivative is rendered to the succeeding subsection. Therefore, the control parameters are updated exactly in the direction given in eq.(4.5).

In order to apply the above gradient following rule to our impedance control problem and derive a learning algorithm for learning impedance, we rewrite the impedance



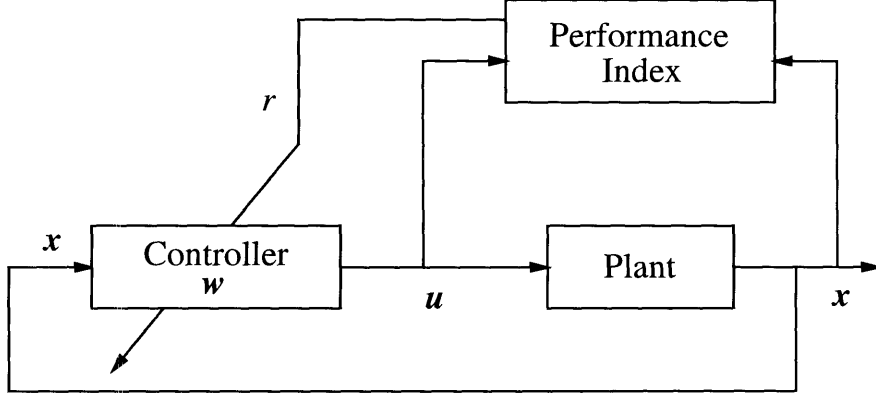


Figure 4.2: Schematic diagram of Gradient Following Approaches

control law given in eq.(4.3) by replacing the inverse of inertia matrix  $M^{-1}$  by  $\bar{M}$ .

$$\mathbf{f} = m_0 \bar{M} K (\mathbf{x}_d - \mathbf{x}) + m_0 \bar{M} D (\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + (m_0 \bar{M} - I) \mathbf{p} \quad (4.6)$$

$$= m_0 \bar{M} K \Delta \mathbf{x} + m_0 \bar{M} D \Delta \mathbf{v} + (m_0 \bar{M} - I) \mathbf{p} \quad (4.7)$$

where  $\Delta \mathbf{x}$  and  $\Delta \mathbf{v}$  are errors of position and velocity, respectively. In this paper, we assume that the impedance matrices are diagonal:

$$K = \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix}, D = \begin{bmatrix} d_x & 0 \\ 0 & d_y \end{bmatrix}, \bar{M} = \begin{bmatrix} \bar{m}_x & 0 \\ 0 & \bar{m}_y \end{bmatrix} \quad (4.8)$$

By applying the gradient following rule in eq.(4.5), we obtain the following learning rules for the impedance parameters:

$$\begin{bmatrix} \Delta k_x \\ \Delta k_y \end{bmatrix} = \alpha_k m_0 \bar{M} \begin{bmatrix} \frac{\partial r}{\partial f_x} & 0 \\ 0 & \frac{\partial r}{\partial f_y} \end{bmatrix} \Delta \mathbf{x} \quad (4.9)$$

$$\begin{bmatrix} \Delta d_x \\ \Delta d_y \end{bmatrix} = \alpha_d m_0 \bar{M} \begin{bmatrix} \frac{\partial r}{\partial f_x} & 0 \\ 0 & \frac{\partial r}{\partial f_y} \end{bmatrix} \Delta \mathbf{v} \quad (4.10)$$

$$\begin{bmatrix} \Delta \bar{m}_x \\ \Delta \bar{m}_y \end{bmatrix} = \alpha_m m_0 \begin{bmatrix} \frac{\partial r}{\partial f_x} & 0 \\ 0 & \frac{\partial r}{\partial f_y} \end{bmatrix} (K \Delta \mathbf{x} + D \Delta \mathbf{v} + \mathbf{p}) \quad (4.11)$$

where  $r$  is the reinforcement defined in eq.(4.4), and  $\alpha_k$ ,  $\alpha_d$  and  $\alpha_m$  are learning rates.

As shown in these equations, changes in the impedance parameters are proportional to the magnitudes of the errors and the reaction force:  $\Delta \mathbf{x}$ ,  $\Delta \mathbf{v}$  and  $\mathbf{p}$ . Namely, the correction of each impedance parameter depends on the magnitude of the corresponding error or reaction force. By combining this algorithm with the motion speed scheduling, we can make the following arguments:

i) The parameter space can be excited selectively by varying the motion speed.

The learning rules given by eqs.(4.9)-(4.11) manifest how the scheduled motion speed selectively excites the individual impedance parameters. When the ball approaches the chamfer as shown in Figure 4.1-(a), the ball's velocity  $\dot{\mathbf{x}}$  is almost the same as its command value  $\dot{\mathbf{x}}_d$ , as long as the system is stable and the ball's motion has reached a steady state. When the motion command is large, a large impact is created by collision with the chamfer. And this large impact creates a large reaction force  $\mathbf{p}$  and deviates the ball's trajectory from the nominal one. The ball may even bounce on the chamfer surface if the impact is very large. The magnitude of the impact is proportional to the ball's approach velocity, which is almost the same as  $\dot{\mathbf{x}}_d$ . Therefore, the impact becomes negligibly small as  $\dot{\mathbf{x}}_d$  becomes small. The low speed motion, however, entails positional deviation  $\Delta \mathbf{x}$  due to the geometric constraint formed by the chamfer. As the ball moves along the chamfer, it deviates from the nominal trajectory. Therefore, even at a low speed where the impact is negligible,  $\Delta \mathbf{x}$  may vary in a broad range and thereby stiffness parameters,  $k_x$  and  $k_y$  in eq.(4.9), are excited. On the other hand, the reaction force and velocity deviation generated by impact are small as low speed, hence the damping and inertia parameters are not fully excited. Note that, according to eq.(4.11), the inertia term may be excited since it depends not only on  $\mathbf{p}$  but also  $\Delta \mathbf{x}$ . However, the excitation is still limited in

magnitude compared with the case where a large impact is generated by collision at high speed. As the motion speed increases,  $\Delta \mathbf{v}$  and  $\mathbf{p}$  become large, hence changes in the damping parameters  $D$  and the inertia parameters  $\bar{M}$  become larger, and these parameters are fully excited. Therefore, a specific region in the parameter space can be excited selectively by varying the motion speed.

ii) Excessive parameter changes can be suppressed.

It is well known that in the gradient following method the learning process may diverge when incremental parameter changes  $\Delta \mathbf{w}$  are too large (see [Acton, 1970] or [Widrow and Stearms, 1985], for example). In our impedance learning problem, if we increase the motion speed abruptly, the impedance controller that has been trained for low speeds may yield large errors and a large reaction force, which may incur divergence due to excessive parameter changes. On the other hand, if the motion speed increases gradually, positional errors and the reaction force can be kept within a small range. Furthermore, the errors and reaction force will decrease as learning proceeds. Therefore, changes in the impedance parameters can be confined to small magnitudes so that the learning process may remain stable. This will help the learning process stay within stable states. Thus, gradually increasing the motion speed may contribute to stabilizing the learning process.

These arguments will be verified through simulation experiments described in Section 4.4.

#### 4.3.2 Local, Progressive Learning of Internal Model

To apply the gradient following rule given by eq.(4.5), the gradient of reinforcement,  $(\partial r / \partial \mathbf{u})^T$ , must be provided. In learning control, however, the process or the plant is not exactly known. Therefore, to fill this void, there must be an

additional mechanism that correlates reinforcement values with control inputs to obtain the gradient information or its equivalent. One approach is to obtain the information equivalent to the gradient in a stochastic manner. For example, the linear random-search algorithm in [Widrow and McCool, 1976] generates a random direction and evaluates the direction at each iteration. The genetic optimizer algorithm in [Etter and Masukawa, 1981] selects a series of random locations in the parameter space to examine the performance surface. These random-search methods have also been applied to reinforcement learning problems. In Stochastic Reinforcement Learning, for example, reinforcement values are stochastically correlated with control parameters by randomly perturbing the control inputs using adaptive probability density functions [Barto, et al., 1983] [Gullapalli, 1990] [Williams, 1992]. These types of learning algorithms, although simple and useful for a class of problems, exhibit extremely slow convergence and erratic behavior due to their trial-and-error natures. These algorithms are not applicable to on-line learning where erratic behavior or total failures are inadmissible, as in the case of robot impedance learning. On the other hand, Model Based Reinforcement Learning uses an internal model network in order to learn the relationship between control inputs and the resultant reinforcement based on available data. As shown in Figure 4.3, in this learning algorithm, the gradient of reinforcement is estimated by feeding back the reinforcement value through the internal model. This learning approach was first suggested by Werbos [Werbos, 1988] and developed independently by Widrow [Widrow, 1986] and Jordan and Rumelhart [Jordan and Rumelhart, 1992]. We summarize the features of this learning approach as well as its drawbacks, and discuss how the drawbacks can be overcome or mitigated by progressive learning.

As shown in Figure 4.3, Model Based Reinforcement Learning uses an additional

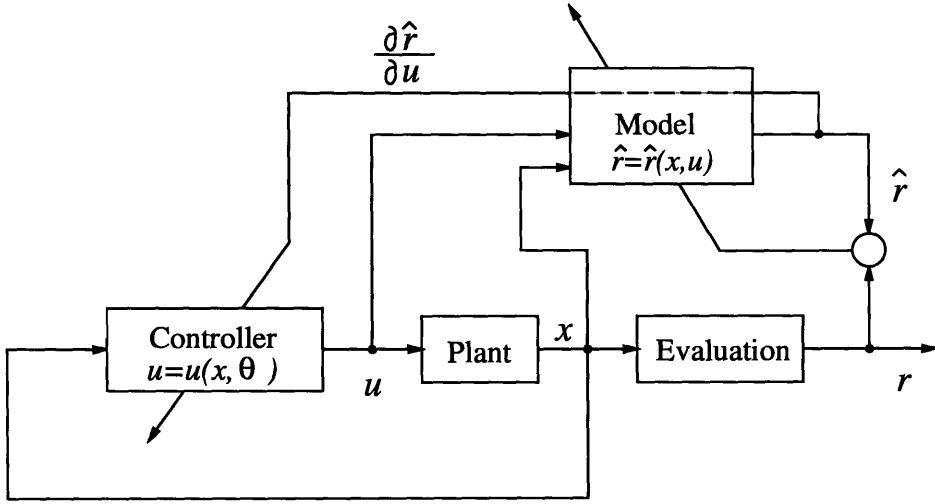


Figure 4.3: Model Based Reinforcement Learning

differentiable network to predict the reinforcement (performance index) function. This model is adaptively identified from input-output observations. Once this model has been identified, the gradient following rule is then applicable to the process. By differentiating the internal model output  $\hat{r}$  in terms of control  $\mathbf{u}$  and replacing the derivative  $\partial r / \partial \mathbf{u}$  in eq.(4.5) by  $\partial \hat{r} / \partial \mathbf{u}$ , we obtain

$$\Delta \hat{\mathbf{w}} = \alpha \left( \frac{\partial \mathbf{u}}{\partial \mathbf{w}} \right)^T \left( \frac{\partial \hat{r}}{\partial \mathbf{u}} \right)^T \quad (4.12)$$

While the stochastic reinforcement algorithms can accomplish the gradient following only in the stochastic sense, the model based approach accomplishes this deterministically, hence it yields much faster convergence. However, identification of the internal model requires a large number of sample data and is extremely difficult especially for a class of tasks where the input dimension of the model is large.

When the internal model is inaccurate, the estimated gradient  $\partial \hat{r} / \partial \mathbf{u}$  has a significant error. As a result, the weight changes  $\Delta \hat{\mathbf{w}}$  may be made in a wrong direction. To evaluate the internal model, let us analyze the influence of the model accuracy upon the

performance index, i.e. the reinforcement. In accordance with [Yang and Asada, 1995], consider the following quantity  $\phi$  termed “improvement in performance” to be made when control parameters are changed from  $\mathbf{w}$  to  $\mathbf{w} + \Delta\hat{\mathbf{w}}$ :

$$\phi(\mathbf{x}) \stackrel{\text{def}}{=} r(\mathbf{x}, \mathbf{u}(\mathbf{x}, \mathbf{w} + \Delta\hat{\mathbf{w}})) - r(\mathbf{x}, \mathbf{u}(\mathbf{x}, \mathbf{w})) \quad (4.13)$$

This quantity manifests whether the parameters have been updated in the right direction for a given  $\mathbf{x}$ . When the progressive learning method is applied, the parameter changes  $\Delta\hat{\mathbf{w}}$  can be kept small, as discussed in the previous section. For small  $\Delta\hat{\mathbf{w}}$ , taking the first order Taylor expansion of  $\phi$  and substituting eq.(4.12) yield

$$\phi \cong \frac{\partial r}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{w}} \Delta\hat{\mathbf{w}} = \alpha \frac{\partial r}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{w}} \left( \frac{\partial \mathbf{u}}{\partial \mathbf{w}} \right)^T \left( \frac{\partial \hat{r}}{\partial \mathbf{u}} \right)^T \quad (4.14)$$

The physical meaning of  $\phi$  is the inner product of the estimated gradient,  $\left( \frac{\partial \hat{r}}{\partial \mathbf{u}} \right)^T = \left( \frac{\partial \mathbf{u}}{\partial \mathbf{w}} \right)^T \left( \frac{\partial \hat{r}}{\partial \mathbf{u}} \right)^T$ , and the true gradient,  $\left( \frac{\partial r}{\partial \mathbf{w}} \right)^T = \left( \frac{\partial \mathbf{u}}{\partial \mathbf{w}} \right)^T \left( \frac{\partial r}{\partial \mathbf{u}} \right)^T$ . In order to maintain the improvement  $\phi$  always positive, the derivative of the internal model must be accurate enough to make this inner product positive at all times. Namely, the internal model must be accurate not only in terms of its outputs, but also in terms of the derivative of the output with respect to control  $\mathbf{u}$ . This condition is difficult to satisfy unless a large number of sample data collected from the whole input space are available and the model is fully trained to generate a smooth function. This requirement is not achievable or feasible in practice especially when the dimension of the input space is large.

In progressive learning, we solve this problem by gradually and intensively exciting a specific local region of the space. To this end, we construct the model using a locally tunable network such as a radial basis function network [Poggio and Girosi, 1989]. A radial basis function network consists of a linear combination of radial basis functions distributed over the input space. When new samples are presented to the network,

the network is modified only in the vicinity of the sample points, leaving the network weights in the other regions unchanged. By incorporating this feature of locally tunable networks into the excitation scheduling technique of progressive learning, we can obtain the following significant advantages:

i) High sample density

In progressive learning, the parameter space can be excited locally and selectively, as addressed in the previous subsection. Likewise, the internal model can be trained locally with many samples only within a local region. Therefore, the density of sample points may be high enough to generate an accurate model.

ii) Small input range

As learning proceeds, errors in  $\mathbf{x}$  and  $\mathbf{v}$  as well as the magnitude of force  $\mathbf{p}$  decrease. Also, since the motion speed increases gradually,  $\Delta\mathbf{x}, \Delta\mathbf{v}$  and  $\mathbf{p}$  remain small as mentioned before. Therefore, the internal model does not have to cover distal points from the origin; the dynamic ranges of  $\Delta\mathbf{x}$ ,  $\Delta\mathbf{v}$  and  $\mathbf{p}$  are not large. The internal model must be accurately trained merely in the vicinity of the origin. Thus, the training may progress quickly.

Because of these features attributed to progressive learning and locally tunable networks, the learning of internal model can be performed accurately and quickly so that the performance improvement quantity  $\phi$  may be kept positive. As long as the motion speed is increased gradually, the internal model can be trained in real-time in the gradually-expanded space.

## 4.4 Implementation and Simulation

In this section, we implement the above progressive learning of impedance parameters and conduct simulation experiments to demonstrate the effectiveness of the

proposed method. Various simulation results are then provided to verify the arguments given above.

#### 4.4.1 Implementation

In this thesis, we implement the progressive learning method by using the Adaptive Reinforcement Learning Algorithm(ARL) [Yang and Asada, 1995]. The ARL algorithm applies a perturbation/correlation technique to learning an internal model. The estimated gradient and parameter changes based on the internal model are less erratic and more robust against uncertainties and noise compared with the general model-based learning algorithms. The details of the ARL algorithm are provided in [Yang and Asada, 1995] and its application to learning static compliance for a simple assembly task is found in [Yang and Asada, 1993-(b)]. In the ARL algorithm, a radial-basis function network is used to represent an internal model. As discussed in the previous section, locally tunable networks “synergize” progressive learning, since the system is excited locally and gradually. Even if the accuracy of the model network is limited at the beginning, the accuracy can be quickly improved during on-line learning due to scheduled excitation.

Figure 4.4 shows the configuration of the task environment for simulation. As shown in the figure, we use a spring-damper model to simulate the chamfer surface and compute the impact of collision between the ball and the chamfer surface. With this model, we can calculate the contact force or impact force  $p_n$  in the direction normal to the chamfer surfaces. Friction  $p_t$  on the chamfer surfaces is given by  $p_t = \mu p_n$ , where  $\mu$  is a friction coefficient.

In each learning iteration, the controller is given a nominal trajectory  $\mathbf{x}_d$ . The trajectory is parallel to but deviated randomly from the center line of the hole so



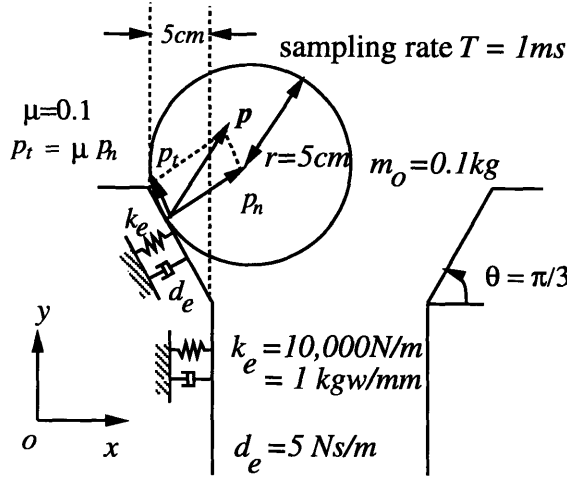


Figure 4.4: Task environment of the ball insertion

that the ball always collides with the chamfer surface. The impedance parameters involved in the controller are tuned in real time using the Adaptive Reinforcement Learning Algorithm. In this simulation experiment, we conducted 300 iterations of learning divided into three phases: the first 100 iterations in a slow speed, the second 100 iterations in a medium speed and the last 100 iterations in a high speed.

#### 4.4.2 Simulation Results

Figure 4.5 shows the transitions of the impedance parameters as well as the reinforcement over the whole learning iterations. Note that for each iteration we averaged the reinforcement over the time period of the iteration. Note also that the erratic behavior of the learning curves is caused by the randomness involved in each iteration as well as in the Adaptive Reinforcement Learning Algorithm. The details of the learning procedure follow.

As described in the previous section, learning started with a slow motion speed to

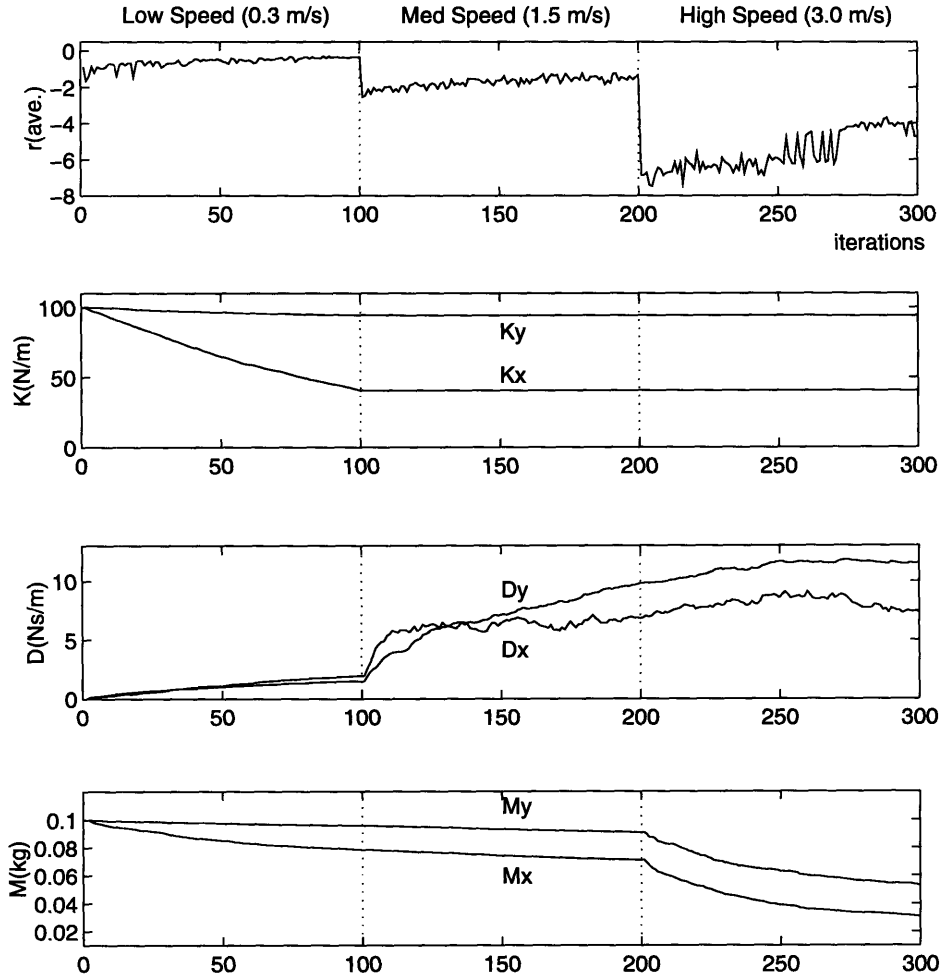


Figure 4.5: Learning histories of reinforcement  $r$  and stiffness  $K$ , damping  $D$  and inertia matrices  $M$  over 300 iterations

excite only the quasi-static terms of the system. The velocity command  $\dot{x}_d$  was set to  $(0, -0.3 \text{ m/s})^T$ . The initial stiffness matrix was given by:

$$K_0 = \begin{bmatrix} 100 \text{ N/m} & 0 \\ 0 & 100 \text{ N/m} \end{bmatrix} \quad (4.15)$$

These stiffness parameters, or position gains, are large enough to follow the desired trajectory. The damping matrix  $D$  was initialized with small positive values and the

inertia matrix was initialized as the plant inertia matrix  $m_0I$ . Figure 4.6-(a) shows the performance of the initial stiffness controller. At a glance, it may be seen that the robot can successfully insert the ball into the hole. However, as seen in the plots of  $p_x$  and  $p_y$ , the reaction forces in both the  $x$  and  $y$  directions are large. Furthermore, since we have only position gains, the ball cannot track the desired trajectory smoothly, resulting in the zigzag motion. Learning is necessary to improve the performance of the controller.

One hundred iterations of learning were performed in this phase. As shown in Figure 4.5, the stiffness in the  $x$  direction decreased remarkably to  $47.41N/m$  while that in the  $y$  direction decreased just slightly to  $96.3N/m$  during the 100 iterations. This shows that the robot has to hold the ball compliantly in the  $x$  direction to comply to the chamfer surface and stiffly in the  $y$  direction to follow the nominal trajectory toward the bottom of the hole. This result is compatible with the argument that was derived by [Witney, 1977] and other researchers. On the other hand, the damping and inertia parameters are almost unchanged, showing that these terms are not excited during this slow speed operation. From this observation, we can draw the conclusion that in the slow speed operation the damping and inertia parameters cannot gain useful information. The average of the reinforcement values converged to -0.5. Figure 4.6-(b) shows the performance of the controller at the 100th iteration. As shown in the figure, the reaction forces are sufficiently suppressed and the motion of the ball is quite smooth compared with the initial trajectory.

Secondly, we increased the motion speed to  $\dot{x}_d = (0, -1.5 \text{ m/s})^T$  to excite the system more dynamically, and conducted another 100 learning iterations. As shown in Figure 4.5, in the beginning of this learning phase, the reinforcement value decreased discontinuously from -0.5 to -2.6 due to the increase of speed. Namely, the controller

trained in the slow speed operation could not perform satisfactorily in this faster motion. However, as learning proceeded, the performance was quickly improved and the reinforcement again started to increase. In this phase, the damping parameters increased steadily and significantly while the stiffness parameters remained almost constant over the iterations. It appears that the stiffness parameters had already been learned in the first learning phase and the controller did not gain any additional information at the faster motion speed. The inertia matrix  $M$  was again almost unchanged in this learning phase.

Finally, we increased the motion speed to the maximum  $\dot{\mathbf{x}}_d = (0, -3.0 \text{ m/s})^T$ . When the task was first performed at maximum speed, the control parameters were succeeded from the previous ones obtained for the medium speed. As shown in Figure 4.7-(a), however, there is a large amount of bouncing on the chamfer surface resulting in huge impact forces. In order to reduce the bouncing and the impact forces, the inertia term must be significantly modified. As shown in Figure 4.5, at this motion speed, the inertia terms were very vigorously excited and the corresponding parameters moved most significantly and converged to the final values. The damping parameters also varied in this learning phase while the stiffness parameters again remained unchanged. Note that after the learning the inertia in the  $x$  direction became much smaller than that in the  $y$  direction, which is also compatible with the theoretical conclusion derived in [Asada and Kakumoto, 1990]. Figure 4.7-(b) shows that the learned impedance controller allows for successful insertion at high speed without a large impact and bouncing on the chamfer.

As shown above, by increasing the motion speed progressively, the controller can learn a better impedance smoothly and stably while maintaining a minimum performance level. To compare the progressive learning method with its traditional coun-

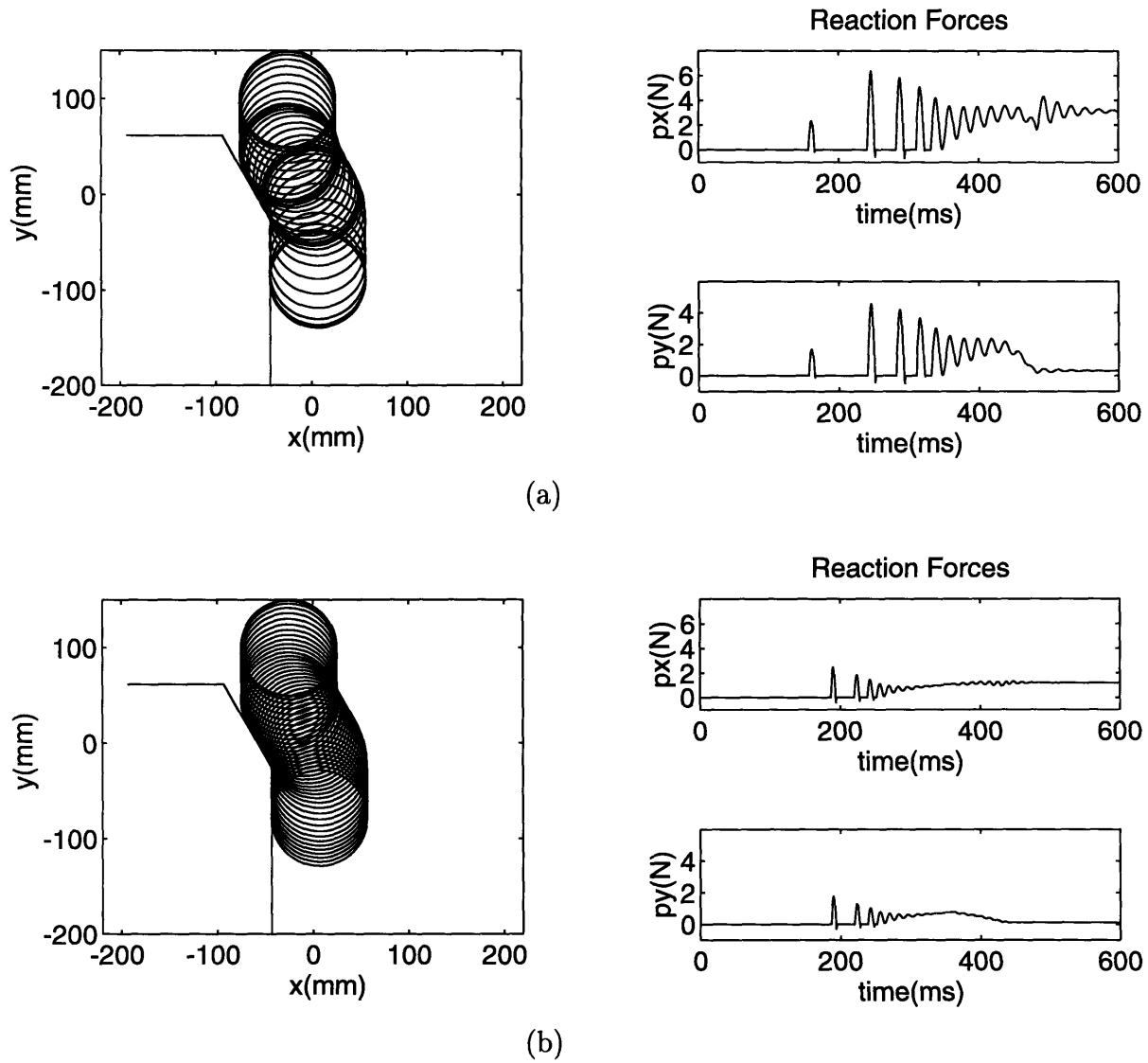
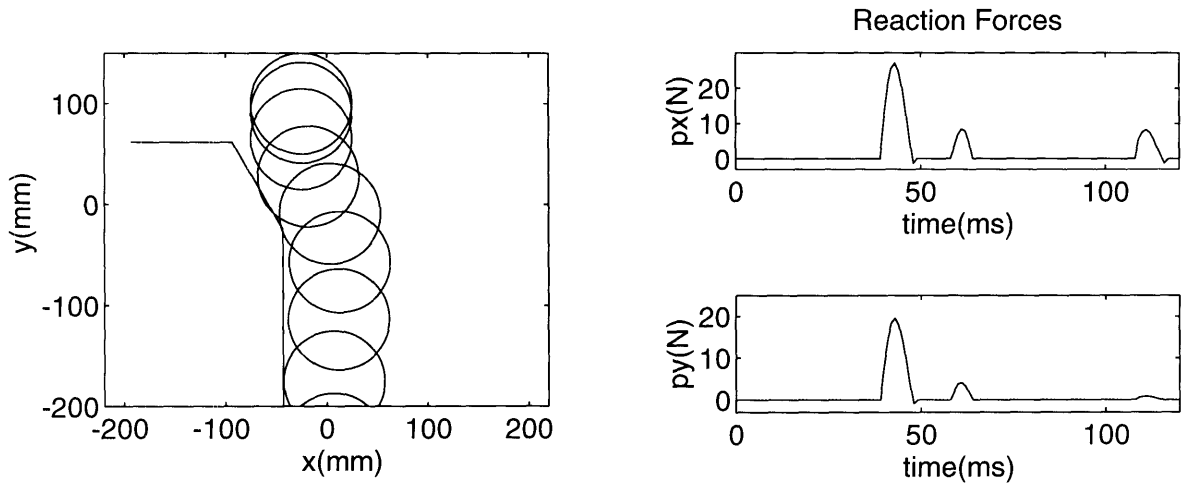
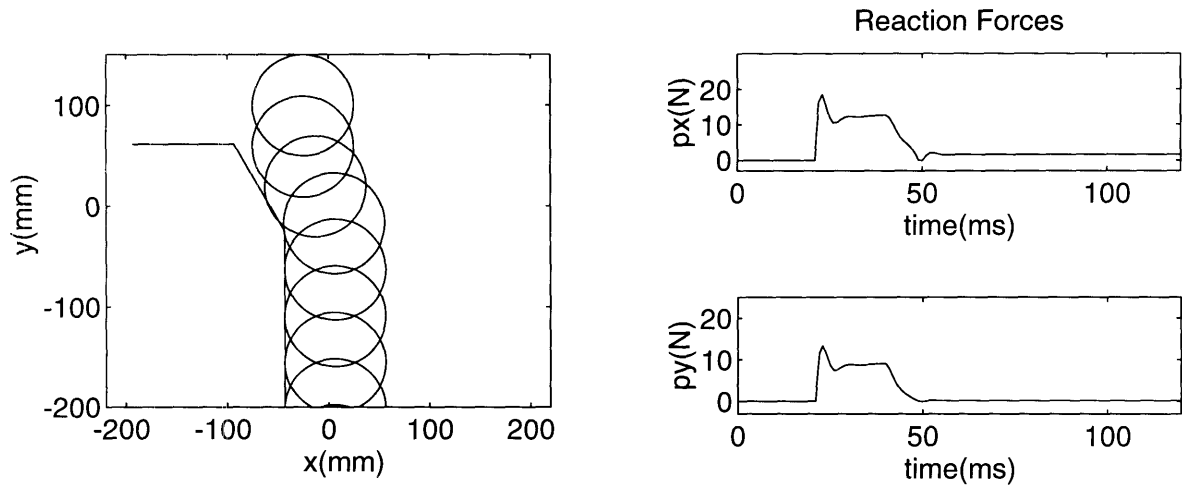


Figure 4.6: Task performances before learning (a) and after 100 learning trials (b)

terpart, we conducted another set of learning simulations. In the traditional non-progressive learning method, a constant motion speed, 3.0 m/s, was used throughout the learning operations. Figure 4.8 shows the learning curves of the two methods. As shown in the figure, the control performance i.e. the reinforcement value, is erratic



(a)



(b)

Figure 4.7: Task performance at high speed before learning inertia  $M$  (a) and after (b)

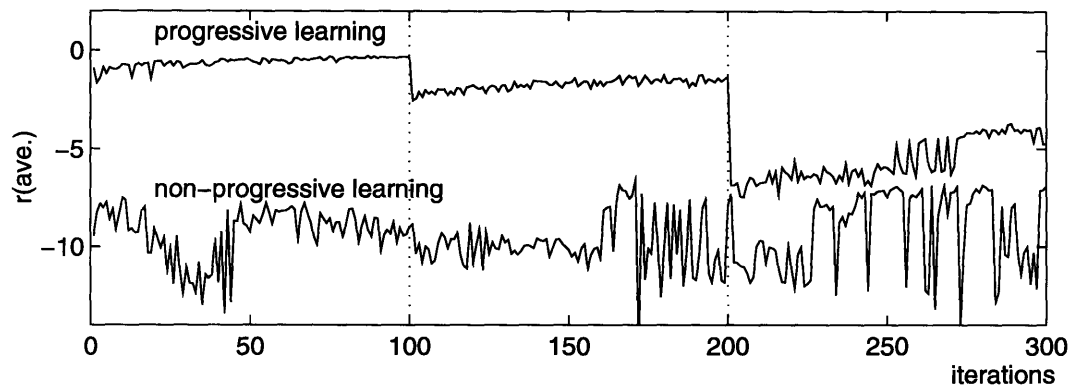


Figure 4.8: Comparison between progressive and non-progressive learning methods

and significantly lower than that of the progressive learning method. Important to note is that the progressive learning method allows the controller to maintain a certain performance level throughout all the learning iterations while non-progressive learning sometimes exhibits intolerable, poor performance, which may incur significant damage to the system.

#### 4.4.3 Verifying the Features of Progressive Learning

In the previous section, we addressed the salient features and advantages of progressive learning. The progressive learning method based on motion speed scheduling mitigates the difficulties of gradient following learning and overcomes the critical problem of model-based reinforcement learning. Overall, the progressive learning method and the model-based, gradient following method work synergistically and accomplish stable learning without fatal damage and poor performance throughout the learning process. We will now verify, through simulation experiments, every argument in favor of progressive learning addressed in the previous section.

### *A. Gradual excitation of the parameter space*

We first consider the fact that in progressive learning the parameter space is gradually excited as the motion speed increases. This argument is clearly supported by the simulation result shown in Figure 4.5. This figure shows that, at the initial low speed, changes in  $D$  and  $M$  are insignificant compared with those in  $K$ . As the motion speed increases,  $D$  and  $M$  become more excited and vary significantly, while  $K$  remains unchanged.

### *B. Suppression of excessive parameter changes*

We have addressed the fact that by keeping errors  $\Delta\mathbf{x}$  and  $\Delta\mathbf{v}$  as well as reaction force  $\mathbf{p}$  small, stable learning can be accomplished. To verify this argument, the magnitudes of the errors and reaction force generated during the previous simulation experiment are shown in Figure 4.9. To aggregate the data, the average  $y$  axis errors over a complete insertion process are plotted against the number of iterations. As shown in this figure, in progressive learning, the errors and reaction force are consistently smaller than those in non-progressive learning. Since we increased the motion speed progressively, these errors and the reaction force were confined to small magnitudes. This contributed to stabilizing the learning process. On the other hand, in non-progressive learning, the magnitudes of the errors and reaction force remained large throughout the process. Consequently, changes in the corresponding parameters,  $\Delta\mathbf{w}$ , became large and the learning process did not converge, as previously shown in Figure 4.8.

### *C. Accurate internal model to assure positive performance improvement*

In the simulation experiment, we also evaluated the accuracy of the internal model by calculating the average improvement in performance  $\phi$  and compared the result



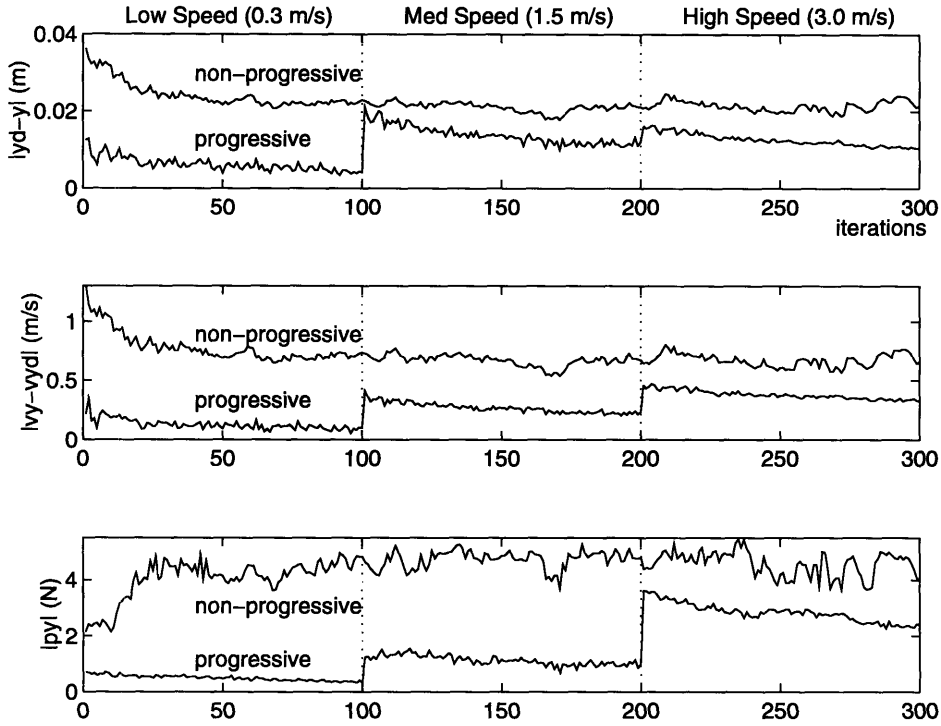


Figure 4.9: Comparison of the magnitudes of errors and reaction force between progressive learning and non-progressive learning

with the one based on non-progressive learning. As discussed in Section 4, the improvement in performance is highly dependent on the accuracy of the model. As shown in Figure 4.10-(b), progressive learning allows effective learning of the internal model so that the value of  $\phi$  is always positive. Namely, performance was improved every time the learning process was repeated. In contrast, the value of  $\phi$  often became negative for non-progressive learning as shown in Figure 4.10-(b). Namely, the internal model cannot be learned effectively using the traditional learning method.

*D. High sample density for training the internal model in a gradually-expanded region*

We have considered the fact that progressive learning allows the internal model to

be trained locally and efficiently with a high sample density in a limited region. To verify this argument, we plotted sample points on a  $\Delta x - \Delta v_x$  plane obtained from a learning process. Figure 4.11-(a) shows distributions of samples (circles in the figure) acquired from three insertion trials during the slow speed learning phase. As seen in the figure, the samples are confined to a small region, especially concentrated on the  $\Delta v_x$  axis. This high sample density allows the internal model to be trained accurately and quickly in this limited region. When the motion speed was increased to medium speed, the sample distribution was expanded as illustrated by circles in Figure 4.11-(b). Note that the sample distribution would be expanded more abruptly to a larger region were the motion speed not increased gradually. With progressive learning, the valid region, that is, the accurately trained region, can be smoothly expanded to a broader area. As learning proceeded, the errors and the reaction force became small and the sample distribution became more concentrated as shown by  $\mathbf{x}$ , in the figure. At the end of the medium speed phase, the samples were distributed only within a small region as shown by the dots and, as a result, the accuracy of the model in that region was further improved. Consequently, by progressively increasing the motion speed, the internal model does not have to cover distal regions from the origin and, therefore, can maintain high accuracy.

## 4.5 Strategies for Increasing the Motion Speed

The main feature of progressive learning is to increase motion speed gradually. In the simulation experiments, our strategy for increasing motion speed was to use a predetermined schedule of motion speeds. In the strategy, the insertion speed is increased after every 100 learning iterations up to the maximum speed. As shown in the previous simulation results, the change of insertion speed caused an abrupt deterioration in performance, though the performance was quickly improved. In this section,

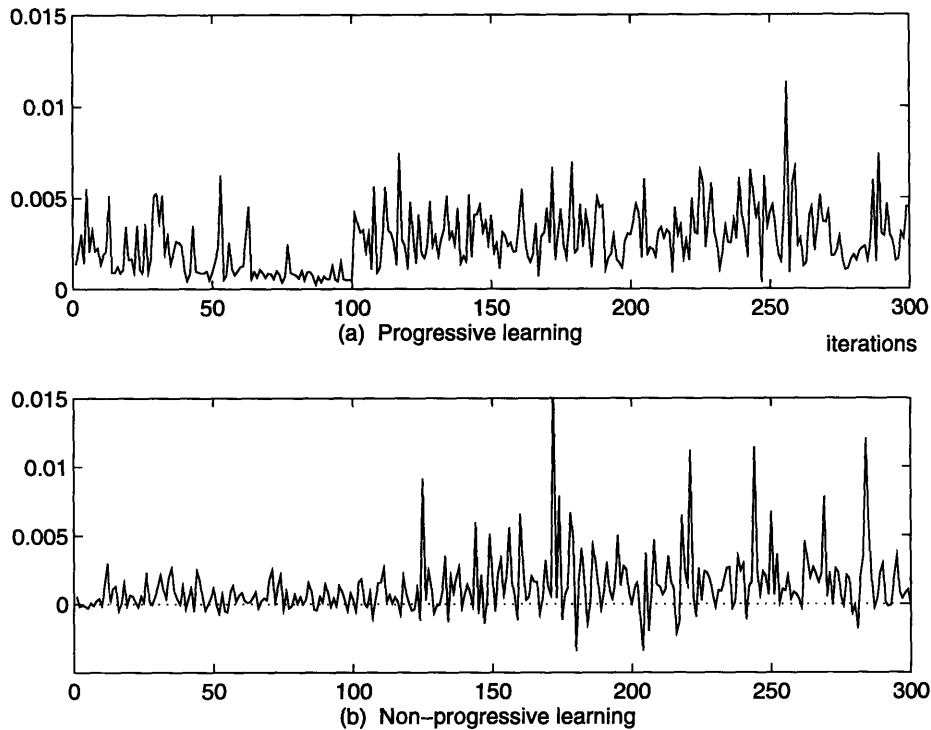


Figure 4.10: Comparison of  $\phi$  between progressive learning (a) and non-progressive learning (b)

to suppress abrupt deterioration, we present a different strategy where the motion speed increases more often with a smaller increment of motion speed. Figure 4.12 shows the results of the learning simulation using the new motion speed scheduling. In the simulation, learning started with the lowest speed (0.3 m/s) as in the previous simulation. However, after every 30 iterations, the motion speed was increased by a small increment and reached the highest speed(3.0 m/s) after 10 increases. As shown in the figure, compared with the original motion scheduling (broken lines), the new motion speed scheduling allows for steadier and smoother learning, especially at high speeds. Namely, the performance deterioration caused by the increase of motion speed

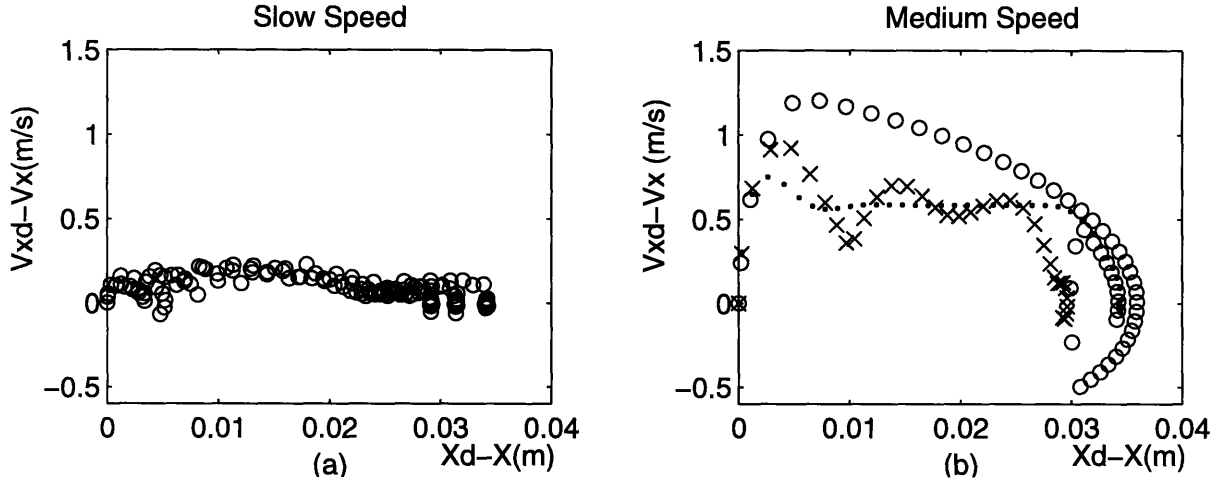


Figure 4.11: Distribution of sample data in  $\Delta x - \Delta v_x$  plane

is successfully suppressed in this new strategy. Note that the changes in the stiffness parameters are much less than the previous results, since the motion speed increases quickly and, therefore, the excitation of the stiffness term fades out in the early stage of this learning process compared with the previous one.

In the above strategies of predetermined motion speed scheduling, the motion speed was increased periodically regardless of the progress in learning. These strategies are often inefficient since, although learning at low speeds was much quicker than that at high speeds, the same number of iterations was performed. It is also difficult to predict the number of iterations needed for each motion speed. To avoid unnecessary repetitions and accelerate the convergence of the whole learning process, we need to monitor the progress of learning and increase the motion speed accordingly, as shown by the second feedback loop in Figure 2. In the following simulation, we implemented the new strategy of changing the motion speed according to the progress in learning. In this simulation, reinforcement values are examined after every 10 iterations and

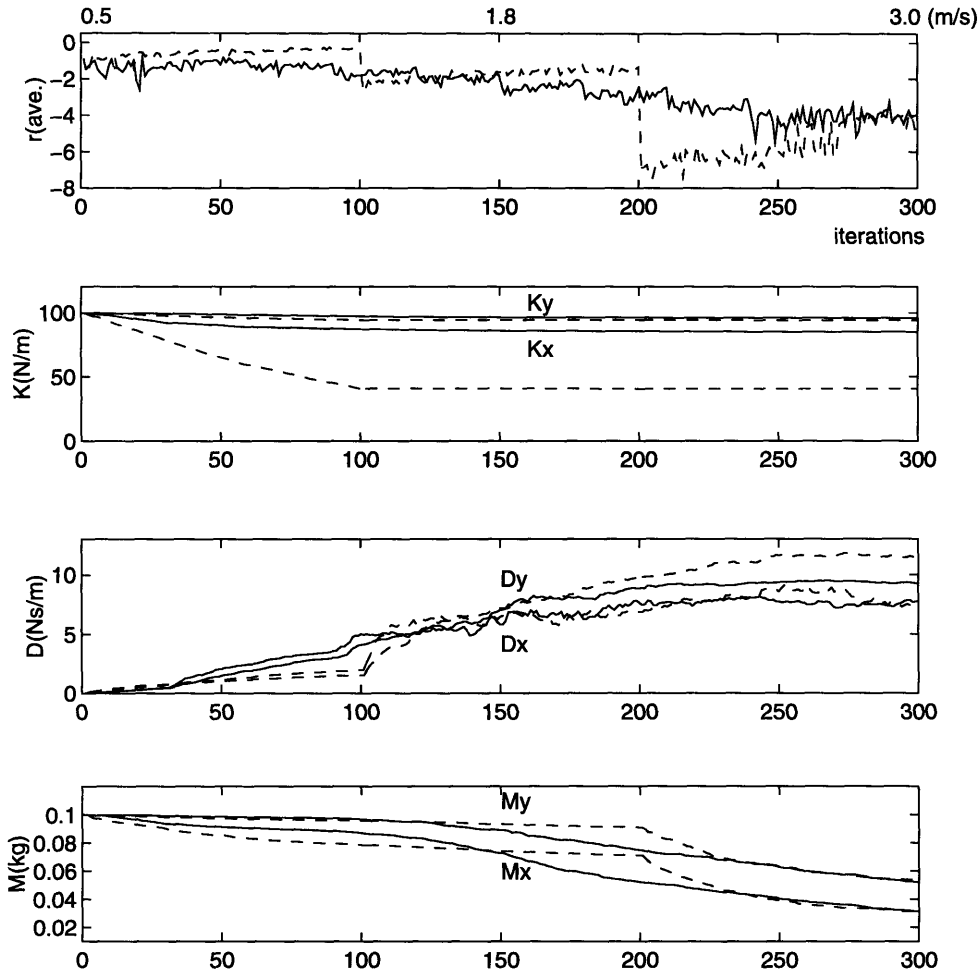


Figure 4.12: Learning histories of reinforcement  $r$  and stiffness  $K$ , damping  $D$  and inertia matrices  $M$  over 300 iterations with a smaller motion speed increment. Broken lines shows the results of the previous progressive learning simulation.

the motion speed is incremented by 0.45 m/s if the increase in the average of the reinforcement values over the last 5 iterations from that of the first 5 iterations is less than 5 %. Figure 4.13 shows the simulation results. As shown in the figure, an important result of this strategy is that low speed learning requires much less learning iterations than the higher speed learning phases. Namely, by reducing the number

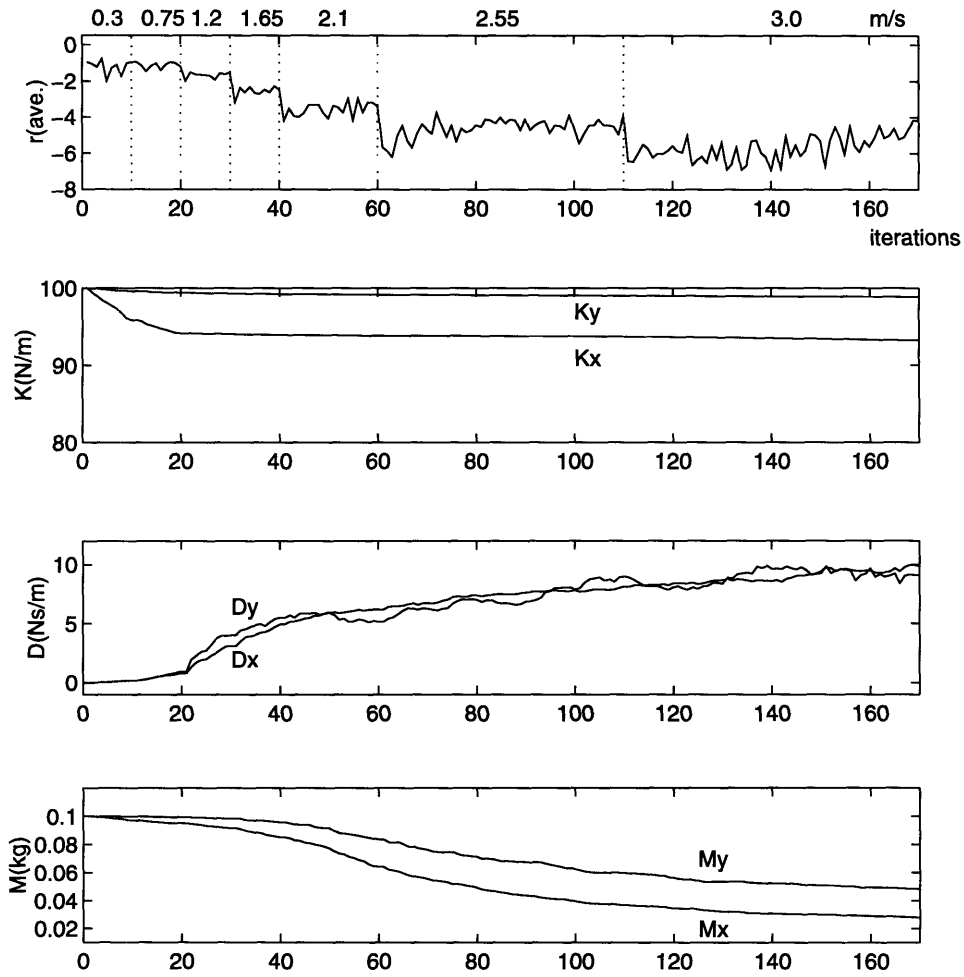


Figure 4.13: Learning histories of reinforcement  $r$  and stiffness  $K$ , damping  $D$  and inertia matrices  $M$  with the new strategy of motion speed scheduling

of iterations significantly at slower motion speeds and increasing the ones at higher motion speeds, we can achieve much faster convergence of the whole learning process while maintaining a certain performance level.

## 4.6 Conclusion

In this chapter, we have applied the concept of progressive learning to a high-speed assembly task and proposed a scheduled excitation method by varying a motion speed command. Based on the progressive learning method, we developed an impedance learning algorithm for high speed insertion. We also manifested the mechanisms of progressive learning and examined why the progressive learning method works successfully for impedance learning. The progressive impedance learning method was implemented by using the adaptive reinforcement learning algorithm, and simulation experiments were conducted to show the effectiveness of the progressive learning method. Different strategies for motion speed scheduling were considered at the end.

## **5.1 Thesis Summary**

This thesis has presented a novel approach to stable learning control, termed “Progressive Learning.” The basic concept of progressive learning stemmed from studies on human learning behavior. Humans perform unfamiliar tasks slowly and meticulously when their knowledge is limited and stringent task specifications must be met. The required level of task performance is compared with their competence to perform the task, and, if difficult to execute, the task complexity is lowered by reducing execution speed, relaxing some conditions, or limiting the scope of the task. As humans gain experience and become familiar with the new task, they increase the task execution speed, or attempt to deal with tougher conditions and a broader range of situations. People continue to learn the task by progressively increasing the level of task complexity and thereby improving the task performance ability. The underlying objective of the thesis is to make machine learning as effective as the human learning. Progressive learning uses scheduled excitation inputs that allow to learn quasi-static, slow modes in the beginning, followed by the learning of faster modes. With the progressive learning method, machine learning can be performed smoothly, quickly and stably without incurring fatal mistakes.

Progressive learning was first defined to be a learning system in which the level of task complexity is gradually increased in accordance with the progress of learning so that minimum task performance requirements can be met throughout the learning process and that the learning process may not diverge as the level of task complexity



increases.

To prove stability and convergence of the progressive learning system, a theory of progressive learning was developed by formulating a gradient based, model reference adaptive control problem. It is well known that in a model reference adaptive control system an excitation at a high frequency causes instability to the system when the relative order of the plant is high and the SPR(strictly positive real) condition is not met. To derive a stability analysis for progressive learning, we applied a method of averaging analysis to describe the behavior of the adaptive system in the frequency domain. Based on this analysis, we proved that the stable convergence of control parameters is guaranteed if the system is excited gradually through the reference input in accordance to the progress of the adaptation. A numerical example was provided to verify the above analysis.

Finally, the concept of progressive learning was applied to robotic assembly to explore the possibility of progressive learning. We used a high speed insertion task as an example, where an impedance control law is learned with the excitation scheduling method. In this method, learning starts with a slow, quasi-static motion and goes to a fast, dynamic motion. During the learning process, the stiffness terms of the impedance controller are learned first, followed by the damping terms and finally by the inertial terms. Consequently, this progressive learning method enables the learning of high-speed dynamic control laws without instability and fatal damage due to high speed collisions. The mechanism of progressive learning was also discussed in detail and verified through simulation experiments.

## 5.2 Contributions

### 5.2.1 Theory of Progressive Learning

A stability analysis of progressive learning was developed by formulating gradient-based, high-relative order model reference adaptive control problem. In the analysis, we proved that an adaptive control system is stable if the system is excited through a reference input within a particular frequency range in which the difference between the phase shifts of the referenced model and the closed-loop system is less than 90 degree. Namely, it was shown that the adaptive system can avoid instability even when the SPR conditions are not met, if the reference input is designed so that its frequency contents are within the particular frequency range. We defined the frequency range as the frequency range for stability. Since the phase shift of the closed-loop system varies depending on the control parameters, another important implication of the above stability analysis is that the frequency range for stability depends only on the control parameters. In other words, the frequency stability range becomes wider as the control parameters approach the optimal parameters, and once the parameters reach the neighbor of the optimal ones the stability range covers the whole frequency range. We finally proved that for given plant, reference model, adaptation rule, and initial parameter values, there always exist a sequence of excitation frequencies such that the system is maintained stable and the control parameters converge to the optimal ones and the output error converges to zero.

### 5.2.2 Application to Robotic Assembly

The concept of progressive learning was applied to a non-linear, more complex problem such as robotic assembly and a new approach to learning control using an excitation scheduling technique was developed. Similarly to the theory of progres-

sive learning, the learning problem was first formulated as a model-based, gradient following reinforcement learning. It was shown that the progressive learning method allows the suppression of excessive parameter changes by increasing the motion speed gradually based on the progress of the learning, and thereby stabilizes the learning process using gradient following. Moreover, it was revealed that the progressive learning method allows the learning of an accurate internal model correlating the reinforcement to control inputs and plant outputs. By gradually increasing the motion speed command, the internal model as well as the control parameters can be learned effectively within a focused, local area in the large parameter space, which is then gradually expanded as the motion speed increases. This mechanism of progressive learning was verified through simulation experiments. Several different strategies for motion speed scheduling were addressed and it was found that we can achieve much faster convergence of the whole learning process while maintaining a certain performance level by reducing the number of iterations significantly at slower motion speeds and increasing the ones at higher motion speeds.

## Appendix A

# Derivation of the Sensitivity Vector

---

In this appendix, we derive the expression of the sensitivity vector given in eq.(3.24). All the notations and definitions are given in the section. In section 2, the sensitivity vector  $\psi(t)$  is defined as

$$\psi(t) = \frac{\partial e_1^T}{\partial \phi}. \quad (\text{A.1})$$

Since  $e_1 = y_p - y_m$  and  $\phi = \theta - \theta^*$  and  $y_m$  is independent of  $\theta$ , the sensitivity vector can be rewritten as

$$\psi(t) = \frac{\partial y_p^T}{\partial \theta}. \quad (\text{A.2})$$

From eq.(3.12),  $y_p$  can be expressed as

$$y_p = W_\theta(s)r \quad (\text{A.3})$$

$$= \frac{k k_p Z_p(s) \lambda(s)}{(\lambda(s) - C(s))R_p(s) - k_p Z_p(s)D(s)} r. \quad (\text{A.4})$$

Also, we can express  $u$  as

$$u = W_p^{-1} y_p = W_p^{-1} W_\theta r \quad (\text{A.5})$$

$$= \frac{k R_p(s) \lambda(s)}{(\lambda(s) - C(s))R_p(s) - k_p Z_p(s)D(s)} r. \quad (\text{A.6})$$

Let  $\chi$  be a vector of differential operators such as

$$\chi \stackrel{\text{def}}{=} [1, s, s^2, \dots, s^{n-2}]^T. \quad (\text{A.7})$$

Then, the polynomials  $C(s)$  and  $D(s)$  given in eqs.(3.10) and (3.11) can be rewritten as

$$C(s) = \chi^T \theta_1 \quad (\text{A.8})$$

$$D(s) = \theta_0 \lambda(s) + \chi^T \theta_2. \quad (\text{A.9})$$

$w_1(t)$  and  $w_2(t)$  in the regressor vector  $w(t)$  also can be rewritten as

$$w_1 = \frac{\chi}{\lambda(s)}u \quad (\text{A.10})$$

$$w_2 = \frac{\chi}{\lambda(s)}y_p. \quad (\text{A.11})$$

Using the above equations, we obtain the partial derivative of  $y_p$  in terms of each control parameter of  $\theta$  as follows:

$$\frac{\partial y_p}{\partial k} = \frac{k_p Z_p \lambda}{(\lambda - C)R_p - k_p Z_p D} r = \bar{W}_\theta r \quad (\text{A.12})$$

$$\frac{\partial y_p^T}{\partial \theta_1} = \frac{k k_p Z_p \lambda R_p f}{((\lambda - C)R_p - k_p Z_p D)^2} r = \frac{k_p Z_p \chi}{(\lambda - C)R_p - k_p Z_p D} u = \bar{W}_\theta w_1 \quad (\text{A.13})$$

$$\frac{\partial y_p}{\partial \theta_0} = \frac{k k_p Z_p \lambda k_p Z_p \lambda}{((\lambda - C)R_p - k_p Z_p D)^2} r = \frac{k_p Z_p \lambda}{(\lambda - C)R_p - k_p Z_p D} y_p = \bar{W}_\theta y_p \quad (\text{A.14})$$

$$\frac{\partial y_p^T}{\partial \theta_2} = \frac{k k_p Z_p \lambda k_p Z_p f}{((\lambda - C)R_p - k_p Z_p D)^2} r = \frac{k_p Z_p \chi}{(\lambda - C)R_p - k_p Z_p D} y_p = \bar{W}_\theta w_2. \quad (\text{A.15})$$

Finally, from the above equations, the sensitivity vector  $\psi(t)$  is given by

$$\psi(t) = \bar{W}_\theta(s)w. \quad (\text{A.16})$$

## Appendix B

# Derivation of the Covariance Matrix

---

The covariance matrix  $R_{w_f w_\theta}(0)$  can be expressed using the cross spectral measure  $S_{w_f w_\theta}(d\omega)$  as

$$R_{w_f w_\theta}(0) = \frac{1}{2\pi} \int S_{w_f w_\theta}(d\omega). \quad (\text{B.1})$$

Namely, the objective of this appendix is to derive an expression of  $S_{w_f w_\theta}(d\omega)$  in terms of the spectral measure of reference  $S_r(d\omega)$  by using the General Harmonic Analysis [Sastry and Bodson, 1989].

First, since  $w = H_{wr}(s, \theta)r$ , we get the spectral measure of  $w$  as

$$S_w(d\omega) = H_{wr}^*(j\omega, \theta) H_{wr}^T(j\omega, \theta) S_r(d\omega). \quad (\text{B.2})$$

Also, since  $w_f = W_m w$ ,

$$S_{w_f}(d\omega) = W_m^*(j\omega) W_m^T(j\omega) S_w(d\omega). \quad (\text{B.3})$$

Since  $w_\theta = \frac{\bar{W}_\theta}{W_m} w_f$ , the cross spectral measure  $S_{w_f w_\theta}(d\omega)$  is given by

$$S_{w_f w_\theta}(d\omega) = \left(\frac{\bar{W}_\theta}{W_m}\right)^* S_{w_f}(d\omega) \quad (\text{B.4})$$

$$= \bar{W}_\theta^* W_m S_w(d\omega) \quad (\text{B.5})$$

$$= H_{wr}^*(j\omega, \theta) H_{wr}^T(j\omega, \theta) \bar{W}_\theta^* W_m S_r(d\omega). \quad (\text{B.6})$$

Next, we need to express  $H_{wr}(j\omega, \theta)$  in terms of  $H_{w_m r}(j\omega)$ . First, let us define the following variables:

$$\bar{\phi} \stackrel{\text{def}}{=} [\theta_1^T - \theta_1^{*T}, \theta_0 - \theta_0^*, \theta_2 - \theta_2^{*T}]^T \quad (\text{B.7})$$

$$\phi_k \stackrel{\text{def}}{=} k - k^* \quad (\text{B.8})$$

$$\bar{w} \stackrel{\text{def}}{=} [w_1^T, y_p, w_2^T]^T \quad (\text{B.9})$$

$$\bar{w}_m \stackrel{\text{def}}{=} [w_{m1}^T, y_m, w_{m2}^T]^T \quad (\text{B.10})$$

$$\bar{H}_{wr}(s, \theta) \stackrel{\text{def}}{=} \begin{bmatrix} (sI - \Lambda)^{-1} l W_p^{-1} W_\theta \\ W_\theta \\ (sI - \Lambda)^{-1} l W_\theta \end{bmatrix} \quad (\text{B.11})$$

$$\bar{H}_{w_m r}(s) \stackrel{\text{def}}{=} \begin{bmatrix} (sI - \Lambda)^{-1} l W_p^{-1} W_m \\ W_m \\ (sI - \Lambda)^{-1} l W_m \end{bmatrix}. \quad (\text{B.12})$$

From eq.(3.17), we have

$$\bar{H}_{wr}(s, \theta) = \frac{k}{k^*} \frac{\Phi_m}{\Phi_\theta} \bar{H}_{w_m r}(s). \quad (\text{B.13})$$

By multiplying  $r$  from the right for both sides of the above equation, we get

$$\bar{w} = \frac{k}{k^*} \frac{\Phi_m}{\Phi_\theta} \bar{w}_m \quad (\text{B.14})$$

$$= \left( \frac{\phi_k}{k^*} + 1 \right) \frac{\Phi_m}{\Phi_\theta} \bar{w}_m \quad (\text{B.15})$$

$$= \frac{\Phi_m}{\Phi_\theta} \bar{w}_m + \frac{\phi_k}{k^*} \frac{\Phi_m}{\Phi_\theta} \bar{w}_m. \quad (\text{B.16})$$

From the output error equation given in eq.(3.20), we have

$$e_1 = \frac{1}{k^*} W_m \phi^T w = \frac{1}{k^*} W_m (\phi_k r + \bar{\phi}^T \bar{w}). \quad (\text{B.17})$$

We also have

$$e_1 = y_p - y_m = W_\theta r - W_m r \quad (\text{B.18})$$

$$= \frac{k}{k^*} \frac{\Phi_m}{\Phi_\theta} W_m r - W_m r. \quad (\text{B.19})$$

Therefore, we get

$$\frac{k}{k^*} \frac{\Phi_m}{\Phi_\theta} W_m r - W_m r = \frac{1}{k^*} W_m \phi_k r + \frac{1}{k^*} W_m \bar{\phi}^T \bar{w}. \quad (\text{B.20})$$

Since  $1 + \frac{\phi_k}{k^*} = \frac{k}{k^*}$ ,

$$\frac{k}{k^*} W_m r = \frac{k}{k^*} \frac{\Phi_m}{\Phi_\theta} W_m r - \frac{1}{k^*} W_m \bar{\phi}^T \bar{w}. \quad (\text{B.21})$$

By eliminating the differential operator  $W_m$ , we obtain

$$r = \frac{\Phi_m}{\Phi_\theta} r - \frac{1}{k} \bar{\phi}^T \bar{w} \quad (\text{B.22})$$

$$= \frac{\Phi_m}{\Phi_\theta} r - \frac{1}{k^*} \frac{\Phi_m}{\Phi_\theta} \bar{\phi}^T \bar{w}_m. \quad (\text{B.23})$$

Finally, from eqs.(B.16) and (B.23), we get the following expression:

$$w = \begin{bmatrix} r \\ \bar{w} \end{bmatrix} = \frac{\Phi_m}{\Phi_\theta} \begin{bmatrix} r \\ \bar{w}_m \end{bmatrix} + \frac{\Phi_m}{\Phi_\theta} \begin{bmatrix} 0 & -\bar{\phi}^T/k^* \\ 0 & \phi_k/k^* I \end{bmatrix} \begin{bmatrix} r \\ \bar{w}_m \end{bmatrix}. \quad (\text{B.24})$$

Therefore, defining

$$G_\theta \stackrel{\text{def}}{=} \begin{bmatrix} 0 & \bar{\phi}^T/k^* \\ 0 & \phi_k/k^* I \end{bmatrix} \in \mathbb{R}^{2n \times 2n}, \quad (\text{B.25})$$

we get

$$w = \frac{\Phi_m}{\Phi_\theta} [I + G_\theta] w_m. \quad (\text{B.26})$$

Since  $w = H_{wr} r$  and  $w_m = H_{w_m r} r$ ,  $H_{wr}$  can be rewritten as

$$H_{wr}(j\omega, \theta) = \frac{\Phi_m}{\Phi_\theta} [I + G_\theta] H_{w_m}(j\omega). \quad (\text{B.27})$$

Consequently, by substituting the above expression and eq.(3.17) into eq.(B.6), we get

$$S_{w_f w_\theta}(d\omega) = \frac{1}{k^*} \left| \frac{\Phi_m(j\omega)}{\Phi_\theta(j\omega)} \right|^2 |W_m(j\omega)|^2 \frac{\Phi_m(j\omega)}{\Phi_\theta(j\omega)} [I + G_\theta] H_{w_m r}^*(j\omega) H_{w_m r}^T(j\omega) [I + G_\theta^T] S_r(d\omega). \quad (\text{B.28})$$



## References

---

- [Acton, 1970] F. S. Acton, *Numerical Methods That Work*, Harper and Row, 1970
- [Allen, 1989] R. B. Allen, "Adaptive Training of Connectionist State Machines," *Proc. of ACM Computer Science Conference*, Louisville, February, 1989
- [Anderson, 1986] B. D. O. Anderson, et al., *Stability of Adaptive Systems: Passivity and Averaging Analysis*, MIT Press, Cambridge, MA 1986
- [Asada and Kakumoto, 1990] H. Asada and Y. Kakumoto, "The Dynamic Analysis and Design of a High-Speed Insertion Hand Using the Generalized Centroid and Virtual Mass," *ASME Journal of DSMC*, vol. 112, pp. 646-652, 1990
- [Åström, 1984] K. J. Åström, "Interactions between Excitation and Unmodeled Dynamics in Adaptive Control," *Proc. of 23rd IEEE CDC*, pp. 1276-1281, Las Vegas, 1984
- [Åström and Wittenmark, 1989] K. J. Åström and B. Wittenmark, *Adaptive Control*, Addison-Wesley, 1989
- [Bai and Sastry, 1987] E. W. Bai and S. S. Sastry, "Global Stability Proofs for Continuous-Time Indirect Adaptive Control Schemes," *IEEE Trans. of Automatic Control*, vol. 32, pp. 537-543, 1987
- [Barto, et al., 1983] A. G. Barto, R. S. Sutton and C. W. Anderson, "Neuronlike Elements That Can Solve Difficult Learning Problems," *IEEE Trans. of Systems, Man, and Cybernetics*, vol. 13(5), p.p. 835-846, 1983

- [Bogoliuboff and Mitropolskii, 1961] N. N. Bogoliuboff and Y. A. Mitropolskii, *Asymptotic Methods in the Theory of Nonlinear Oscillators*, Gordon and Breach, New York, 1961
- [Boyd and Sastry, 1986] S. Boyd and S. S. Sastry, "Necessary and Sufficient Conditions for Parameter Convergence in Adaptive Control," *Automatica*, vol. 22, pp. 629-639, 1986
- [Etter and Masukawa, 1981] D. M. Etter and M. M. Masukawa, "A Comparison of Algorithms for Adaptive Estimation of the Time Delay Between Sample Signals," *Proc. of ICASSP-81*, pp. 1253, March, 1981
- [Gullapalli, 1990] V. Gullapalli, "A Stochastic Reinforcement Learning Algorithm for Learning Real-Valued Functions", *Neural Networks*, Vol. 3, pp. 671-692, 1990
- [Gullapalli and Barto, 1990] V. Gullapalli and A. G. Barto, "Shaping as a Method for Accelerating Reinforcement Learning", *Proc. of the 1992 IEEE International Symposium on Intelligent Control*, pp 554- 559, Glasgow, Scotland, 1992
- [Hale, 1980] J. K. Hale, *Ordinary Differential Equations*, Krieger, Huntington, New York, 1980
- [Hogan, 1985] N. Hogan, "Impedance Control: An Approach to Manipulation: Part I-III," *ASME Journal of DSMC*, vol. 107-1, 1985
- [Honig and Staddon, 1977] W. K. Honig and J. E. R. Staddon, *Handbook of Operant Behavior*, Prentice Hall, Englewood Cliffs, NJ, 1977
- [Horowitz, et al., 1990] R. Horowitz, et al., "Convergence Properties of Learning Controllers for Robot Manipulators," *Proc. of the Japan-USA Symp. on Flexible Automation*, Kyoto, Japan, July, 1990

- [Jordan and Rumelhart, 1992] M. I. Jordan and D. E. Rumelhart, "Forward Models: Supervised Learning with a Distal Teacher", *Cognitive Science*, 16, 1992
- [Kokotović, et al., 1985] P. Kokotovic, et al., "On a Stability Criterion for Continuous Slow Adaptation," *Systems & Control Letters*, vol. 6, pp. 7-14, June, 1985
- [Kokotović, et al., 1992] P. Kokotović, et al., "Backstepping to Passivity: Recursive Design of Adaptive Systems," Proc. of the 31st IEEE Conf. on Decision and Control, Arizona, December, 1992
- [Narendra, et al., 1980] K. S. Narendra, et al., "Stable Adaptive Controller Design, Part II: Proof of Stability," *IEEE Trans. on Automatic Control*, vol. 25, pp. 440-448, 1980
- [Narendra and Annaswamy, 1987] K. S. Narendra and A. M. Annaswamy, "Persistent Excitation of Adaptive Systems," *Int. Journal of Control*, vol. 45, pp. 127-160, 1987
- [Narendra and Annaswamy, 1989] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*, Prince Hall, 1989
- [Osburn, et al., 1961] P. V. Osburn, H. P. Whitaker and A. Kezer, "New Developments in the Design of Adaptive Control Systems," Paper No 61-39, Inst. Aeronautical Sciences, Feb. 1961
- [Parks, 1966] P. C. Parks, "Lyapunov Redesign of Model Reference Adaptive Control Systems," *IEEE Trans. of Automatic Control*, vol. 11, pp. 362-365, 1966
- [Poggio and Girosi, 1989] T. Poggio and F. Girosi, "A Theory of Networks for Approximation and Learning," *A.I. Memo* No. 1140, Artificial Intelligence Laboratory, MIT, 1989

- [Riedle and Kokotovic, 1985] B. D. Riedle and P. V. Kokotovic, "A Stability-Instability Boundary for Disturbance-Free Slow Adaptation with Unmodeled Dynamics," *IEEE Trans. on Automatic Control*, vol. 30, no. 10, pp. 1027-1030, 1985
- [Rohrs, et al., 1982] C. E. Rohrs, et al., "Robustness of Adaptive Algorithms in the Presence of Unmodeled Dynamics," Proc. of 21st IEEE CDC, Florida, 1982
- [Sastry and Bodson, 1989] S. Sastry and M. Bodson, Adaptive Control: Stability, Convergence and Robustness, Prentice Hall, 1989
- [Sethna, 1973] P. R. Sethna, "Method of Averaging for Systems Bounded for Positive Time," *Journal of Math. Anal. and Applications*, vol. 41, pp. 621-631, 1973
- [Werbos, 1988] P. J. Werbos, "Generalization of Back Propagation with Application to a Recurrent Gas Market Model", *Neural Networks*, 1, pp. 339-356, 1988
- [Widrow and McCool, 1976] B. Widrow and J. M. McCool, "A Comparison of Adaptive Algorithms Based on the Methods of Steepest Descent and Random Search," *IEEE Trans. of Antennas Propag.*, vol. AP-24, pp. 615, Sep. 1976
- [Widrow and Stearns, 1985] B. Widrow and S. Stearns, Adaptive Signal Processing, Prentice-Hall, 1985
- [Widrow, 1986] B. Widrow, "Adaptive Inverse Control," *Proc. of Second IFAC Workshop on Adaptive Systems in Control and Signal Processing*, pp. 1-5, Lund, Sweden, July, 1986
- [Williams, 1992] R. J. Williams, "Simple Statistical Gradient-Following Algorithm for Connectionist Reinforcement Learning", *Machine Learning*, 8, 1992

[Whitney, 1977] D. E. Whitney, "Force Feedback Control of Manipulator Fine Motions," ASME Journal of DSMC, vol. 99, no. 2, pp. 91-97, 1977

[Yang and Asada, 1995] B.-H. Yang and H. Asada, "Adaptive Reinforcement Learning and Its Application to Robot Compliance Learning", *Journal of Robotics and Mechatronics*, vol. 7, No. 3, pp. 250-262, 1995 Mechatronics, and Haptic Interfaces, ASME Winter Annual Meeting, 1993

[Yang and Asada, 1993-(b)] B.-H. Yang and H. Asada, "Reinforcement Learning of Assembly Robots," *Proc. of the Third Int. Symp. on Experimental Robotics*, Kyoto, Japan, Oct. 1993