# Progressive Learning of Endpoint Feedback Systems with Model Uncertainty and Sensor Noise

by

**Shih-Hung Li**

B.S.M.E., University of Texas at Austin, 1990

M.S.M.E., Massachusetts Institute of Technology, 1992

**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, 1996

Signature of Author ______________________________
Department of Mechanical Engineering
March 29, 1996

Certified by ______________________________
Haruhiko Asada
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by ______________________________
Ain A. Sonin
Chairman, Department Committee on Graduate Students

# Progressive Learning of Endpoint Feedback Systems with Model Uncertainty and Sensor Noise

by

**Shih-Hung Li**

Submitted to the Department of Mechanical Engineering on February 29, 1996
In partial fulfillment of the requirements for the degree of Doctor of Philosophy

## Abstract

In the control of high performance robots and machine tools, feedback from the endpoint sensor measuring the position of the end effector has great promise for improving accuracy. This endpoint feedback, however, often incurs instability due to the non-collocated sensor-actuator configuration coupled with unmodeled dynamics and sensor noise. Plant dynamics is often uncertain, and the tuning of endpoint feedback controllers is beyond the capacity of the users. The objective of this thesis is to develop an automatic tuning algorithm for non-collocated endpoint feedback systems.

In our attempt to solve problems associated with adaptive control, we proposed a *progressive learning* algorithm. The idea of *progressive learning* is to tune the system step by step in the frequency domain in order to expand the system bandwidth. In this work, the stability issue has been discussed in the framework of the model reference adaptive control (MRAC). Our main focus of the research is to guarantee the stability and the robustness of the system subject to the presence of the unmodeled dynamics and output noise. The system is tuned gradually and progressively by increasing either the trajectory frequency or the controller's order. Thus, fewer parameters need to be learned in each stage. This progressive cascading of controller stage by stage in order to achieve wider control bandwidth is referred to as *model augmentation.* In this research, we address the model augmentation as to when to augment the model and how to maintain stability, despite unmodeled dynamics and sensor noise. Next, a series of reference trajectories are designed in such a way that the system can be excited progressively starting from a low frequency range moving up to a full spectrum. To validate the theoretical results, a simulation is shown first and followed by experimental results and discussions of three endpoint controlled systems: a high-speed chip-placement machine, a linear slider, and a coordinate measuring machine (CMM).

| | |
|---|---|
| Thesis Supervisor: | Haruhiko Asada |
| Title: | Professor of Mechanical Engineering |

Star light, star bright; first star I see tonight.

I wish I may, I wish I might

Of the wish I wish

Comes true

Tonight.

# Dedication

To my parents
and
All the people
Who have helped me for the past twelve years

# Acknowledgment

My trip to this podium has taken me more than 20 years of my life and more than 10,000 miles away from home to complete it. Most importantly, much luck came from many people along the way. Therefore, besides acknowledging only the people who have helped while I was at MIT, the thank is not complete until I have thanked all the other people who have made me what I am today.

Starting at the age of 5 or 6 years old, my parents have started constantly reminding me how much better my life would be if I do have a Ph.D. diploma from MIT. At the age of 14, I was sent away to United States to start chasing my version of the American dream.

My first section of my journey started in Virginia. During my high school era, I would like to thank Teresa Tsia who was brave enough to take the responsibility of me without any direct relationship with the family. For that four years, she had given me a home away from home. I would like to thank Michael Shen, Michael and Alice Hsia for those four years that we spent together and developed and shared a stronger bond that the most of the siblings will never be able to achieve. Without your loving support and encouragement words, I do not believe that I would have ever gone this far.

My next section of my journey started in Austin, Texas where I went for my college education. I would like to thank my grandma, my uncle, Ben Cheng, and my aunt, Mindy Chang, and their families for giving me another home away from home. I would also like to thank Dr. Ilene Busch-Vishniac who had given me the opportunity that eventually landed me the chance to attend MIT. Most important of all, thank you for your kind, encouraging advice and guidance through all these years.

My final section of journey, of course, started in Cambridge, MA. I still remembered what was on my mind when I first saw the main building when I stood on the Killian Court. Finally the dream is within my reach, at least that was what I thought at the time. I would like to thank my aunt, Wendy Kung and her family for giving me my third home away from home. It is your moral support that has helped me through my darkest moment of my journey when my dream for a Ph.D. was almost shattered. I would like to thank Professor Asada who guided me through my final journey to the podium for the past six years. It is from you I learned the meaning of patience, persistence, perseverance, and honesty. It is your hands that have molded me who I am today both in academia and outside the academia arena. I would like to thank Professor Annaswamy and Professor Youcef Tourmi, my other two thesis advisors. Thank you so much for your time and effort that you have spent on me. Without your guidance, I probably still can not see the light at the end of the tunnel. I would also like to thank Dr. Booho Yang who though was not officially on my committee but had spent equally amount of time and effort to guide me. Last, I would like to thank all the past and present members in the CIDMS for keeping my sanity intact: Dr. Johng Park, Dr. Sheng Liu, Dr. Kevin Brown, Dr. Mark West, Dr. Xingdong He, Dr. Anton Pil, Dr. Sooyong Lee; many doctor-to-be: Ming Zhou, Joe Spano, Muralidhar Ravuri, Sokwoo Rhee, Steve Mascaro, Rolland Doubleday, and Susan Ipri; friends from the other side of world: Kenji Okamoto, Nori Yushida, Nori Fujiwara, Toizumi Atsushi, Toshi Mimura; and finally, of course, Kari Kulaszewlcz, our lovely secretary. I will always remember all the good times we had. I will also cherish all these memories for many years to come. Besides all the people in the lab, I also like to thank David and I-Ching for being the people that I can lean on for these past six years; Shih-Ken for walking together through that darkest moment and lending a helping hand in ROCSA. I would also like to thank anyone who associated with ROCSA and had helped it along the way especially those of you that were under my terrible leadership that year. I would like to name you all, but there are just too many of you. Instead of having the environmental people on my back for accusing me for killing too many trees to printing my thesis, I would just like to say "thank you."

Finally, I would like to thank God for protecting me and giving me the luck when I needed the most for all these years. I would like to thank my family who has supported me all these years. Mom and Dad, thank you for giving me this dream to complete and lighting up my path along the way. Sis, thank you for taking care the family for all these years so I can concentrate on my school work. Thank you for being there when I needed the most. Finally, I would also like to thank Michelle Jean. Thank you for the past two and half years for giving me the immediately supports when I needed it the most. Thank you so much for bearing with me when I was down, sharing my happiness and my sadness, and being the person I can always talk to. Soon, you will be my partner to walk together for the journey ahead of us. Thank you and may God bless you.

Looking back, there are just too many friends to thank and too many memories to hold on to. On June 7, I will officially leave this place, and my journey to the podium will finally be completed. I will always miss you all and my heart will always be with you. I would like all of you to know that this is not the end of it but just a new beginning. I would like to say "thank you very much" one more time, and "good luck" to all of you before I head off to my next journey in my life.

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background

In the manufacturing area, we judge the performance of a machine based on its productivity, its ability to adapt quickly for various task complexities, its reliability, and its repeatability under various operating conditions. However, whenever a new task is assigned or a changeover occurs, most of the machines need to be shut down for adjustments. This down-time that the machine is unproductive is unacceptable for high-volume production machines. Also, during the production stage, operating conditions change as the environmental conditions vary, thus in order to maintain the performance, its ability for on-line adjustment is an absolute necessity.

Speed and accuracy are two fundamental performance specifications of many kinds of machines. This is specially true for precision machines such as chip-placement machine, robots, and numerical controlled (NC) coordinate measuring machine (CMM), where demands for higher speed and tighter tolerance are increasing. Since a machine performs a task by moving its end-effector, i.e., the cutter or a chip-gripper, the positioning accuracy of its endpoint rather than the accuracy at the motor axes is critically important. Most of today's machine are controlled based on the sensing information obtained from the actuator axes. Therefore, anything beyond the actuator axes is out of the feedback loop. Backlash at the gearing and transmission mechanisms, misalignment of the structure, deformation due to the payload, etc. are all outside of the position feedback loop. To compensate for these sources of endpoint error, we need to measure the endpoint position to perform closed-loop feedback. This type of feedback control that uses the endpoint sensor data for feedback is referred to as the *endpoint feedback control.* The advantage of using the endpoint feedback is that it will give better performance results than the conventional collocated sensor/actuator control. The endpoint feedback, however, needs a special care in tuning the feedback loop because the endpoint feedback tends to have a large phase lag due to the non-collocated sensor/actuator configuration of sensors and actuators. As a result, the control system becomes difficult to stabilize.

At the accuracy required and speed operated by these precision machines, the systems need to be constantly monitored and controlled. It is known that any physical system has an infinitely system dynamics order; however, as a wide practice, by defining operating condition, we then use a finite-complexity model to describe the system's behavior. It is based on this "reduced" model that we perform the control. Thus, this "model" could be as general as the task, or as finite as the actual system. In most cases, either learning or adaptive control needs to be applied. The learning control "controls" at the task-level, thus retraining is a must when a different task is required to be performed. The adaptive control, on the other hand, controls the system at the system level directly or indirectly based on an assumed system model. Thus, different tasks that operate at the same or relative condition can be tolerated without retraining. As a part of our requirements is to minimize the time for retraining, we need to implement the adaptive control scheme. Nonetheless, due to the assumed model formulation, there will always be the presence of unmodeled dynamics. In terms of adaptive control, as the system order and complexity increase, the number of the parameters needs to be tuned increases at a much faster pace. It becomes very hard to tune many parameters while still maintaining system and parameter stability with the presence of unmodeled dynamics. The direct results of this drawback for the adaptive control scheme lead to seldom practice on actual complex systems. Another drawback for the adaptive scheme is the time required and the method for tuning the adaptive parameters. During the tuning, the machine has to be shut down from its actual operation which is detrimental for any high-volume production machines. Since a large number of adaptive parameters needs to be tuned, stability has become a big issue.

In most of the actual systems, sensor noise could also be detrimental to the adaptive control algorithm since it is based on this corrupted output sensing data that the adaptive control is operating on. Therefore, it is quite important to develop a strategy that allows the system to be excited gradually, yet in a stable manner and is robust enough to counter-measure the noise as well as the unmodeled dynamics.

## 1.2 Objective and Approach

The objective of this thesis is to formulate a stable, automatic and cohesive adaptive/tuning endpoint feedback control algorithm to improve the system performance at vast operating conditions. Our approach of this thesis is to explore a learning method in which the level of the task complexity advances progressively in accordance with the learner's competence and level of accomplishment for improving the performance of endpoint feedback systems.

In the learning method, termed, *progressive learning* which is first presented in [Yang, 1995], stated that it is possible to learn/adaptive input/output of a complex system with high relative degree in its transfer function between its input and output by designing a series of tasks with different complexity levels that reflects the current knowledge of a leaner. In other words, *progressive learning* is an algorithm that helps the controller to learn the system step by step in the frequency domain in order to expand the system bandwidth. In this original work, the stability issue has been discussed in the framework of the model reference adaptive control (MRAC).

The original work of the *progressive learning*, like conventional adaptive control, does not relate a true task to the system level input in training, thus random or non-task-related but stable signals are subjected to the system for tuning. Thus, the system has to be shut down for the tuning. Since we have to minimize this unnecessary "shut-down", we need to solidify the relation between the stable training signals and the actual task. Another drawback of the original work of *progressive learning* is that the dynamics system is considered to be ideal, i.e., the complete order of the system is known as well as a system that is free of noise. However, for any physical system, the complete system order may not be known ahead of time or due to the hardware issue, full-order of the system complexity may not be considered all at once. Besides, there will be always some sensor noise present in the system. Therefore, it is our intention to modify and improve the current state of *progressive learning*. In this research, the concept of *progressive learning* has been extended so that it can be applied to most of the endpoint feedback controlled manufacturing machinery. Due to the fact that any physical system has infinite order, we can never obtain a full-representation of the model. Instead, we will obtain a reduced form model. Thus, our main focus of the research is to guarantee the stability and robustness of the system during the tuning with the presence of the unmodeled dynamics as well as the output noise. In most cases, the information at higher frequency range is not so reliable due to the noise. At the same time, the higher the order of the system we use, the more parameters need to be tuned at each time. Thus, another incentive for us to use the reduced-formed model is so that a less number of parameters needs to be tuned each time.

Another emphasis of the research is to link the required task to the stable reference design so we can tune the system while it is performing an actual task. Then the system is tuned gradually and progressively by increasing either the reference excitation frequency bandwidth or the system's dynamics model's complexity depending on the progress of the tuning.

The nature of any physical system when it is excited at low frequencies is that the system behaves as an ordinary rigid-body system; therefore, there is no reason to tune and control the system using a complex controller for it. To take advantage of the reduced dynamics behavior of the physical system, the system is first partitioned into a series of stages arranging by their natural frequencies. Then, the system is "progressively" adaptively learned and controlled stage by stage instead of all at once. As we increase the trajectory speed, we start to excite higher modes of the system, thus more complex controller needs to be designed and controlled. The way of progressively cascading the controller to control stage by stage is referred to as *model augmentation*. Thus, less amount of the parameters need to be tuned at each time. With these intermediate stage training results, a "lean" controller for vast operating conditions, or a most compact controller without degradation of the system performance at each operating regions can be easily obtained and implemented. In this research, based on our problem formulations of stability and robustness analysis, we address the model augmentation in terms of when to augment the model and how the parameters should be expanded. The concept of using a trajectory with a known designed frequency spectrum in the beginning, and increasing the designed frequency bandwidth as tuning proceeds to generate a stable reference input is referred to as the *trajectory synthesis*. As more knowledge of the system is gained, the machines' throughput is guaranteed to increase since wider control bandwidth can be expected. More importantly, since trajectory design has incorporated the system dynamics, better accuracy can be expected than the traditional static or quasi-static assumptions usually assumed for the trajectory design. Together, they are cohesively linked to the frame of robustness and stability of the adaptation dynamics in the final form of *Extended Progressive Learning*. The proposed tuning algorithm combines the learning, the adaptation, and the augmentation cohesively and automatically.

To validate the research, a simulation is shown first to demonstrate the concept. Then it is followed by a series of actual experimental results and discussions of three endpoint feedback controlled systems: a high-speed chip-placement machine, a linear slider, and a coordinate measuring machine(CMM).

## 1.3 Outline of the Thesis

This thesis is composed of three main parts: the theory, the algorithm to apply the theoretical results, and experimental results by applying the *progressive learning* to the three actual systems mentioned earlier. In the theory portion of the thesis, a problem formulation based on the reduced model in the frame of MRAC is first described. The stability analysis first conducted by assuming that the effect of the unmodeled dynamics and output noise are negligible in Sec. 3.4. In Sec. 3.5, we perform the robustness analysis to see the effect of unmodeled dynamics and sensor noise on the stability of the adaptation dynamics. Finally, based on the robustness analysis, we describe how to design stable reference input through the use of *trajectory synthesis* is discussed in Sec. 3.6.

After describing the theoretical developments, we will formally describe the *extended progressive learning* algorithm that bridges the differences between the actual applications and the theoretical results in Chapter 4. In this chapter, we discuss how we can transform an actual system application and apply the *extended progressive learning* on it. Several of the practical issues such as determining the model structure, sensor noise, and on-line validation of the results are all discussed in this chapter. Also, based on the stability and the robust analysis, when and how to perform model augmentation are discussed in Sec. 4.3. Then it is based on this algorithm we perform the simulation to validate the proposed research in Sec. 4.5. In Chapter. 5, we presented three actual experimental tuning results: a high-speed chip placement machine, a linear slider, and a CMM. Finally, the conclusions and future works are given in Chapter 6.

Chapter 2

# Extended Progressive Learning for Endpoint Feedback Controlled Systems

## 2.1 Introduction

Endpoint feedback systems have been one of the systems to which adaptive and learning controls are difficult to apply. In most of the cases, the relative degree, which is defined as the order difference between the denominator and numerator of a transfer function, of the input/output transfer function is high or greater than two. One of most fundamental requirements for an adaptive controller to work is to "persistently" excite the system. However, because of the parameter uncertainty and large phase lag, the system might become unstable when it is persistently excited. Naturally, the key question is how to design the input signals that will persistently excite the system without making it unstable. As described earlier, the productivity is directly proportional to the speed which the machine can perform a task. In our approach, we view the task as a particular form of trajectory. By carefully designing the input signals to satisfy the persistent excitation requirement and the trajectory requirement, the system parameter convergence is guaranteed. More importantly, the system performs a true task so the minimum productivity is maintained throughout the process. Then, the controller will make different decisions based on the system performance. If the controller has detected any significant degradation of the results, the controller will increase the assumed controller model order to continue train the system. Otherwise, the controller will just increase the task's speed to increase the productivity while tuning the system at higher bandwidth. The decision making potion of the controller which is the main focus point of thesis will be presented later. The main objective of this chapter is first to review works in the related topics. The second objective of this chapter is to give a brief overview or the concept of the proposed research. Next, we are going to discuss each of them separately.

## 2.2 Related Works

Some related works with the same or partially the same interest include stable adaptive control, persistent excitation, progressive control, process control and supervisory control. We will discuss them individually to see how they fit into and differ from our research.

The stable adaptive control has been well formulated by [Narendra and Annaswamy, 1989]. In stable adaptive control, for example, the design of reference signals has been a central issue to the convergence of system parameters. In particular, the concept of persistent excitation (PE) has played an important role for the design of reference inputs. The PE provides conditions for the parameters of a plant or a controller to converge to their desired values [Narendra and Annaswamy, 1989]. By making the assumptions that the time constant of the convergence of the system parameter is much smaller than the system dynamics, [ Åström and Wittenmark, 1989] has proposed to use the averaging theory to examine the stability and convergent rate based on the PE condition. As pointed out by [Kosut, et. al, 1985], the averaging theory has its uses and limitations for adaptive systems. In the first place, the theory requires slow adaptation. Secondly, averaging theory is a form of linearization while the actual adaptive system is a non-linear system. Thus, the nonlinearity must be treated by initializing the system in a neighborhood of the tuned system. The "tuned system" is defined as the plant with its closed-loop characteristics equivalent to the desired model proposed after tuning the controller parameters. Nevertheless, the averaging technique does explain the system's behavior near the tuned system. [Riedle and Kokotovic, 1985] later proposed to use the integral manifold approach by applying the singular perturbation technique when the assumption of the two time constants can not be met as for the averaging technique. Then, [Yang and Asada, 1993] first proposed the use of *progressive learning* to address the design of the reference input in conjunction with the system performance. By using the technique of averaging theory and the gradient descent rule for the adaptation, [Yang, 1995] is able to guarantee stable parameter convergence for systems that have relative order greater than three with the use of the *progressive learning*.

All of the above research works address the convergence without addressing the unmodeled dynamics which are always present in any actual system. [Rohrs, et. al, 1985] first examined the robustness properties of existing adaptive control algorithms to unmodeled plant dynamics and unmeasurable output disturbance in the high frequency range. Due to the unmodeled dynamics, PE condition may not be guaranteed. Nevertheless, as pointed out in [Boyd and Sastry, 1986], the estimated system parameters will still achieve partial parameter convergence with respect to the excitation frequencies.

In the case of the supervisory control, similar research works appear in the area of gain scheduling, [Rugh, 1990], [ Åström and Wittenmark, 1989], and [Tran and Hrovat, 1993] to name a few. The gain scheduling is a counter part of adaptive control. The gain scheduler uses a "look-up" table that stored different predetermined controller gains to control the system open-loop at various operating points. Nevertheless, it provides some insight in relating the performance to the learner. However, so-called "look-up" table needs to be "pregenerated" and "pre-determined" for the gain scheduler.

All of the above research works have concentrated in control. As part of our task, we need to also study the trajectory synthesis for generating stable reference inputs based on the prescribed task. Most of the research works that study the trajectory synthesis problem are based on an assumed kinematic model or a dynamic model. There are less attentions paid to the controller performance and model uncertainty.

In the field of the process control, stochastic process control (SPC) is probably the best known and widely practiced control scheme [Owen, 1989], [Phadke, 1989], [Pyzdek and Berger, 1992]. In SPC, the data are collected, analyzed, and compared with the previous data through a series of eight prescribed rules at each of the predetermined sampling time period. Based on the operator's interpretation of the current analyzed data, a different action is taken. Thus, the SPC is an off-line estimation process and no prediction is made at any time. The advantage of the SPC is that it gives us some quick understanding of the system's behavior off-line. The disadvantage is that the sampling frequency is predetermined and requires human intervention.

In conclusion, there is not a fully automatic adaptive/tuning system process control algorithm available to address all of the above issues. Next we are going to describe a brief overview of our proposed research.

## 2.3 Concept of the Extended Progressive Learning

In this thesis, I propose to extend works done on *progressive learning* originally developed by [Yang and Asada 1993]. In this thesis, we integrate the learning rule together with the learning scheduler. The learning scheduler can vary the task complexity level or the learner's structure depending on the learning progress. The final learning method, which we refer to as *Extended Progressive Learning*, is defined as follows:

> Extended Progressive Learning is a learning method in which the level of task complexity and the level of learner's internal structure complexity are gradually increased in accordance with the progress of learning. This is done so that the 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.

The progressive learning is a dynamic process since task assignments vary dynamically during the learning process. As shown in Figure 2.1, the system consists of a learner, a task process or a plant, a performance evaluator, and a supervisor. The desired trajectory is pre-designed to meet a set of target points. The supervisor determines appropriate strategy for the system to take by either increasing desired trajectory speed with the same complexity level for the learner or increase the complexity level for the learner to achieve better system overall performance.

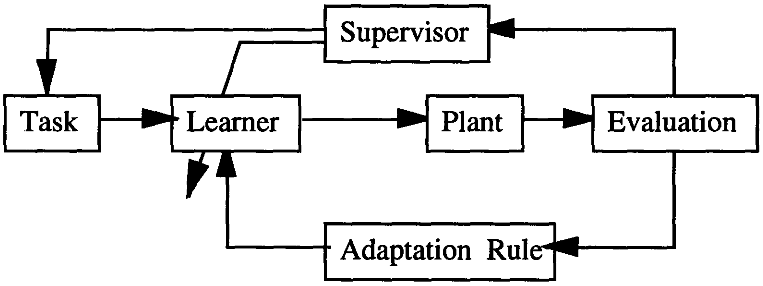


Figure 2.1: Progressive learning control block diagram

As stated earlier, there are four major parts of the development of a progressive learning controller. The stability analysis is first discussed based on the formulation. The robustness analysis is then discussed to extend the stability analysis to cover the effect of unmodeled dynamics and output sensor noise. Then based these two theoretical results, an optimal augmentation scheme will be discuss as when and how to augment the controller's structure. Finally the trajectory synthesis is discussed. Figure 2.1 summarizes them in a block diagram form. In the following chapter, we will discuss each part in more detail.

Chapter 3

# The Theory of Model Augmentation and Trajectory Synthesis through Progressive Learning

## 3.1 Introduction

In this chapter, we are going to discuss the theory of *extended progressive learning* in particular the *model augmentation* and the *trajectory synthesis*. First, we state the problem statement of the proposed research then we will formulate the problem. Then based on the problem formulation, we will first focus on the stability issue relating to the model augmentation in Sec. 3.4 under the ideal situation where the contribution from the unmodeled dynamics does not have any effect on the system's performance and the corrupted sensor output by noise is minimal. Sec. 3.5 will focus on the robustness issue of the augmentation when the effect of unmodeled dynamics and sensor noise do affect the tuning performance. Finally I will focus on the issues and approaches we made for the stable reference input design in the context of trajectory synthesis in Sec. 3.6.

## 3.2 The Problem Statement

Endpoint feedback systems are one class of non-collocated systems with a high relative degree and have been one of the systems to which adaptive and learning controls are difficult to apply. One of the most fundamental requirements for an adaptive controller to work is to "persistently" excite the system. However, because of the parameter uncertainty and large phase lag between the input and output signals, the system might become unstable when it is persistently excited. Naturally, the key question is how to design the input signals that will persistently excite the system without making it unstable. As described earlier, the productivity is directly proportional to the tracking speed. By carefully designing the input signals to satisfy both the persistent excitation requirement and trajectory requirement, the system can be persistently excited in a stable fashion for the system parameter adaptation to converge and more importantly, the system performs a true task so the minimum productivity is kept throughout the process. Based on the system performance, the controller needs to make on-line decision in determining whether to stay at the current operating speed by modifying the assumed control structure to take on the "non-collocated" behavior, or just to increase the operating speed with the same control structure. In either case, it will need to guarantee equal if not better tracking performance.

The trajectory is specified by a finite set of points in space as shown in Figure 3.1. It is required to go through all these discrete points, and come back to the original point, but the intermediate points that connect each prespecified points are not determined. Therefore, there are certain degrees of freedom in generating an actual path. Also the total cycle time required to complete one cycle of trajectory tracking is unspecified. As the learning proceeds, the cycle time should be shortened as much as possible. During the learning phase, however, the system is allowed to track the trajectory at slower speeds, as long as all the specified points are visited correctly.

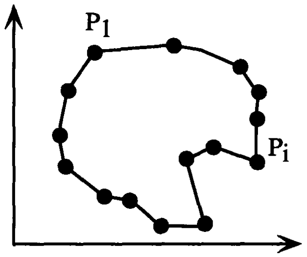


Figure 3.1: Specified points and spatial trajectory

Figure 3.2 shows time profiles of the same trajectory that go through all the specified points. Note that these trajectories have the same spatial frequency spectrum since they all have to go through the same sets of points in space. However, they all have different temporal spectrum since their operating speeds or the time profiles are different. In other words, they all go through the same point in space at a different time. The temporal frequency spectrum expands or shrinks depending on the cycle time $T$ and tracking speed. According to the progressive learning theory, we should start with low excitation frequencies, and gradually increase the frequencies as learning proceeds. Therefore, as shown in the figure, the system first learns the slow trajectory with the longest cycle time $T_1$, and after completing the learning of the slow trajectory it learns the medium speed trajec-

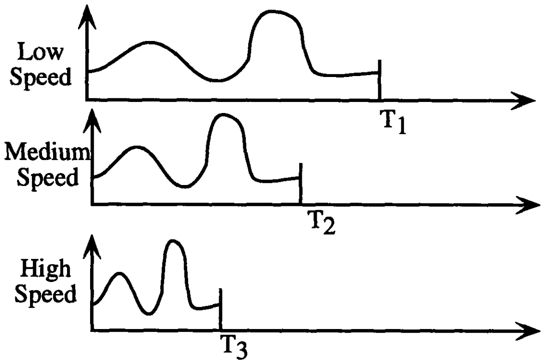


Figure 3.2: Time profiles for different cycle times

tory with a shorter cycle time, $T_2$. The learning process would be repeated for shorter cycle times as far as the control system can track all the specified points within a tolerance error. When the tracking speed exceeds the physical limit of the control system, the tracking error may neither be kept within the tolerance nor be improved by learning. The learning process is terminated when this phenomenon is observed. Thus, the system is guaranteed to perform the trajectory tracking with a specified accuracy throughout the learning process. This feature of progressive learning meets industrial needs and fits to actual manufacturing environment, where minimum task specifications must be satisfied even in the early stage of learning and the maximum productivity must be achieved as production demands increase.

It is a known fact that any physical system, particularly the endpoint feedback system, has an infinite system order between the input and output of the system. Based on the desired operating bandwidth, we will simplify this "infinite" order system to a "lumped" parameter model as shown in Figure 3.3. In this figure, the actuation will always be applied to $M_1$ while the load is applied to the furthest point. The output sensing is then obtained from the point of load directly. The system may have a phase lag between the actuator and the load due to the compliance at the transmission, but it is negligible at low frequencies. Therefore, the whole system can be modeled as a single rigid body, as shown in Figure 3.3A. Both the actuator encoder and the endpoint sensor provide basically the same positional information, as long as the system tracks a trajectory at a low speed and the frequency spectrum of the trajectory is within a low frequency range. Tuning of control parameters is rather straightforward for this collocated system.

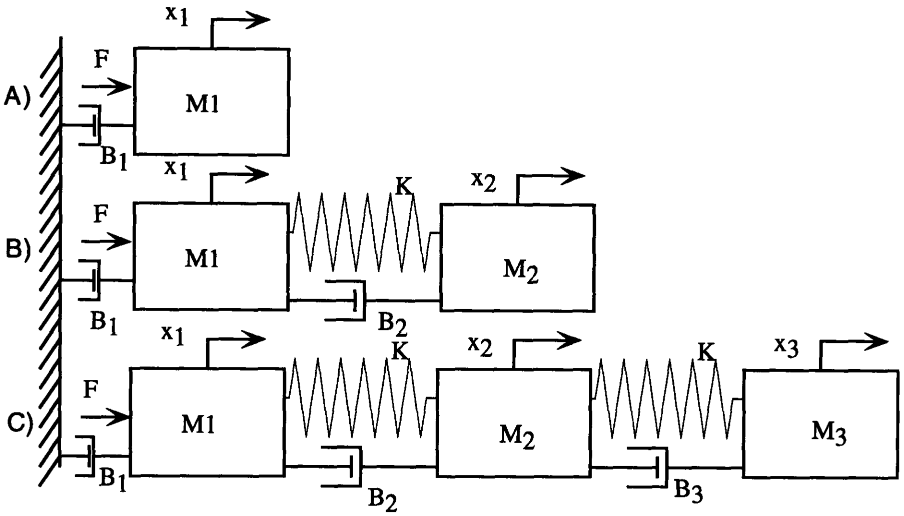


Figure 3.3: Lumped parameter models with different system orders

As the frequency spectrum expands, the simplified system model must be updated to the one involving the next higher dynamic mode. As shown in Figure 3.3B, the single rigid body is now split into motor inertia, $m_1$, and a load inertia $m_2$. Both masses are connected by the equivalent stiffness $k$ and damping $b$. Since our goal is to close the control loop from the endpoint sensor, the control system becomes non-collocated as the second mode becomes prominent. Tuning of such a non-collocated system is much harder than that of a collocated system. Applying the progressive learning method; however, we can resolve the difficulty of such a non-collocated system. After tuning at a slow speed, the stable range of excitation frequency has been expanded. As a result, the learning at a faster speed may be stable. The stability is guaranteed even for a non-collocated system where the relative order is three or higher.

As the cycle time further reduces and the excitation frequencies become too high, the vibration modes of the structure will be excited. As shown in Figure 3.3C, the system must be modeled as a three mass system. The system order as well as the relative order increase and, as a result, the order of the controller and observer must be increased. Unless the controller and observer have the right orders with sufficient parameters to be tuned, the learning can not be performed correctly. In other words, in progressive learning the control system cannot learn when trajectories having a frequency spectrum that excites the unmodeled dynamics.

In Sec. 3.4 we will first discuss the stability issue and in Sec. 3.5 we will discuss the robustness of the adaptive system in the presence of the unmodeled dynamics as well as the output disturbance noise. Then we will address how to design the reference input, i.e., the synthesis of trajectories. The trajectory must satisfy not only geometric requirements described by a set of points but also the stability and convergence conditions. Each frequency component involved in the trajectory may influence the stability of the learning process as well as its convergence speed. In Sec. 3.6, we will analyze these conditions and formulate a trajectory synthesis problem.

Next, we are going to first formulate the problem before we discuss each part.

## 3.3 Problem Formulation

In this section, we consider a model reference adaptive control(MRAC) scheme of the type treated in standard textbooks (e.g., [Narendra and Annaswamy, 1987]). The system that we are considered here is a single-input-single-output system. The overall 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

$$y_p = W_p(s)u \tag{3.1}$$

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

$$y_m = W_m(s)r \tag{3.2}$$

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

$$e_1 = y_p - y_m \tag{3.3}$$

converges to zero asymptotically for arbitrary initial conditions and arbitrary piece-wise 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 $W_m(s)$ with $s$ denoting either the Laplace variable or the differential operator:

(A1) $R_p(s)$ is a monic polynomial of known degree $n$,

(A2) $Z_p(s)$ is a monic Huriwitz polynomial of known degree $m$,

(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.

and further assume

(A6) $W_p$ can be distinctly cascaded by series of first order or second order system as shown below

$$W_p = \prod_{i=1}^{n} W_{p_i} = \prod_{i=1}^{n} \frac{K_{Wp_i} Z_{Wp_i}}{R_{Wp_i}} = \frac{k_p Z_p}{R_p} \tag{3.4}$$

where index $i$ goes from 1 to $n$ as the bandwidth varies from narrow to wide.

(A7) $W_m$ can also be cascaded by series of first order or second order system as shown below

$$W_m = \prod_{i=1}^{n} W_{m_i} = \prod_{i=1}^{n} \frac{K_{Wm_i} Z_{Wm_i}}{R_{Wm_i}} \tag{3.5}$$

where index $i$ goes from 1 to $n$ as the bandwidth varies from narrow to wide.

Our major research goal is to progressively "augment" the system dynamics as we gain more information about the system in order to increase the control bandwidth. We will make extra assumptions about the plant and model dynamics at each "stage" to assume they can be approximated in the following forms:

(A8) $W_p$ can be cascaded into $G_{p_q}$ and $G_{r_q}$

$$G_{p_q} = \prod_{i=1}^{q} W_{p_i} = \frac{k_{Gp_q} Z_{Gp_q}}{R_{Gp_q}} \qquad \text{with order of } n_q \tag{3.6}$$

$$G_{r_q} = \prod_{i=q+1}^{n} W_{p_i} = \frac{k_{Gr_q} Z_{Gr_q}}{R_{Gr_q}} \qquad \text{with order of } n_n - n_q \tag{3.7}$$

(A9) Based on $G_{p_q}$, $G_{m_q}$ is chosen as

$$G_{m_q} = \prod_{i=1}^{k} W_{m_q} = \frac{k_{m_q} Z_{m_q}}{R_{m_q}} \qquad \text{with order of } n_q \tag{3.8}$$

where the subscribe $m_q$ and $p_q$ denote the approximated desired model and plant model respectively at each stage $q$; $r_q$ denotes the residual plant model respectively at each stage $q$; and $n_q$ denotes the order of the system at the stage $q$ where the overall order of the system order is $n$. By using the notation denoted sequentially, $n$ is represented as $n_n$ at $n$th stage. To simplify the description, from this point on, the transfer function that is in question is defined as the transfer function between the input, or the actuation, and the output, the output sensor. $W$ represents the overall system transfer function. $G$ represents the reduced order transfer function at each stage. Appendix B summarizes all the notations used in this thesis.

*A. Control Structure*

The control scheme proposed by [Narendra and Annaswamy, 1987] is shown in Figure 3.4. The controller is described completely by the following differential equations and definitions at each stage:

$$\dot{w}_{1_q} = \Lambda_q w_{1_q} + lu \tag{3.9}$$

$$\dot{w}_{2_q} = \Lambda_q w_{2_q} + l y_p \tag{3.10}$$

$$w_q \overset{\text{def}}{=} [r,\ w_{1_q}^T,\ y_p,\ w_{2_q}^T]^T \tag{3.11}$$

$$\theta_q \overset{\text{def}}{=} [k_q,\ \theta_{1_q}^T,\ \theta_{0_q},\ \theta_{2_q}^T]^T \tag{3.12}$$

$$u = \theta_q^T w_q \tag{3.13}$$

where $\theta_{1_q}, \theta_{2_q}, w_{1_q}, w_{2_q} \in \mathbb{R}^{n_q - 1}$, $k_q, \theta_{0_q} \in \mathbb{R}$, and $(\Lambda_q,\ l)$ is an asymptotically stable system in controllable canonical form with

$$\Delta_q(s) \overset{\text{def}}{=} \det(sI - \Lambda_q) = \Delta_{0_q}(s) Z_{m_q}(s) \tag{3.14}$$

for some monic Hurwitz polynomial $\Delta_{0_q}$ of degree $n_q - m_q - 1$.

Figure 3.4: Model Reference Adaptive Controller

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

$$\frac{\Delta_q(s)}{\Delta_q(s) - C_q(s)} \quad \text{and} \quad \frac{D_q(s)}{\Delta_q(s)}$$

where

$$\frac{C_q(s)}{\Delta_q(s)} = \theta_{1_q}^T (sI - \Lambda_q)^{-1} l, \tag{3.15}$$

$$\frac{D_q(s)}{\Delta_q(s)} = \theta_{0_q} + \theta_{2_q}^T (sI - \Lambda_q)^{-1} l, \tag{3.16}$$

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

$$F_{p_q}(s) = \frac{k k_p Z_p(s) \Delta_q(s)}{(\Delta_q(s) - C_q(s)) R_p(s) - k_p Z_p(s) D_q(s)} \tag{3.17}$$

$$= \frac{k k_{Gp_q} Z_{Gp_q}(s) \Delta_q(s)}{(\Delta_q(s) - C_q(s)) R_{Gp_q}(s) - k_{Gp_q} Z_{Gp_q}(s) D_q(s)}$$

$$\times \frac{k_{Gr_q} Z_{Gr_q} \left[ (\Delta_q(s) - C_q(s)) R_{Gp_q}(s) - k_{Gp_q} Z_{Gp_q}(s) D_q(s) \right]}{(\Delta_q(s) - C_q(s)) R_{p_q}(s) - k_{p_q} Z_{p_q}(s) D_q(s)} \tag{3.18}$$

$$= F_{G_q}(s) \frac{k_{Gr_q} Z_{Gr_q} \left[ (\Delta_q(s) - C_q(s)) R_{Gp_q}(s) - k_{Gp_q} Z_{Gp_q}(s) D_q(s) \right]}{(\Delta_q(s) - C_q(s)) R_{p_q}(s) - k_{p_q} Z_{p_q}(s) D_q(s)} \tag{3.19}$$

where $F_{G_q}(s)$ represents the estimated reduced order closed-loop transfer function and defined as below:

$$F_{G_q}(s) = \frac{k k_{Gp_q} Z_{Gp_q}(s) \Delta_q(s)}{(\Delta_q(s) - C_q(s)) R_{Gp_q}(s) - k_{Gp_q} Z_{Gp_q}(s) D_q(s)}. \tag{3.20}$$

First we defined the following $\Phi_{Fp_q}$ to denote the actual overall closed-loop characteristics polynomial at stage $q$ as

$$\Phi_{Fp_q}(s) = (\Delta_q(s) - C_q(s))R_p(s) - k_p Z_p(s) D_q(s) \tag{3.21}$$

and $\Phi_{Gp_q}$ to denote the estimated reduced-form of the closed-loop characteristics equation as

$$\Phi_{Gp_q}(s) = (\Delta_q(s) - C_q(s))R_{Gp_q}(s) - k_{Gp_q} Z_{Gp_q}(s) D_q(s) \tag{3.22}$$

Then Eqn.(3.18) can now be rewritten as

$$F_{p_q} = F_{G_q} \frac{k_{Gr_q} Z_{Gr_q} \Phi_{Gp_q}}{\Phi_{Fp_q}} = F_{G_q} F_{r_q} \tag{3.23}$$

where

$$F_{r_q} = \frac{k_{Gr_q} Z_{Gr_q} \Phi_{Gp_q}}{\Phi_{Fp_q}} \tag{3.24}$$

We will first assume now that the excitation frequency, $\omega$ is low so that the remaining higher order dynamics at $q+1$ to $n$ have no effect on the overall system. By making such assumption, we define the following relation:

$$F_{p_q}(j\omega) \overset{\text{def}}{=} F_{G_q}(j\omega) \tag{3.25}$$

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

$$H_{wr_q}(s, \theta_q) = \begin{bmatrix} 1 \\ (sI_q - \Lambda_q)^{-1} l G_{p_q}^{-1} F_{G_q} \\ F_{p_q} \\ (sI_q - \Lambda_q)^{-1} l F_{G_q} \end{bmatrix}. \tag{3.26}$$

### B. Nominal Representation of Reference Model

It is well known that under the above assumptions and control structure there exists an unique constant vector $\theta_q^o$ such that the closed-loop transfer function $F_{G_q}(s, \theta_q^o)$ matches $G_{m_q}(s)$ exactly. Namely, we can express the reference model as the plant $F_{G_q}(s, \theta_q)$ with the same controller at

$\theta_q = \theta_q^o$. In this representation, the regressor vector $w_{m_q}$ is given by

$$w_{m_q} = [r,\ w_{m1_q}^T,\ y_m,\ w_{m2_q}^T]^T. \tag{3.27}$$

Let $\Phi_{m_q}(s)$ be the model characteristic function, that is, the closed-loop characteristic function when $\theta = \theta_q^o$, and it can be derived that

$$\Phi_{m_q}(s) = Z_{Gp_q}(s)\Delta_{0_q}(s)R_{m_q}(s) \tag{3.28}$$

where $\Delta_{0_q}Z_{m_q} = \Delta_q$. For a given constant parameter vector $\theta$, the closed-loop transfer function is expressed as follows:

$$F_{G_p}(s) = \frac{k_q k_{Gp_q} Z_{Gp_q}\Delta_{0_q}Z_{m_q}}{k_q k_{Gp_q}\Phi_{Gp_q}} = \frac{\Phi_{m_q}Z_{m_q}}{\Phi_{Gp_q}R_{m_q}} = \frac{k_q}{k_q^o}\frac{\Phi_{m_q}(s)}{\Phi_{Gp_q}(s)}G_{m_q}(s) \tag{3.29}$$

where $k_q^o = k_{m_q}/k_{Gp_q}$ is the nominal value of gain $k$ at stage $q$. Also note that $w_{m_q}$ is the output of a stable linear time invariant system driven by $r(t)$ and its transfer function is

$$H_{w_m r_q} = H_{wr_q}(s, \theta_q^o) = \begin{bmatrix} 1 \\ (sI - \Lambda_q)^{-1} l G_{p_q}^{-1} G_{m_q} \\ G_{m_q} \\ (sI - \Lambda_q)^{-1} l G_{m_q} \end{bmatrix}. \tag{3.30}$$

*C. Output Error Dynamics*

Let us define the parameter error vector as

$$\phi_q \overset{\text{def}}{=} \theta_q - \theta_q^o. \tag{3.31}$$

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

$$e_1 = \frac{1}{k_q^o}W_m(s)\phi^T w. \tag{3.32}$$

*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

$$\dot{\phi}_q(t) = \dot{\theta}_q(t) = -e_1(t)w_q(t) \tag{3.33}$$

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, 1987]). 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 $w_q(t)$ is excited at a high frequency (e.g., [Kokotovic, et al., 1985]).

In this research, we use the gradient descent rule for adaptation. The idea of gradient descent rule is to reduce $e_1^2$ by adjusting $\theta$ along the direction of steepest descent. Namely, the gradient descent rule can be expressed as

$$\dot{\phi}_q(t) = -\frac{1}{2}\alpha\frac{\partial e_1^2}{\partial \phi_q}^T = -\alpha e_1\frac{\partial e_1}{\partial \phi_q}^T \tag{3.34}$$

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

$$\begin{aligned} \psi_q(t) &= \frac{\partial e_1}{\partial \phi_q}^T && (3.35) \\ &= \bar{F}_{G_q}(s)w && (3.36) \end{aligned}$$

where $\bar{F}_{G_q} = F_{G_q}/k_q$, that is,

$$\bar{F}_{G_q}(s) = \frac{k_{Gp_q}Z_{Gp_q}\Delta_q(s)}{(\Delta_q(s) - C_q(s))R_{Gp_q}(s) - k_{Gp_q}(s)Z_{Gp_q}(s)D_q(s)}. \tag{3.37}$$

The derivation of the above equation is provided in [Yang, 1995]

It has been empirically and analytically shown that the closed-loop stability of the gradient descent rule depends on the adaptation gain and the magnitude of the reference signal. It has also been shown that the gradient descent rule may cause instability depending on the initial values of the control parameters. In other words, the gradient descent 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 here is to show that the adaptive system can be stabilized even with the gradient descent rule if the system is excited progressively by changing the frequency content of the reference input according to the progress of the adaptation. Based on the full-order representation, [Yang, 1995] has derived a stability condition in the frequency domain for the gradient descent rule and proved 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. Also, based on the full-order representation with the stability analysis, [Yang, 1995] proved that there always exists a sequence of reference inputs that guarantee the stability for a plant with a high relative order. However, there is no direct link between the task requirement and the reference input design provided by the original progressive learning. Also, due to the extra assumptions we made from A(6)-A(9), the analyses will be different. Thus, in the next few sections, we will focus on the augmentation: Sec. 3.4 will focus on the stability issue relating to the model augmentation.

## 3.4 Stability Analysis Using Averaging Theorem

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. [Yang, 1995] has applied the average theorem and further proved the stability and its bound of using the gradient descent adaptation rule adaptation rule for the adaptive system when full-order assumption is used.

The objective of this section is to examine the stability of the reduced system when the unmodeled dynamics are within the assumptions made earlier.

The dynamics equation of the control parameter as stated in Eqn.(3.36) as follows:

$$\dot{\theta}_q(t) = \dot{\phi}_q(t) = -\frac{1}{2}\alpha\frac{\partial e_1^2}{\partial \phi_q}^T = -\alpha e_1\frac{\partial e_1}{\partial \phi_q}^T \tag{3.38}$$

where $\phi_q$, the parameter error vector, is defined as $\theta_q - \theta_q^o$ where $\theta_q^o$ is the desired parameter vector and $(\partial e_1/\partial\phi_q)^T$ is the sensitivity vector denoted by $\psi_q(t)$, and can be derived as

$$\begin{aligned} \psi_q(t) &= \frac{\partial e_1}{\partial \phi_q}^T && (3.39)\\ &= \bar{F}_{G_q}(s)w && (3.40)\end{aligned}$$

where $\bar{F}_{G_q} = F_{G_q}/k_q$, that is,

$$\bar{F}_{G_q}(s) = \frac{k_{Gp_q}Z_{Gp_q}\Delta_q(s)}{(\Delta_q(s) - C_q(s))R_{Gp_q}(s) - k_{Gp_q}(s)Z_{Gp_q}(s)D_q(s)}. \tag{3.41}$$

The derivation of the above equation can be found in [Yang, 1995].

By applying the averaging method given in [Sastry and Bodson, 1989], we can approximate the original system in Eqn.(3.38) by using an averaged system as

$$\begin{aligned} \dot{\phi}_{av_q} &= -\frac{\alpha}{k_{o_q}}\left[\lim_{T\to\infty}\frac{1}{T}\int_{t_0}^{t_0+T} G_{m_q}\phi_{av_q}^T w\bar{F}_{G_q} w dt\right] && (3.42)\\ &= -\frac{\alpha}{k_{o_q}}\left[\lim_{T\to\infty}\frac{1}{T}\int_{t_0}^{t_0+T} \bar{F}_{G_q} w G_{m_q} w^T dt\right]\phi_{av_q}. && (3.43) \end{aligned}$$

Defining $w_f \overset{\text{def}}{=} G_{m_q}w$ and $w_\theta \overset{\text{def}}{=} \bar{F}_{G_q}w$, and assuming the cross correlation between these two exists, we obtain

$$\begin{aligned} \dot{\phi}_{av_q} &= -\frac{\alpha}{k_q^o}\left[\lim_{T\to\infty}\frac{1}{T}\int_{t_0}^{t_0+T} w_f w_\theta^T dt\right]\phi_{av_q} && (3.44)\\ &= -\frac{\alpha}{k_q^o}R_{w_f w_\theta}(0)\phi_{av_q}. && (3.45) \end{aligned}$$

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 Eqn.(3.45).

Defining $S_r(d\omega)$ be the spectral measure of the reference input, we can express $R_{w_f w_\theta}(0)$ as

$$R_{w_f w_\theta}(0) = \frac{1}{2\pi k_q^o}\int |\frac{\Phi_{m_q}^H(j\omega)}{\Phi_{G_q}^H(j\omega)}|^2 |G_{m_q}(j\omega)|^2 H_{w_m r}(j\omega) H_{w_m r}^H(j\omega) S_r(d\omega) \qquad (3.46)$$

where $A^H$ denotes the Hermitian transpose of A if A is a matrix or a vector, or the conjugate of A if A is a scalar. See [Yang, 1995] for the derivation of the above expression.

We next assume the reference input $r$ is regulated and bounded as defined earlier shown in generalized form shown below:

$$r(t) = \sum_{i=1}^{N} R_i \sin(\omega_i + \varphi_i), \quad R_i > 0 \text{ for all } i. \qquad (3.47)$$

where $R_i$ and $\varphi_i$ are the amplitude and phase components of the reference input in the form of

Fourier series at frequency, $\omega_i$ and $N$ is the number of the distinct Fourier frequencies. $R_{w_f w_\theta}(0)$, then, can be expressed as

$$\begin{aligned} R_{w_f w_\theta}(0) &= \frac{1}{k_q^o}\sum_{i=1}^{N} R_i^2 |W_m(j\omega_i)|^2 |\mathcal{R}e\{\frac{\Phi_{m_q}(j\omega_i)}{\Phi_{G_q}(j\omega_i)} H_{w_m r}(j\omega_i) H_{w_m r}^H(j\omega_i)\} \\ &= \frac{1}{k_q^o}\sum_{i=1}^{N} R_i^2 |W_m(j\omega_i)|^2 |\mathcal{R}e\{\frac{\Phi_{m_q}(j\omega_i)}{\Phi_{G_q}(j\omega_i)}\} |\mathcal{R}e\{H_{w_m r}(j\omega_i) H_{w_m r}^H(j\omega_i)\} \\ &\quad -\frac{1}{k_q^o}\sum_{i=1}^{N} R_i^2 |W_m(j\omega_i)|^2 |\mathcal{I}m\{\frac{\Phi_{m_q}(j\omega_i)}{\Phi_{G_q}(j\omega_i)}\} |\mathcal{I}m\{H_{w_m r}(j\omega_i) H_{w_m r}^H(j\omega_i)\} \qquad (3.48) \end{aligned}$$

By using the Lyapunov function given as

$$V(\phi_{av_q}) = |\phi_{av_q}|^2 \qquad (3.49)$$

and Eqn.(3.45),

$$\dot{V}(\phi_{av_q}) = -\frac{\alpha}{2}\phi_{av_q}^T [R_{w_f w_\theta}(0) + R_{w_f w_\theta}(0)^T]\phi_{av_q}. \qquad (3.50)$$

As stated in [Yang, 1995], 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 $\lambda_{\text{min}}$ be the smallest eigenvalues of $R_{w_f w_\theta}(0)$ within the stability range of $\phi_{av}$, we get

$$-\dot{V}(\phi_{av_q}) \geq \alpha\lambda_{\text{min}} V(\phi_{av_q}) > 0. \qquad (3.51)$$

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

From Eqn.(3.48), the symmetric part of $R_{w_f w_\theta}(0)$ can be expressed as

$$\frac{1}{2}\left[R_{w_f w_\theta}(0) + R_{w_f w_\theta}(0)^T\right] = \frac{1}{k_q^o}\sum_{i=1}^{N} R_i^2 |W_m(j\omega_i)|^2 |\mathcal{R}e\{\frac{\Phi_{m_q}(j\omega_i)}{\Phi_{G_q}(j\omega_i)} H_{w_m r}(j\omega_i) H_{w_m r}^H(j\omega_i)\} \qquad (3.52)$$

Assume that there exist $2n$ linearly independent vectors in vector $H_{w_m r}(j\omega_i), i = 1, \cdots, N$ where $N \geq 2n$. Namely, vectors $H_{w_m r}(j\omega_i)$ span the whole $2n$-dimensional vector space. Then, since $R_i^2|W_m(j\omega_i)|^2$ is strictly positive given any $\phi_{av_q}$ bounded, the averaging theorem automatically prove the following stability condition as below :

**Theorem 1** *Suppose assumptions (A1) to (A9) for the adaptive system are met, then, the original system given in Eqn.(3.45) is exponentially stable, if*

$$\mathcal{R}e\{\frac{\Phi_{m_q}(j\omega_i)}{\Phi_{G_q}(j\omega_i)}\} > 0 \qquad \textit{for all } \omega_i, \tag{3.53}$$

*or*

$$|\angle\{\Phi_{m_q}(j\omega)\} - \angle\{\Phi_{G_q}(j\omega)\}| < \frac{\pi}{2} \qquad \textit{for all } \omega_i, \tag{3.54}$$

The idea of progressive learning is that the system is excited in low frequencies in the beginning to avoid the instability and the frequency range for stability is expanded gradually according to the progress of learning. Defining $\Omega_\theta$ be the frequency range of stability as

$$\Omega_\theta = \left\{\omega \mid |\angle\{\Phi_{m_q}(j\omega)\} - \angle\{\Phi_{G_q}(j\omega)\}| < \frac{\pi}{2}\right\} \tag{3.55}$$

the main theoretical results that supports the above argument is given as:

**Theorem 2 (Progressive Excitation Theorem)** *Suppose, for a given $\theta^{(h-1)}$, there exists $\omega^{(h)}$ such that*

$$\Omega_{\theta^{(h-1)}} = \{\omega | 0 \leq \omega \leq \omega^{(h)}\}. \tag{3.56}$$

*where the superscript h stands for the hth trial. 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*

$$\omega^{(h+1)} = \omega^{(h)} + \varepsilon, \quad \textit{and} \tag{3.57}$$

$$|\angle\{\Phi_m(j\omega)\} - \angle\{\Phi_{\theta^{(h)}}(j\omega)\}| < \frac{\pi}{2} \ \textit{for all} \ \ \omega \in \{\omega|\ 0 \leq \omega \leq \omega_{\theta^{(h+1)}}\}. \tag{3.58}$$

*where $\varepsilon$ is originally defined as the increased expansion frequency from hth trial to $h+1$ trail.*

See [Yang, 1995] for the proof. This theory stated that as long as we follow the stability condition stated earlier to excite the system, we can always achieve a possible increment in expansion of the frequency range. Here, we extend the original definition of $\varepsilon$ to its implied meaning as the increment in frequency of the matched phase region. Another word, $\varepsilon$ will represent the amount of the frequency range where the phase plot of the closed-loop transfer function in the region of excitation has improved after each trial.

Untill now, we have assumed that the unmodeled dynamics does not make any contribution to the stability. Next, we are going to rederive the stability condition with this assumption removed and relax several assumptions we made earlier in order to suit it to the actual implementation.

## 3.5 Robustness of Augmentation

In the previous sections, we have introduced the systems under ideal conditions. Since our research goal is to apply the proposed algorithm to real physical systems, we need to consider the robustness of the proposed control algorithm under two "idealized" real physical effects: the unmodeled dynamics, and the sensor output noise, which will be the main focus of this section. Here, we will retain the same notations that we have used throughout this thesis.

Most of the adaptive control algorithms we see today are mainly focused on the ideal case for systems, namely, when assumption A(1)-A(6) are followed. However, this is usually never true in any actual physical system. In most cases, the system order of the actual physical system, $W_p(s)$ that we assumed to be $n_n$ earlier is unknown. At the same time, the designer dose not have a detailed state-space model of the plant to be controlled, either because it is too complex or its dynamics is not well understood. It is based on this available information as well as the performance requirements that the designer determines the relevant frequency bandwidth of the system. Through the interpretation of the available open-loop bode plot, the designer then determines the plant model, $G_{p_q}(s)$ which is the lowest order of $n_q$, $n_q < n_n$, and which is valid within the frequency range determined. The designer then designs an appropriate reference model, $G_{m_q}(s)$ which has order of $n_q$, and the controller as in the case of the MRAC to control the actual system, $W_p(s)$. In our definition, $W_p(s) = G_{p_q}(s)G_{r_q}(s)$ where the unmodeled dynamics, $G_{r_q}(s)$, has order of $n_u$, $n_u = n_n - n_q$. Together, they can be summarized in Figure 3.5. In the actual system, instead of the noise-free nominal output, $y_{p_s}(t)$, the clean sensor output, $y_p(t)$, is usually corrupted with sensor noise, $v(t)$. One

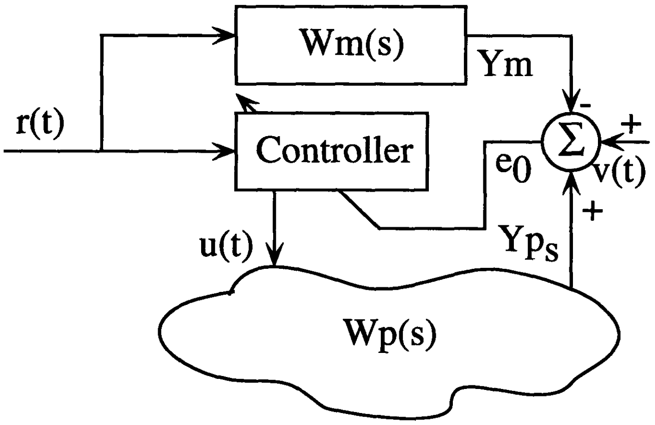


Figure 3.5: Model following control system

important thing to note is that $n_q$ is only an assumption based on the designer's own interpretation in the predetermined frequency range of interest. Thus, naturally, there exist an $\omega_{H_q}$, the marginal stability frequency, for the assumed model to be "sufficiently" representing the actual system. As we push for wider system bandwidth, we need to vary the assumed system order for the plant from $q$ to $j$ where $j > q$. By increasing to another stage, $j$, we can then achieve a higher $\omega_{H_j}$ with a higher controller complexity $\theta_j \in \mathrm{I\!R}^{2 \times n_j}$ with a wider bandwidth as the task requires.

In this section, we will examine the robustness of the proposed algorithm under these effects:

- presence of the sensor noise,
- presence of the unmodeled dynamics,

in terms of the local stability. We then validate the robustness with a simulation.

The robustness which will be introduced is defined to be that the controller parameter adaptation dynamics is ineffective by the presence of unmodeled dynamics as well as the presence of noise. The robustness issue was first raised by [Rohrs, et al., 1982] and [Rohrs, et al., 1985] when they showed that several algorithms can become unstable when some of the assumptions required by the stability proofs are not satisfied. Since then, many researchers have been working on how to modify the existing adaptive control algorithms, mainly on the adaptive law, to expand the robustness margin. To name a few, *hybrid adaptive control* is first presented by [Gawthrop, 1980], *dead zone* is presented by [Peterson and Narendra, 1982], [Sastry, 1984], [Narendra and Annaswamy, 1986],[Narendra and Annaswamy, 1986], etc., *$\sigma$-modification* is presented by [Ioannou and Kokotivic, 1983], and *regressor vector filtering* by [Wittenmark and Astrom, 1984]. It is our intention here to examine the robustness of our algorithm without changing the adaptive law chosen which is the gradient descent rule. It, in many cases as presented by many researchers, is the worse adaptive law since it may introduce instability even without the presence of these effects. The main reason that we still use the gradient descent rule instead of other adaptation rules is so that we can show the effectiveness of our algorithm is robust enough even for the worse adaptation rule. Nevertheless, as pointed out by [Sastry and Bodson, 1989], robustness depends on the plant, the controller as well as on reference inputs. We will show how each item listed above affects the stability and how we can ensure the robustness by setting constrains on them from the stability analysis to be introduced later. We are now in position to examine the robustness to each of the cases itemized earlier in the following sections.

### 3.5.1 Problem Reformulation for Robustness

As presented earlier, in each stage of the approximation, based on the controller structure, $\theta_q$, the observer polynomial $\Delta_q$, the currently projected plant, $G_{p_q}$, and the desired reference model, $G_{m_q}$, we can obtain the desired optimal control parameters $\theta_q^o$ such that the closed-loop transfer function $F_{Gp_q}$ equals to the desired reference plant, $G_{m_q}$ or

$$F_{Gp_q}(s, \theta_q^o) = G_{m_q} \tag{3.59}$$

where

$$\theta_q^o = \left[k_{c_q}^o, \quad \theta_{1_q}^o, \quad \theta_{0_q}^o, \quad \theta_{2_q}^o\right] \in \left[\mathbb{R}, \quad \mathbb{R}^{n_q - 1}, \quad \mathbb{R}, \quad \mathbb{R}^{n_q - 1}\right] \tag{3.60}$$

$$\chi = \left[1, \quad s, \quad \cdots, \quad s^{n_q - 1}\right] \tag{3.61}$$

$$\Delta_q = Det\left[s\mathbf{I} - \Lambda_q\right] \tag{3.62}$$

$$F_{Gp_q}(s, \theta_q^o) \overset{\text{def}}{=} \frac{k_{c_q}^o G_{p_q} \Delta_q}{(\Delta_q - C_q^o) - G_{p_q} D_q^o} \tag{3.63}$$

$$C_q^o = \theta_{1_q}^o \chi \tag{3.64}$$

$$D_q^o = \theta_{2_q}^o \chi + \theta_{0_q}^o \Delta_q \tag{3.65}$$

Nevertheless, in the actual situation, the neglected plant transfer function, $G_{r_q}$, will contribute to the parameter adaptation dynamics. Thus, instead of the $\theta_q^o$, we will obtain a set of deviated control parameters, $\theta_{t_q}$, such that it stabilizes the overall closed-loop transfer function $F_{p_q}$ with the controller parameters, $\theta_q$ set to $\theta_{t_q}$. Thus, this set of deviated parameters, $\theta_{t_q}$, is tuned to reflect the actual system including the unmodeled dynamics, $G_{r_q}$. As long as the effect of unmodeled dynamics is small, the tuned plant transfer function that approximately matches the model transfer function, $G_{m_q}$ can be achieved as long as the persistent excitation condition is satisfied as well. This implies that $\theta_{t_q}$ will be very close to the $\theta_q^o$ since the effect from the unmodeled dynamics is negligible at low frequency. As pointed out by [Sastry and Bodson, 1989] in terms of *partial matching*, $\theta_{t_q}$ is derived based on the elastic averaging between the actual system and the reference model at each excitation frequency. As described earlier, for a given $q$th stage controller with order of $2n_q - 1$, complete one-to-one matching between the closed transfer function based on the reduced order plant, $F_{G_q}$, and the reference plant, $G_{m_q}$ is possible. However, the actual system with the $q$th stage controller has order of $n_q + n_n - 1$ instead of $2n_q - 1$. Thus, complete matching between $G_{m_q}$ and the actual closed-loop transfer function, $F_{p_q}$ that includes the unmodeled dynamics, $G_{r_q}$ is questionable. To differentiate $G_{m_q}$ from the tuned plant, we will use the notation $F_{t_q}$ to represent the tuned plant.

$$
\begin{aligned}
F_{t_q} &= \frac{k_{t_q} W_p \Delta_q}{(\Delta_q - C_{t_q}) - W_p D_{t_q}} && (3.66) \\
&\overset{\text{def}}{=} G_{m_q} F_{r_{t_q}} && (3.67)
\end{aligned}
$$

where $F_{r_{t_q}}$ represents the residual transfer function and has the following relationship

$$
F_{r_{t_q}} = \left(\frac{k_{t_q}}{k_q^o}\right) G_{\underset{r}{p_q}} \left[\frac{\Delta_q - C_q^o - G_{p_q} D_q^o}{\Delta_q - C_{t_q} - W_p D_{t_q}}\right] \qquad (3.68)
$$

$F_{r_{t_q}}$ as explained earlier.

In the single excitation frequency case, clearly, in the region where unmodeled dynamics is small, we can still achieve complete matching. As we start to excite the unmodeled dynamics, $\theta_{t_q}$ will start to drift to compensate for the contribution from the unmodeled dynamics. This slow drift characteristics is due to the internal matching between the plant transfer function and the desired transfer function where the closed-loop transfer function has higher order than the desired transfer function. Similar drift characteristics is also found by [Sastry and Bodson, 1989] and [Åström and Wittenmark, 1989].

In the multiple excitation frequency case, due to varying $\theta_{t_q}$ achieved at various excitation frequencies at the region where $\theta_{t_q}$ starts to drift, complete matching is not possible to satisfy all excitation frequencies. Thus, it is important to limit the excitation frequencies to be within the region where $\theta_{t_q}(j\omega)$ remains almost invariant. The same interpretation can be taken in Eqn.(3.67) is that as long as the excitation frequency, $\omega$, is within a frequency range $\Omega_s \in \{\omega | 0 < \omega < \omega_s\}$ where contribution from $F_{r_{t_q}}(j\omega)$ is within a given bound, $\theta_{t_q}$ obtained after elastic averaging of all the excitation frequencies will remain almost invariant. It is this elastic averaging effect, though we can still achieve stable tuning behind $\omega_s$ in the single excitation frequency case, oscillation around the elastic averaged $\theta_{t_q}$ based on the multiple excitation frequencies can be expected. Due to this fact, $\omega_s$ is thus defined as the switching frequency for $q$th reference model and controller to achieve a set of converged control parameter, $\theta_{t_q}$, without augmenting.

The reason for maintaining parameters to remain invariant around the tuned parameters, $\theta_{t_q}$, is that due to the fact that for any given controller and physical plant, there always exists a bound on the variation of the control parameters such that the system is stable as in the case of the *Routh's Stability Criterion*[Ogata, 1990]. *Robustness Control* further expands the stability criterion to include the parameter variations in the form of the bound on parameters. Based on Eqn.(3.67), we see clearly, as $F_{r_{t_q}}$ becomes excited more and more as we push for higher excitation frequency, the tuned parameter, $\theta_{t_q}$ which "stabilizes" plant will eventually drift into instability at $\omega_{H_q}$ where $\omega_{H_q}$defines the absolute limit where stable parameter tuning can be achieved based on the stability analysis to be presented later. Thus, it is important that we maintain a bound on parameters for the parameter dynamics to remain stable despite the presence of unmodeled dynamics and the presence of the output noise. We can then directly derive a bound obtained from the well-established robustness bound on system variation for a given controller. Since the excitation frequency is directly related to parameter dynamics and we have full control of it, our attention has now shifted to derive a robustness bound on the excitation frequency. The definition of the robustness we will derive is a bound that the control parameter will remain bounded with the presence of the unmodeled dynamics and noise as well.

A similar robustness bound of the existence of such tuned parameters, $\theta_{t_q}$ with the presence of the unmodeled dynamics and output noise based on the SRP rule is stated in [Sastry and Bodson, 1989] and [Anderson, et. al, 1986]. In Appendix A, we restated several theorems used in[Sastry and Bodson, 1989]'s for future references.

Next, we are going to follow the same formulations but restated in our situation through the use of the averaging theorem and tie it to the local stability we have derived earlier to state the bound of the robustness based on the gradient descent rule instead of SPR.

### 3.5.2 Derivation of the Robustness Bound

First of all, we would like to defined an additive assumption to the open- and closed-loop system besides the multiplicative assumption we have made earlier:

$$\begin{aligned} W_p &= G_{p_q} + G_{ad_{r_q}} && \text{open-loop, additive uncertainty} \\ &= G_{p_q} G_{r_q} && \text{open-loop, multiplicative uncertainty} \\ F_{p_q} &= F_{Gp_q} + F_{ar_q} && \text{closed-loop, additive uncertainty} \\ &= F_{Gp_q} F_{r_q} && \text{open-loop, multiplicative uncertainty} \end{aligned} \tag{3.69}$$

and from the earlier derivation of the regressor dynamics equation $w_q$, we can obtain

$$w_q = \begin{bmatrix} r \\ w_1 \\ y_p \\ w_2 \end{bmatrix} = \begin{bmatrix} r \\ [s - \Lambda]^{-1} lu \\ y_p \\ [s - \Lambda]^{-1} ly_p \end{bmatrix} \tag{3.70}$$

Since in our case, we have only the nominal information of the system, the regressor will contain the nominal information plus the noise term, $v$. If we make further assumption that the noise is only linear superposition of the unmodeled dynamics plus the actual output disturbance, $v$, we can rewrite the above regressor equation based on the derivation of $w_q$ as below:

$$u = W_p^{-1}(F_{Gp_q} + F_{ar_q})r \tag{3.71}$$

$$= W_p^{-1}(F_{p_q})r \tag{3.72}$$

$$y_p = F_{Gp_q} r + F_{ar_q} r + v \tag{3.73}$$

$$= (F_{p_q})r + v \tag{3.74}$$

$$w_q = \begin{bmatrix} 1 \\ [s\mathbf{I} - \Lambda]^{-1} l W_p^{-1}(F_{p_q}) \\ F_{p_q} \\ [s\mathbf{I} - \Lambda]^{-1} l F_{p_q} \end{bmatrix} r + \begin{bmatrix} 0 \\ 0 \\ 1 \\ [s - \Lambda]^{-1} l \end{bmatrix} v \tag{3.75}$$

$$= H_{wr} r + H_{wv} v \tag{3.76}$$

Then we can use the same averaging theorem to derive the following equation between the reference and actual system:

$$\begin{aligned} \dot{\phi}_{av_q} &= -\frac{\alpha}{k_{t_q}} \left[ \lim_{T\to\infty} \int_{to}^{to+T} \bar{F}_{p_q} w F_{t_q} w^T dt \right] \phi_{av_q} && (3.77) \\ &= -\frac{\alpha}{k_{t_q}} R_{w_f w_\theta}(0) \phi_{av_q} && (3.78) \end{aligned}$$

where $\bar{F}_{p_q} = F_{p_q}/k_{t_q}$, that is,

$$\bar{F}_{p_q}(s) = \frac{k_p Z_p \Delta_q(s)}{(\Delta_q(s) - C_q(s))R_p(s) - k_p(s) Z_p(s) D_q(s)}. \qquad (3.79)$$

and

$$R_{w_f w_\theta}(0) = \frac{1}{2\pi} \int S_{w_f w_\theta}(d\omega). \qquad (3.80)$$

We further assume that the reference signal $r$ and the sensor noise $v$ are both stationary and have no correlation between them or $S(rv) = 0$.Then, we can derive the cross-correlation of the regressor, $S_w$, as follow:

$$S_w(d\omega) = H_{wr}(j\omega) H^H_{wr}(j\omega) S_r(d\omega) + H_{wv}(j\omega) H^H_{wv}(j\omega) S_v(d\omega) \qquad (3.81)$$

Since $w_\theta = \frac{\bar{F}_{pq}}{F_{t_q}} w_f$, then the cross spectral measure $S_{w_f w_\theta}(d\omega)$ is given by

$$\begin{aligned} S_{w_f w_\theta}(d\omega) &= \left( \frac{\bar{F}_{p_q}}{F_{t_q}} \right)^H S_{w_f}(d\omega) && (3.82) \\ &= \bar{F}^H_{p_q} F_{t_q} S_w(d\omega) && (3.83) \end{aligned}$$

$$\begin{aligned} S_{w_f w_\theta}(d\omega) &= H_{w_m r}(j\omega) H^H_{w_m r}(j\omega) \bar{F}^H_{p_q}(j\omega) F_{t_q}(j\omega) S_r(d\omega) \\ &\quad + H_{wv}(j\omega) H^H_{wv}(j\omega) \bar{F}^H_{p_q}(j\omega) F_{t_q}(j\omega) S_v(d\omega) && (3.84) \\ &= \frac{1}{k_{t_q}} |F_{t_q}(j\omega)|^2 \frac{\Phi^H_{t_q}(j\omega)}{\Phi^H_{F_{p_q}}(j\omega)} H_{w_m r}(j\omega) H^H_{w_m r}(j\omega) S_r(d\omega) \\ &\quad + \frac{1}{k_{t_q}} |F_{t_q}(j\omega)|^2 \frac{\Phi^H_{t_q}(j\omega)}{\Phi^H_{F_{p_q}}(j\omega)} H_{wv}(j\omega) H^H_{wv}(j\omega) S_v(d\omega) && (3.85) \end{aligned}$$

Finally, we have obtained the symmetric part of $R_{w_f w_\theta}(0)$ with the assumption of the piece-wise reference input, $r$ and noise, $v$ as below:

$$\begin{aligned}\frac{1}{2}\left[R_{w_f w_\theta}(0) + R_{w_f w_\theta}(0)^T\right] &= \frac{\alpha}{k_{t_q}}\sum_{i=1}^{N} R_i^2 |F_{t_q}(j\omega_i)|^2 \mathcal{R}e\{\frac{\Phi_{t_q}(j\omega)}{\Phi_{Fp_q}(j\omega)}\}\mathcal{R}e\{H_{w_m r}(j\omega_i)H^H_{w_m r}(j\omega_i)\} + \\ &\quad \frac{\alpha}{k_{t_q}}\sum_{j=1}^{M} v_j^2 |F_{t_q}(j\omega_j)|^2 \mathcal{R}e\{\frac{\Phi_{t_q}(j\omega)}{\Phi_{Fp_q}(j\omega)}\}\mathcal{R}e\{H_{wv}(j\omega_j)H^H_{wv}(j\omega_j)\} \qquad (3.86) \\ &= R_r + R_v \qquad (3.87)\end{aligned}$$

where $R_r$ is all the terms that depend on signal $r$ and $R_v$ is all the terms that depend on signal $v$. $M$ is the number of the noise spectrum. We will state this $M$ later. Based on the same Lyapunov function derived earlier, we can start to exam the stability and robustness based on the newly derived $R_{w_f w_\theta}$ as shown in Eqn.(3.87).

### 3.5.2.1 Robustness to Unmodeled Dynamics

In this section, we will examine the stability based on Eqn.(3.87) where the system does include some unmodeled dynamics where $n_q < n_n$ but no noise, thus, the second summation in Eqn.(3.87) will drop out and the symmetric part of $R_{w_f w_\theta}$, will be left with the following term which is very similar to the nominal term:

$$frac12 \left[R_{w_f w_\theta}(0) + R_{w_f w_\theta}(0)^T\right] = \frac{\alpha}{k_{t_q}}\sum_{i=1}^{N} R_i^2 |F_{t_q}(j\omega_i)|^2 \mathcal{R}e\{\frac{\Phi_{t_q}(j\omega)}{\Phi_{Fp_q}(j\omega)}\}\mathcal{R}e\{H_{w_m r}(j\omega_i)H^H_{w_m r}(j\omega_i)\} \qquad (3.88)$$

In Eqn.(3.88), we see a term that is similar to the nominal term required for the stability as follow:

$$|\angle\{\Phi_{t_q}(j\omega_i)\} - \angle\{\Phi_{Fp_q}(j\omega_i)\}| < \frac{\pi}{2} \qquad (3.89)$$

Eqn.(3.89) is different than the stability condition derived earlier due to the difference between $G_{m_q}(j\omega)$ and $F_{t_q}(j\omega)$. Due to the fact that $F_{t_q}(j\omega)$ is not available as a measured signal, we need to relate it to the $G_{m_q}(j\omega)$. Based on an earlier assumption that

$$F_{t_q}(j\omega) = G_{m_q}(j\omega)F_{rt_q}(j\omega) \tag{3.90}$$

and derivation shown in Appendix C, we can obtain the following:

$$\begin{aligned} \frac{\Phi_{t_q}}{\Phi_{p_q}} &= \frac{k_p k_c Z_p \Delta_{0_q} Z_{m_q}}{\Phi_{p_q}} \frac{1}{G_{m_q} F_{rt_q}} && (3.91) \\ &= \frac{F_{p_q}}{G_{m_q} F_{rt_q}} && (3.92) \\ \angle\{\frac{\Phi_{t_q}}{\Phi_{p_q}}\} &= \angle\{F_{p_q}\} - \angle\{G_{m_q}\} - \angle\{F_{rt_q}\} && (3.93) \end{aligned}$$

Since the phase angle on $F_{rt_q}$ could be both positive or negative, we then rearrange the above equation into the following form:

$$|\angle\{F_{p_q}\} - \angle\{G_{m_q}\}| < \frac{\pi}{2} - |\Gamma_{r_q}| \tag{3.94}$$

where $\Gamma_{r_q} = \angle\{F_{rt_q}\}$. Thus, this defined the maximum allowable excitation frequency $\omega_{H_q}$ such that

$$|\Gamma_{r_q}(j\omega_{H_q})| = \frac{\pi}{2} \tag{3.95}$$

for us to achieve a stable tuning. The definition of $\omega_{H_q}$ becomes even self-evident in the single excitation frequency case as described earlier. In the single excitation frequency case, the reference input is composed of only one frequency; in other words, it is scalar. In order for the control parameter adaptation dynamics to be stable, $R_{w_f w_\theta}(0)$, which is also a scalar quantity, has to be greater than zero. Thus, the constraint on the excitation frequency to be less than $\omega_{H_q}$ has become both a sufficient and necessary condition.

In the multiple excitation frequency case as stated earlier, oscillation of parameters around the tuned parameters is expected unless we further constrain the excitation frequency range to be less than $\omega_s$. As stated earlier, our goal is to derive a bound such that the tuned parameters remain almost stationary. This can be easily achieved by minimizing the extra phase angle introduced by the residual closed-loop transfer function, $F_{rt_q}$. We can only achieve this by exciting the system within the range of $\Omega_s \in \{\omega | 0 < \omega < \omega_s\}$ where $\omega_s$ is defined as

$$|\Gamma_{r_q}(j\omega_s)| = \Gamma_{max} \tag{3.96}$$

where $\Gamma_{max}$ is the maximum allowable phase prespecified to achieve minimal oscillation of the tuned parameters. Thus the conservative robustness bound $\omega_{rob_q}$ is then defined as

$$\omega_{rob_q} \stackrel{\text{def}}{=} \omega_{H_q} - \omega_{s_q}. \tag{3.97}$$

As long as we can satisfy this robustness frequency bound, the stability can be achieved and improvement on $\varepsilon$, the increment on the matched frequency range can be achieved even with the presence of the unmodeled dynamics.

### 3.5.2.2 Robustness to both Sensor Noise and Unmodeled dynamics

In this section, we will examine the stability based on Eqn.(3.87) where system includes both the sensor noise and the unmodeled dynamics. However, due to there are several types of output disturbance noise, we will later further distinguish them in this section.

In general, we have the following

$$R_{w_f w_\theta} = R_r + R_v, \tag{3.98}$$

and the noise is white noise or $M = \infty$. In order for the parameter dynamics to remain stable, all eigenvalues of $R_{w_f w_\theta}$ have to remain positive. To make an even stronger constrain on the system we can rewrite the above equation as below:

$$R_{w_f w_\theta} = R_r \left[\mathbf{I} + R_r^{-1} R_v\right] \tag{3.99}$$

so that the variation of the $\dot{\theta} = \dot{\phi}$ is greatly depending on the nominal terms, or

$$||R_r^{-1} R_v|| \ll 1. \tag{3.100}$$

The physical meaning of the above statement is that as long as the correlation between the reference and the adaptive dynamics dominates the correlation between the noise and the adaptive dynamics. Another word, the effect of the noise is negligible as compared to the reference signals. Based on the relation for the norm, we obtained

$$||R_r^{-1}R_v|| \leq ||R_r^{-1}||||R_v||. \tag{3.101}$$

Further more, based on the relation between the norm and the singular value, we obtained:

$$||R_r^{-1}||||R_v|| = \frac{\sigma_{max}(R_v)}{\sigma_{min}(R_r)} \tag{3.102}$$

where $\sigma_{min}(R_r)$ is the minimum singular value from the single value decomposition of $R_r$. Then as long as

$$\sigma_{min}(R_r) \gg \sigma_{max}(R_v), \tag{3.103}$$

then the adaptive system is robust to the output disturbance noise. By specifying a bound on $\sigma_{max}(R_v)$ to be $b_v$ and a given allowable ratio, $\zeta$, then we can see that as long as

$$\sigma_{min}(R_r) \geq \zeta b_v > 0, \tag{3.104}$$

then the adaptive system is robust to the output disturbance noise besides satisfying the $\omega_{rob}$ stated earlier.

Next, we are going to illustrate the robustness analysis presented above with a simulation.

### 3.5.3 Robustness Simulation

Now that we have introduced the robustness based on the stability of the parameter dynamics, we will use a simulation to demonstrate the derived robustness bound.

The simulation case we used is the same simulation example stated in [Rohrs, et al., 1982] to demonstrate the robustness. The overall MRAC system can be seen in Figure 3.6. The plant used in the simulation is presented in Tab.3.1.

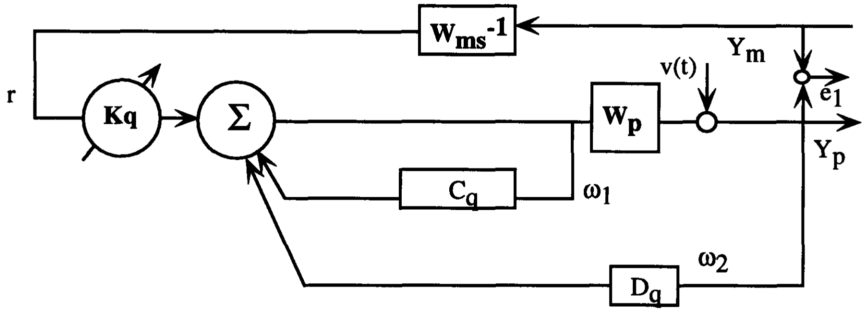


Figure 3.6: Overall simulated MRAC system

Table 3.1: Robustness simulation setup

| Name | Command Setup |
|---|---|
| $G_{m_1}$ | $\frac{3}{s+3}$ |
| $W_p$ | $\frac{2}{s+1}\frac{229}{s^2+30s+229}$ |
| $v$ | $A\sin\omega t \qquad 0 < A < 0.2, \qquad 0 < \omega < 94.25r/s$ |

### 3.5.3.1 Robustness Simulation and Results

Based on the setup, the third order system is being estimated using only a first-order plant model where

$$G_{p_1}(s) = \frac{\hat{b}}{s+\hat{a}}. \tag{3.105}$$

At each excitation frequency, the actual third-order system as seen using the first-order system model can be derived as below:

$$W_p(j\omega_r) = \frac{2}{j\omega_r+1}\frac{229}{-\omega_r^2+30\omega_r j+229} \tag{3.106}$$

$$= \frac{458}{(259\omega_r-\omega_r^3)j+(229-31\omega_r^2)} \tag{3.107}$$

$$= \frac{\frac{458}{259-\omega_r^2}}{(j\omega_r+\frac{229-31\omega_r^2}{259-\omega_r^2})}. \tag{3.108}$$

Thus, the estimated plant parameters based on the first-order system model are:

$$\hat{b} = \frac{458}{259-\omega_r^2}; \qquad \hat{a} = \frac{229-31\omega_r^2}{259-\omega_r^2} \tag{3.109}$$

where $\omega_r$ is the excitation frequency. The tuned steady state parameters can then be calculated as

$$k = \frac{3}{\hat{b}}; \qquad \theta_0 = -\frac{3-\hat{a}}{\hat{b}}; \tag{3.110}$$

Due to the fact that there are only two parameters that need to be tuned, the reference input needs to have only one excitation frequency with the exception of constant gain. As described earlier, in the single excitation frequency case, excitation up and to the marginal stability frequency, $\omega_H$ has become both necessary and sufficient requirement for the parameter dynamics to remain stable. Based on the excitation frequency, $\omega_r$, we can obtain Figure 3.7.Figure 3.8 shows the bode-plot for the $W_p$,$G_{m_1}$, $F_{t_1}$, and $F_{r_1}$ where

$$F_{r_{t1}} = 1.18\frac{s+2(1-\theta_0)}{s^3+31s^2+289s+458(1-\theta_0)}. \tag{3.111}$$

From the phase plot, we see that $F_{r_t}$ starts making contribution to the overall phase plot around $1.0rad/s$ and its phase reaches $90^o$ around $8rad/s$. Based on these two plots, we see that the stabilized, or tuned, parameters are very closed to the projected parameters as long as the excitation is within $1.5rad/s$. From that point on, the steady state parameter will start to drift. However, the system will become unstable when $\theta_0 \geq 0.5$ and $\theta_0 \leq -17.03$. Thus, even if it is still able to find steady state tuned parameters, the system may not be stable. This happens when the excitation frequency goes beyond $\omega_H$, where $\angle\{F_{r_1}\}$ reaches $90^0$, or at around $10rad/s$.

To further demonstrate the effect, we excite the system with a single excitation frequency ranging from $0.5rad/s$ to $8rad/s$. We also added the noise with the amplitude equals to 10% of the excitation amplitude. We then plotted the tracking error to the norm of the parameter error $|\phi|$ in Figure 3.9 where the noise is not present and Figure 3.10 where the noise is present for all the cases. At $6rad/s$, we still can achieve both tuning and tracking though the presence of the unmodeled dynamics. With the added noise, we can only achieve the same performance when the excitation is within $4rad/s$. Even at $4rad/s$, $\angle\{F_{r_1}(j\omega)\}$ is greater than$45^0$. Thus, as long as the $\omega_r$ is within $\omega_s$, then the system is not affected by the noise and unmodeled dynamics.

In this section, we have derived some theoretical results for the parameter dynamics robustness. This simulation further demonstrated some of the theoretical results we stated earlier. Next, we will

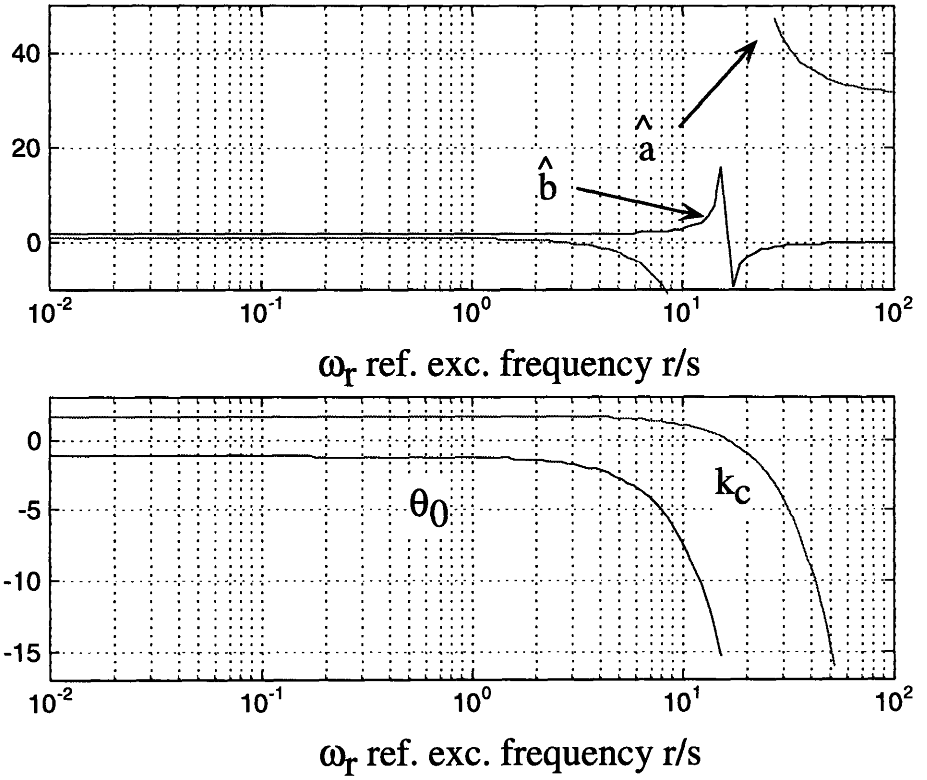


Figure 3.7: Estimated parameter and tuned parameter as function of the excitation

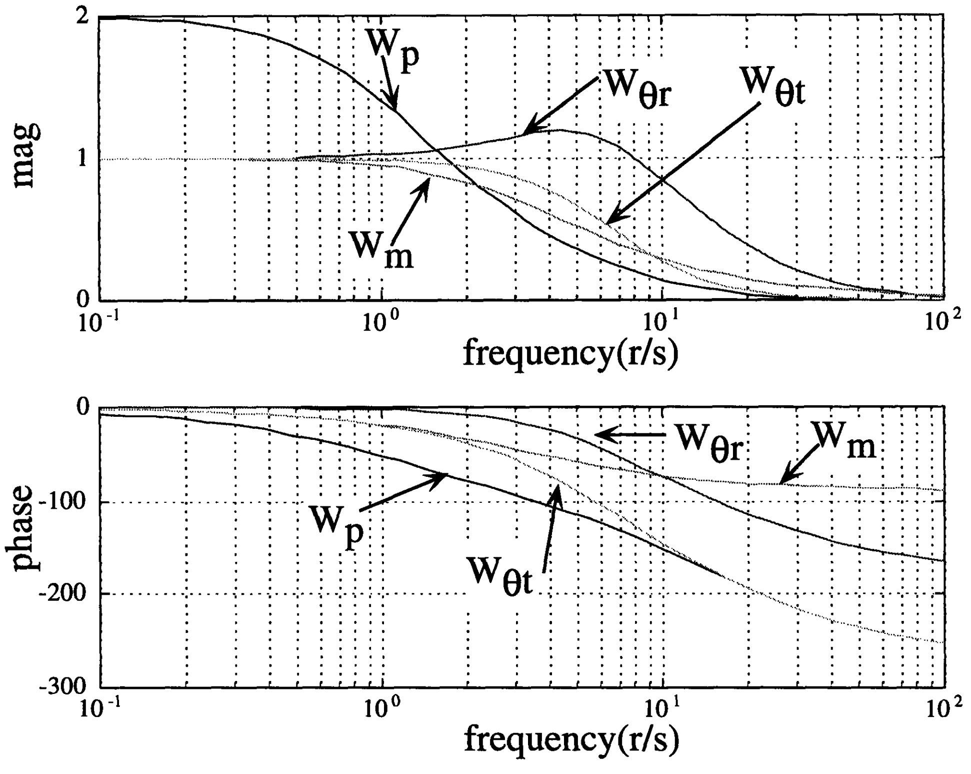


Figure 3.8: Bode plot of the system

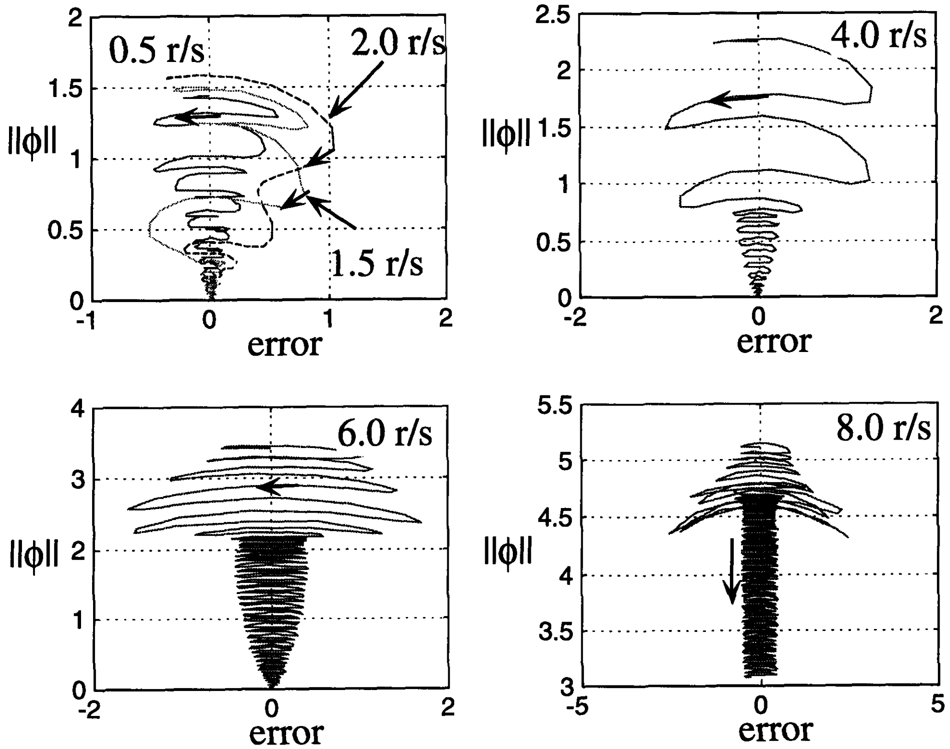


Figure 3.9: Norm parameter error to tracking error

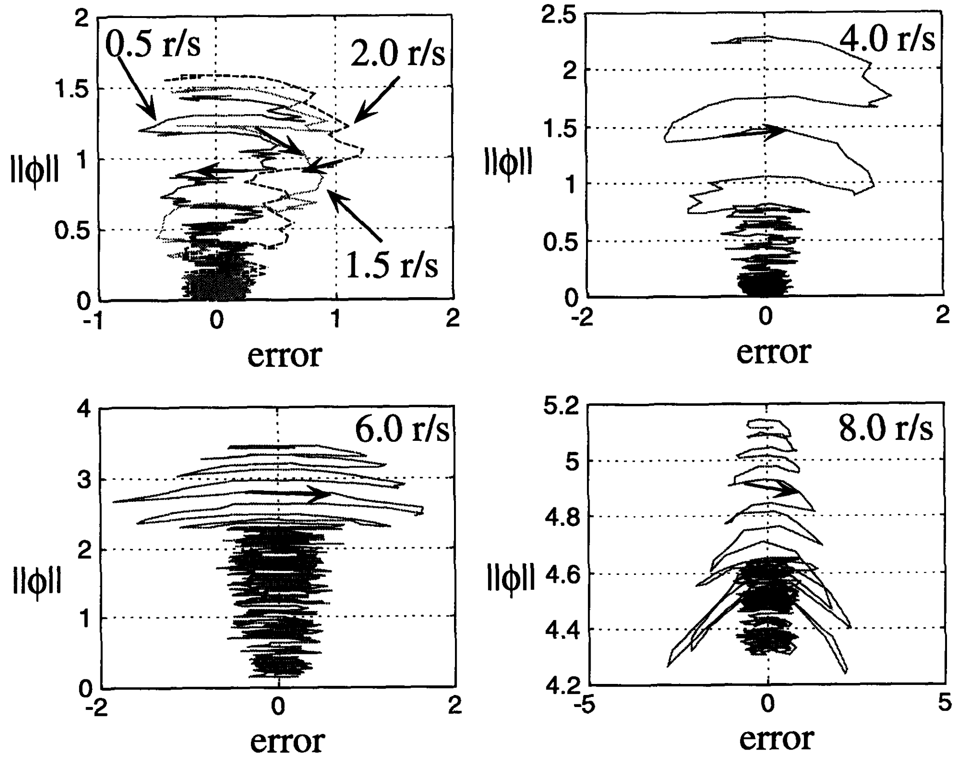


Figure 3.10: Norm parameter error to tracking error with the presence of the noise

describe how we design the stable reference inputs based on the given task.

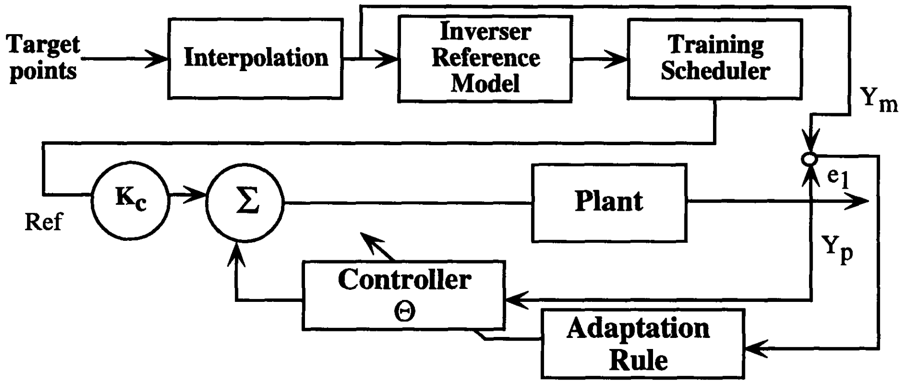


Figure 3.11: Progressive learning with trajectory synthesis

## 3.6 Trajectory Synthesis Using Progressive Learning

In this section, the trajectory synthesis problem will be formulated in order to determine the maximum increment of excitation frequency. In the design of the trajectory, there are two main issues involves: task fulfillment issue where the designed desired trajectory needs to meet the task requirements, and stable excitation issue, where excitation needs to guarantee stable tuning of the control parameters from the given set of target points. The main purpose of the trajectory synthesis is to tune the system with a known excitation stability range and to have the system performing actual tasks without being shutting down. The overview of the trajectory synthesis as part of the progressive tuning is shown in Figure 3.11 in particular the top portion of the graphics where the task, or the target points, are been transformed into stable reference inputs to the system. It is important to note that, due to the fact that we are performing the adaptive control on the system level, although the system is tuned to a specific geometric trajectory, the learned controller is valid for any arbitrary trajectory. This is because that the whole frequency spectrum is covered throughout the progressive learning.

### 3.6.1 Local Stability Analysis

In our earlier stability analysis, we have found that as long as we can maintain the robustness condition in tuning the system, we can always achieve a positive improvement, $\varepsilon$, in the frequency range where the dynamics range of the actual system has been improved to be the same as the tuned system. From these robustness conditions, we can maximize $\varepsilon$ by selecting the set of excitation

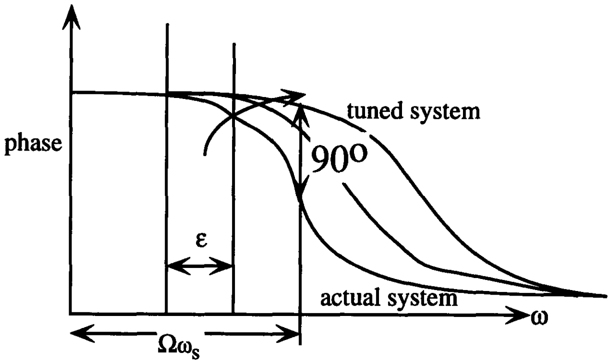


Figure 3.12: Stable improvement bound $\epsilon$

frequencies that satisfy the upper phase limit for each stage of recursive learning operations. As presented earlier, Eqn. (3.52) is a nonlinear matrix that governs stability of the parameter dynamics. This equation is only applicable when the averaging theorem is valid. The eigenvector and eigenvalues of the matrix, $R_{w_f w_\theta}$, govern the local convergence rate with respect to the excitation frequency used. The main reason it is local stability instead of global stability is due to the use of averaging theorem. The lowest eigenvalue, $\lambda_{min}$, of $R_{w_f w_\theta}$, determines the convergence rate at that given trial and the eigenvector determines the direction parameters update themselves. In order to achieve better convergence speed as well as local stability, we need to "reshape" this matrix. By observing the matrix $R_{w_f w_\theta}$, we see that we have two variables since we can freely choose the frequency content and amplitude content of the reference inputs. Since our objective is to maximize the minimum eigenvalue of the matrix, we can rewrite the maximing function as follows:

$$\{\mathbf{R},\ \underline{\omega}\} \quad = \quad arg \max_{\underline{\omega},\mathbf{R}} \left(\lambda_{min}(R_{w_f w_\theta})\right) \qquad (3.112)$$

where $\mathbf{R} = [R_1,\ R_2,\ \cdots,\ R_n]^T$, the magnitude of the all reference inputs and $\underline{\omega} = [\omega_1,\ \omega_2,\ \cdots,\ \omega_n]^T$, the set of excitation frequencies.

### 3.6.2 Requirements for Tracking

We have previously stated that as long as we follow the conditions suggested by the progressive learning theory and persistent excitation requirement, we can improve the system performance. However, the specifications on reference signals are described in the frequency domain while the

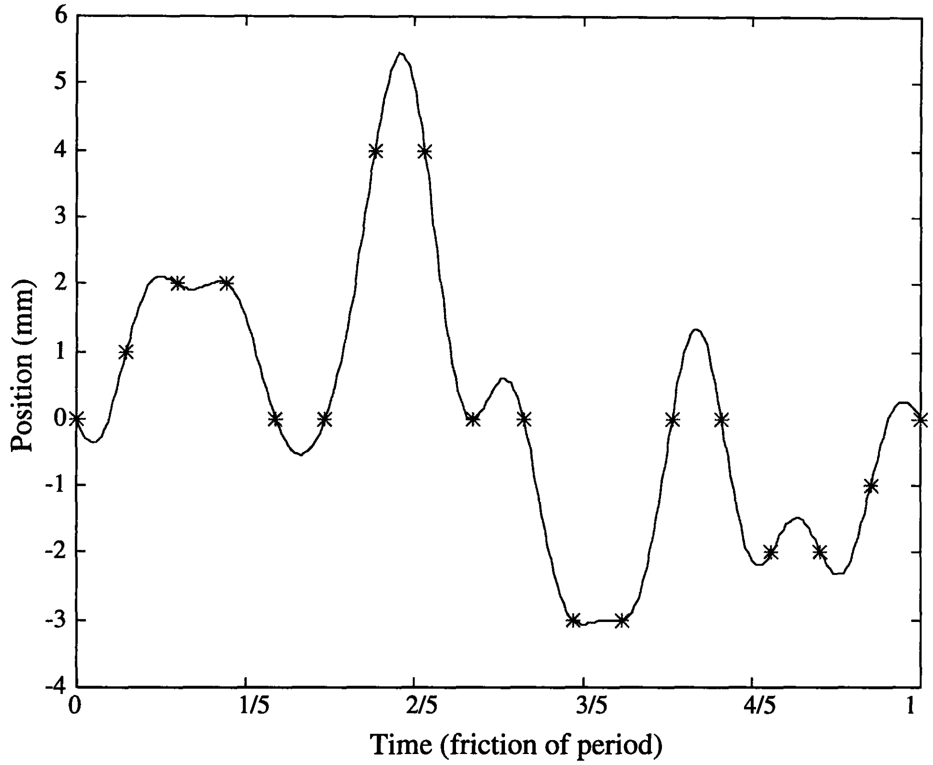


Figure 3.13: A typical desired trajectory used for the given target points

task is described in the time domain. The main purpose of this section is to transform the reference trajectory or any other task from the time domain to the frequency domain that will satisfy the reference input requirement. Usually, the target points we need to meet are prespecified as shown in Figure 3.13. We are usually given a duration, or period,"T", that we need to complete the task. One of the most straightforward approaches relating the time domain to the frequency domain is to perform the Fourier transformation. Another advantage of using Fourier transformation is due to the fact that the sine and cosine functions are cyclic function and are always continuous and differentiable. These characteristics will become self-evident and important, as shown later.

The trajectory formulation states that for M target points, $f(t_1)$,$f(t_2)$,$\cdots$, $f(t_M)$, we need at least $M/2$ distinct frequencies. The time counter, $t_q$, expands from 0 to T where T is time needed to complete the task or the period. For a given frequency set,$\omega_1$, $\omega_2$, $\cdots$, $\omega_m$, the following equation determines the amplitude and the phase shift for each frequency:

$$\sum_{m=1}^{M/2} [A_m \sin(\omega_m t_j) + B_m \cos(\omega_m t_j)] = f(t_j) \text{ where } j = 1, 2, \cdots, M \tag{3.113}$$

where $A_m$ and $B_m$ are the amplitudes. Once we can take the trajectory and decompose it using Eqn. (3.113) in terms of T, we have in fact transformed $f(t)$ from the time domain into the spatial domain. Then, depending on the specified T, i.e., 10s or 10ms, we have further translated $f(t)$ into the frequency domain by fixing the frequencies of which the trajectory function is composed. Using this relationship, we have determined the set of $\omega$ for a given trajectory time profile. By varying T progressively, we can sweep out the whole frequency spectrum band.

The results obtained from Eqn.(3.113) represents the complete trajectory that meets all the target point requirements and the way we interpolate between each target point. However, it represents only the desired output. In order for the system to obtain such desired output we need to find out the desired reference input that can deliver such output characteristics. To do so, we need to inverse the desired model transfer function as shown in Figure 3.11. Thus, the reference input to the system can produce the desired output without the phase lag. All the dynamic system transfer functions can be viewed as a low-pass filter; in some cases, the inverse of the low-pass filter does not exist. However, since we choose to use the sine function which is always differentiable and cyclic, we can guarantee a solution after the inversion since the actual inversion is performed through the use of algebraic manipulation process. Finally, we arrive at the reference input function that satisfies the desired output requirement as shown below:

$$r(t) = \sum_{i=1}^{M} B_{final_q} \sin\left(\omega_q + \varphi_{final_q}\right) \tag{3.114}$$

where $B_{final_q}$ represents the amplitude of each reference input and $\varphi_{final_q}$ represents the phase shift of each reference input after the algebraic manipulation process.

Finally, Eqn.(3.114) after transformation from Eqn.(3.113) represents the reference input synthesis equation from the trajectory tracking point of view. Next, we are going to combine both requirements into one coherent equation that will complete the theoretical background of the trajectory synthesis.

### 3.6.3 Input Design for Stable Adaptation Tracking

After we have presented both sets of requirements in terms of the reference input earlier, we can transform Eqn. (3.52) by adding Eqn. (3.114) into

$$\frac{1}{2}\left[R_{w_f w_\theta}(0) + R_{w_f w_\theta}(0)^T\right] = \frac{\alpha}{k_{t_q}} \sum_{i=1}^{N} R_i R_i^H |F_{t_q}(j\omega_i)|^2 \mathcal{R}e\{frac\Phi_{t_q}(j\omega)\Phi_{Fp_q}(j\omega)\} H_{w_m r}(j\omega_i) H_{w_m r}^H(j\omega_i) \tag{3.115}$$

where

$$R_i = \left(B_{final_q} \exp^{j\varphi_{final_q}}\right) \tag{3.116}$$

and the final form of the trajectory synthesis after transformation is:

$$\underline{\omega} = arg \max_{\underline{\omega}}(\lambda_{min}(R_{w_f w_\theta}) + \gamma(\sum_{i=1}^{M/2} B_{final_q} \exp^{j(\omega_q t + \varphi_{final_q})} - W_m^{-1} f(t))) \tag{3.117}$$

where $\gamma$ is the Lagrange multiplier. As described earlier, originally, we can arbitrarily choose the amplitude and frequency for the reference input in terms of the stability analysis. Once we put the geometric constraint on it as shown in Eqn.(3.117), we have only one freedom in arbitrary selecting the frequency but not the amplitude. Based on this equation, we have the freedom to choose any distinct frequency to make up the set of the frequencies that will satisfy the stable tracking. This means that the desired trajectory can go through all the targets while it has a known frequency contend to ensure stable, faster convergent adaptive tuning. The use of the Lagrange multiplier is to connect the optimal equation to its constraint.

Now, we have completed described the theoretical results for the model augmentation and trajectory synthesis. Next, we are going to describe the progressive learning algorithm that is based on these theoretical results.

## Chapter 4

# Implementation Procedure

## 4.1 Introduction

In this chapter, we will mainly focus on how we actually apply the theoretical results obtained earlier to the actual application. We will first describe the summary of the implementation procedure before we describe each step in a greater detail. In terms of the model augmentation, Sec. 4.2 focuses on all the necessary steps we need to perform or justify about the application we have in mind. Sec. 4.3 focuses mainly on when is the best switching strategy that we can use while still maintain the stability and fast convergence of the overall control parameters. Sec. 4.3.2 focuses mainly on how to achieve fast convergence and stable parameter space expansion as we switch from one stage to the next. Then we will discuss the termination procedure. To demonstrate the proposed algorithm, we will use a simulation as shown in Sec.4.5to conclude this chapter.

## 4.2 Summary of the Implementation Procedure

In the earlier sections, we have introduced the theory of our research. In this section, we will describe the general procedure in actual applying the proposed algorithm to actual systems step by step and this is the general procedure we used in obtaining the results shown later.

In general, there are three main steps that we have to go through for implementing the progressive learning algorithm. They are listed below:

1. Experimental setup

2. Experiment trials

3. Post-experimental process

Though they are very broad; they covered all the aspects. Next we will discuss each of them individually.

### 4.2.1 Experimental Setup

In the experimental setup section, we need to obtain some preliminary information about the system that we want to tune. Based on this preliminary information, our main objective is to first evaluate the effectiveness or applicability of the progressive learning to the system of interest. Second, based on the user-defined performance specification, we need to determine how we actually prepare each stage for running actual experiments.

In general, in order to know if we can effectively use the progressive learning algorithm is to follow the steps listed below:

1. Obtain a open-loop bode plot of the system intended to control
2. Obtain a noise frequency spectrum to determine the bound on noise, $b_v$
3. Determine a frequency range of interest where the minimal tracking speed is essential
4. Determine the lowest order of estimate of the plant transfer function that will fit the frequency range of interest
5. Determine the maximum phase allowable for the discrepancy between the estimate transfer function and the actual bode plot. This should be less than $\frac{\pi}{2}$
6. Determine the estimated switching frequency, $\omega_s$, based on the residual dynamics.
7. Determine the observer polynomial coefficients. They should reflect the dynamics bandwidth of the subsystem and reject the noise penetration.

The main reason for obtaining the spectrum information about the system, is due to the fact that all our analytical results are based on the frequency domain analysis. Furthermore, due to the fact that results from our stability analysis and robustness analysis require us to know only the phase portion of the spectrum information, we need to obtain at least some rough estimation of the phase information if not the spectrum information. Once we have obtained the phase information, we can start to evaluate the system if we in fact can apply the progressive learning algorithm.

Though the concept of progressive learning algorithm seems nature; nevertheless, there are several limitations that will prevent us to use the newly developed extended progressive learning algorithm. First limitation of the progressive learning algorithm is to apply it to a non-minimal phase system. The main reason is that when we derive the stability and robustness bound, in order to achieve a conservative estimation, we use only the phase information. Due to the fact that the interpretation of the phase plot for non-minimal phase systems has different meaning as compared to the phase plot of the minimal phase systems, we can not yet guarantee the similar results can be completely extended to the non-minimal phase system at this time. Thus, currently, we have to restrict our applications to be minimal phase endpoint feedback systems. Next, in order to apply the extended progressive learning algorithm where we have added the model augmentation and trajectory synthesis parts to it, the system spectrum has to display clear separation among various modes. Each individual mode can be composed of any system order. However, on the other side, if no clear separation can be observed, the effectiveness of model augmentation may be limited. It may not be efficient and effective to apply the model augmentation to the system controller.

In order to determine if the system has "clear separable" modes, or stages, we offer the follow suggestion. Since we are mainly concerned with only the phase information, we can use the small angle approximation guideline to determine the separation region. Once the user defined what is allowable phase angle deviation allowed from the higher mode, $\Gamma_{max}$ which should be less than $20^o$ due to the assumption of the small angle approximation, we can partake the frequency band into various pieces and estimate the order of the system within each region. Consequently, we will determine the estimated $\omega_{s_q}$ which is the maximum frequency where the higher order dynamics phase contribution is still within the $\Gamma_{max}$ for each stage and the order of the subsystem in each of the stage. To give an theoretical example, Appendix D has derived the required separation ratio between two second order sub-systems has to be in order to be considered "separable." To keep inline with the notation we used, each separable region is considered to be one stage. Then, based on the estimated order of the subsystem with the estimated system model parameters, we can also determine the correct maximum excitation frequency for us to start tuning the system. This maximum excitation frequency will in fact determine the "allowable speed" that the system can run initially. Appendix D also derived the required excitation ratio between the maximum excitation frequency and the nature frequency of the first second-order system for the same system with two second order sub-systems. As a side note for determining the order of the subsystem, we need to be very conservative. Though in the deterministic robust control where the controller parameters are fixed irrespectively of the true system behavior, over-estimating or under estimating system's behavior may not cause instability; however, in the adaptive control field, parameters are estimated based on the assumed model frame with true data that behave non-linearly, thus system dynamics becomes depending on the outcome of the parameter adaptation dynamics. By over-estimating the system's complexity, we require more parameters to be tuned thus stability and high convergence rate may be hard to be achieved and impossible to be implemented. On the other hand, by under-estimating the system behavior, the internal unmodeled dynamics will cause assumed modeled-framed parameters to change naturally irrespective of the system dynamics. Thus, it is important as well as a practical reason to assume a simplest model or the lowest order assumed system model that will describe the system behavior in the region that we are interested in or the lowest excitation frequency range that will still produces tolerated performance.

After we determined the maximum allowable excitation frequency ratio, we can run the trajectory synthesis algorithm as presented earlier to determine the correct amplitude and phase shift in order to meet the geometric requirement of the given task. The main reason for keeping the correct excitation ratio and separation ratio is that once we have started tuning, we can achieve stable tuning without exciting the undesirable higher order mode and correctly initialize the control parameters need to be tuned. Then, we also need to design the observer dynamics to have a relative dynamics region so that it can have a good cut-off frequency to eliminate the noise penetration to the adaptation dynamics as mention earlier.

Though in theory, determination of $b_v$, the bound on noise, is trivial; however, it is much harder to determine in the real application since $M$, the number of the noise spectrum or the frequency range, is harder to estimate. In general, $M$, the number of the noise spectrum is infinite; however, by carefully selecting the observer characteristics polynomial, $\Delta_q$, we can then limit the noise frequency bandwidth. The physical interpretation of the $\Delta_q$ is a low pass filter. By selecting the range of interested frequency, we can then have the correct cut-off frequency to limit the penetration of the noise to the regressor signals. Thus, the effective frequency range of the noise spectrum will have the same frequency bandwidth as the tuned closed-loop frequency bandwidth. Due to the physical limitation on the bandwidth of the servo amplifier together with the noise spectrum, there exist a frequency $\omega_v$ such that the proper S/N ratio can not be achieved. Thus, this provides the marginal stability limit that the tuning can expand the system's bandwidth. As long as we can achieve the proposed bound, then the system is robust from the noise. In conducting the trajectory synthesis, another warning in terms of the difference between the theory and the implementation is the status of persistent excitation. The theory of the persistent excitation only provides sufficient condition but does not guarantee the signal-independence for the $H_{wr}(j\omega_i)H_{wr}^H(j\omega_i)$. Thus, unless we excite the system with the signals that have enough energy content in the frequency range where the plant model is of sufficient order to represent the actual plant, the "precise" persistent excitation is questionable. In the case when the system is not be persistently excited by the stable reference signal, since due to the fact that if the noise is white noise, the overall dynamics will still achieve persistent excitation condition but just not on the reference signal. Therefor the robustness is questionable since $R_r^{-1}$ is not defined. Thus, as a practical issue, it is important to start with a lower order system model which requires lower number of parameters to be estimated and to excite the system with more frequency contents than the minimum requirements to ensure the parameter convergence.

After we have completed all the steps described above, we are ready to conduct actual experiments to tune the system.

### 4.2.2 Experimental Trials

The main objective of this section is to faithfully execute the experiments with correct model order structure and tracking speed in mind. Within each stage, we need to repeat various number of runs to completely tune the control parameters. Each individual run of a complete task is considered to be one trial. The length of a trial is fixed by the excitation frequency range used. The number of trials required, however, is determined by the actual experimental results. In this stage, our main task is to collect as much information as we can that will help us determine various actions to be taken after completing one trial. For most of our applications that require endpoint feedback, our objective here at this section is to collect the output sensor data, process the date, and perform online adaptive control according to the gradient descent rule as suggested earlier. Then based on the output error, the observer dynamics estimates the missing state and the adaptive controller estimates the "correct" control parameter values in order to achieve full-state feedback and to mimics the desired model output.

Based on these input and output information between the actual experimental data and the desired output, we need to determine the appropriate actions to take. Detailed descriptions of how we determine and how we execute these actions is described in the next section.

### 4.2.3 Post-Experimental Process

Similar to the previous two sections, we need to again go through several steps as described below:

1. Determining the phase angle at each of the excitation frequencies.
2. Comparing with the desired model to determine the $\varepsilon$ for each trial.
   A. Increasing the excitation frequency range if $\varepsilon$ continues to be positive and/or increasing.
   B. Start the augmentation procedure if the progress of $varepsilon$ starts to slow down.
3. Compare the frequency range, hardware limit, and noise frequency limit to determine the termination of the experiments

From the above, we see the general overview in terms of list of steps that we need to follow for the algorithm. Next, we will discuss the heart of the algorithm which is the decision making process for augmentation. The "adequate" decision process includes on-line validation of the switching frequency and the determination of the correct switching scheme. Then, we will discuss under what condition we will terminate the experiments.

## 4.3 Determining the Model Augmentation Scheme

In the previous section, we have found that the system will remain stable even though we start to excite the unmodeled dynamics as long as Eqn.(3.87) is followed. Our next goal is to propose an augmentation scheme to expand the original system to include the unmodeled dynamics. The main purpose of this section is to propose the model augmentation scheme to be used based on the stability analysis and robust analysis presented earlier.

### 4.3.1 Stable Tuning Range and Switching Frequency

In this section, we will determine the switching frequency as well as the stable tuning range. As stated earlier in our robustness analysis , as long as we excite the system within the region where the phase contribution of the unmodeled dynamics is less than $\Gamma_{max}$, we can achieve stable adaptive control despite the pre-assumed modeled dynamics. Thus, based on the stability condition derived in the robustness analysis, $\omega_s$ that we determined in the setup section is defined as the switching frequency. As we start exciting at a frequency higher than $\omega_s$, unmodeled dynamics will cause the tuned parameters to drift and the stability range to decrease as demonstrated in the robustness simulation. Thus, our first task is to determine the location of the switching frequency, $\omega_s$. Although determination of the exact location of $\omega_s$ may not be possible, we can estimate it and validate it later on line. In order to estimate the location of $\omega_s$, we have to examine the unmodeled dynamics. Once we have determined the estimated order of the system, the $G_r(j\omega)$ can then be determined as the difference between the actual system and the assumed plant model. As stated earlier, phase of the $F_{r_{t_q}}$ is shown below:

$$F_{r_{t_q}} = |\angle\{F_{t_q}(j\omega)\} - \angle\{G_{m_q}(j\omega)\}| \tag{4.1}$$

Then based on the relation between the two and the phase contribution from the high order dynamics at low frequency range is minimal, $F_{r_t}$ can be further estimated as

$$F_{r_t} = G_r. \tag{4.2}$$

Then, $\omega_s$ can be easily estimated as below:

$$|\angle\{G_{r_q}(j\omega_s)\}| = \Gamma_{max} \tag{4.3}$$

In the setup, though based on the assumption of the modeled plant and the chosen reference model, we estimate $\omega_s$ ahead of time; however, we need to validate the estimation of $\omega_s$ on-line in the actual implementation. Before doing so, we need to estimate the phase angle at each excitation on line as well.

The determination of $\angle\{F_{p_q}\}$ can be easily obtained through the use of simple sample signal data analysis. From the trajectory synthesis, the reference signal $r(t)$ can be decomposed into spectrum signal at frequency $\omega_i$ as $r_i(t) = \alpha_i \sin(\omega_i t)$ and its output $y_i(t) = \alpha_i \mid G(\omega_i) \mid sin(\omega_i t + \varphi_i)$ where $\alpha$ is the excitation amplitude, $G_i(\omega)$ is the output transfer function gain, and $\varphi_i$ is the $\angle\{F_{p_q}\}$. By using the averaging technique again, we can obtain the output averaging with its crossing correlation with both the sin and cos functions as

$$Y_{c_i} = \frac{1}{T}\int_{t=0}^{T} y_i(t)cos(\omega_i t)dt = \frac{\alpha}{2} \mid G(\omega_i) \mid sin(\varphi_i) \tag{4.4}$$

and

$$Y_{s_i} = \frac{1}{T}\int_{t=0}^{T} y_i(t)sin(\omega_i t)dt = \frac{\alpha}{2} \mid G(\omega_i) \mid cos(\varphi_i) \tag{4.5}$$

Then $\mid G(\omega_i) \mid$ can be easily determined as below:

$$\mid G(\omega_i) \mid = \frac{2}{\alpha}\sqrt{Y_{c_i}^2 + Y_{s_i}^2} \tag{4.6}$$

and the phase shift $\angle\{F_{p_q}\}$ can be then determined as

$$\angle\{F_{p_q}\} = \tan^{-1}\left(\frac{Y_{c_i}}{Y_{s_i}}\right) \tag{4.7}$$

Once the phase of the closed-loop transfer function has been determined at each of the reference input frequency, we can evaluate $\varepsilon$, the improvement in tuning for each trial. As pointed out earlier in the robustness analysis, we can not determine $F_{t_q}$ ahead of time since the $\theta_t$, the desired tuned control parameters, can not be obtained advance. Thus, besides limiting the phase contribution to $\frac{\pi}{2}$ less than the allow phase contribution, $\Gamma_{max}$, we need to validate $\omega_s$ that was determined based on the open-loop residual transfer function, $G_{r_q}$ in advance to maintain the stability. As pointed out in the stability analysis, according to expansion theorem presented earlier, as long as we maintain these conservative expansions, we will always achieve a positive $\varepsilon$. The interpretation of Eqn.(3.94) is that due to the uncertainty, the maximum allowable phase difference between the reference model and the actual closed-loop transfer function at each excitation frequency in order to achieve a positive expansion $\varepsilon$ is $\frac{\pi}{2} - |\Gamma_{r_q}|$. As we start to excite the unmodeled dynamics, the residual dynamics will cause the parameter to drift in order to match the reference model at these frequency ranges. One of the direct results of it is that we will observe parameters start to vary quite significantly and the other direct result is to observe the magnitude for improvement on $\varepsilon$ to decrease. At the point of $\omega_H$, we will in fact either obtain a negative or zero improvement since the stability of the parameter dynamics has become questionable. Thus, one of the natural methods directly resulting from this is to put the maximum parameter variation bound on a given tuning session [Narendra and Annaswamy, 1987]. In our case, we will rely on the $\varepsilon$ obtained to ensure the validation of $\omega_s$ on-line. According to the expansion theorem, if we observed the rate of change in $\varepsilon$ to slow down, we have in fact started exciting the unmodeled dynamics. Thus, we can either stop expansion of excitation frequency or start the augmentation process.

Once the $\omega_s$ is determined, the stable expansion frequency range, $\Omega_s$ can be determined as follows:

$$\Omega_{\omega_s} \in \{\omega || \angle\{G_{m_q}(j\omega)\} - \angle\{F_{p_q}(j\omega)\}| < \frac{\pi}{2} - \Gamma_{max} \text{ and } 0 < \omega < \omega_s\} \tag{4.8}$$

Then, our next step is to propose a stable augmentation scheme to determine how we expand from one stage to another in terms of the control parameter space.

### 4.3.2 Expansion of Parameter Space

In the previous sections, we have presented the stability analysis in terms of unmodeled dynamics. In this section, we will present a way of expanding the parameter space when we perform the model augmentation.

Earlier, we have stated when to augment the controller complexity. Since at $q$th stage, the parameter space, $\theta_q$ is $\in \mathbb{R}^{2(q)}$ while at $(q+1)$th stage, the parameter space is $\theta_{q+1}$ is $\in \mathbb{R}^{2(q+1)}$, we need to describe the way of initializing the newly added parameters. From some preliminary experimental results we have obtained, if we do augment both the controller complexity as well as reference model at the same time, the initial parameters values are quite oscillatory. This is mainly due to the fact that the newly added parameter space has not been correctly initialized. Thus, this has led us to propose a new augmentation scheme in terms of expansion of parameter space.

Based on the MRAC formulation, the closed-loop transfer function for the $q$th stage is

$$F_{p_q} = \frac{K_p k_{c_q} Z_p \Delta_q}{(\Delta_q - C_q)R_p - k_p Z_p D_q} \tag{4.9}$$

with $G_{m_q}$ as the reference plant while the closed-loop transfer function for the $(q+1)$th stage is at

$$F_{p_{(q+1)}} = \frac{K_p k_{c_{(q+1)}} Z_p \Delta_{(q+1)}}{(\Delta_{(q+1)} - C_{(q+1)})R_p - k_p Z_p D_{(q+1)}} \tag{4.10}$$

with $G_{m_{q+1}}$ as the reference plant model. In order to retain the stability from $q$th stage to $q+1$th stage, we need to carefully initialize the newly added parameters. Instead of directly augmenting from $q$th stage to $q+1$th stage, we propose a two-step augmentation scheme. During the transition stage, $tf_q$, only the observer dynamics has been expanded as well as the controller parameter space instead of expanding the excitation frequency range. The purpose is to achieve stable initialization of the newly added parameter values since the effect of the higher order dynamics is low at this point. Thus, the prior knowledge of the system is "preserved" and new knowledge is learned through the augmenting the observer dynamics.

Once the improvement in $\varepsilon$ is restored, we will perform the second step in augmentation which is to augment the reference input model to the new stage.

Next, we will conclude this algorithm by describing how we actually terminate the tuning experiments.

## 4.4 Termination Conditions for Progressive Learning

Besides the physical reasons for termination of experiment such as performance requirements, servo-bandwidth, noise spectrum, and etc., there are other limitations for us to further expand the control bandwidth of the system. As we push for higher performance or a wider-bandwidth, we have to extend the controller complexity or increase the stage number. However, in the actual implementation issue, we need to study the feasibility of augmentation. In order to augment the expanding of the frequency spectrum, we need to consider that the noise spectrum will not vary or diminish plus many physical limitations such as servo bandwidth and rated output, sensor bandwidth, calculation time-cycle for the hardware, and etc. Based on the projected next stage controller gains with the same $\Gamma_{max}$, a new set of $\omega_s$ and $\omega_H$ can be calculated. Then we need to compared to the $\omega_v$. If they are both smaller than $\omega_v$, then augmentation is achievable; otherwise, unless there is a hardware change, we will not be able to make any improvement. We should also terminate the experiments by observing $\varepsilon$ specially after we have augmented. After we have completed an experiment, we should validate the experimental results by running the same test but with different task or different trajectory to ensure the quality of the learning.

To summarize all the procedure, a simulation is done next before we actually show the actual implementation results.

## 4.5 Simulation

In this section, a simulation is shown here to demonstrate the use of the algorithm, in particular, the *model augmentation* and the *trajectory synthesis* as described earlier. A plant with relative degree of three is used as an example. We also use it to demonstrate 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

$$W_p(s) = \frac{Z_p(s)}{R_p(s)} = \frac{4(s+0.6)}{(s^2+0.4s+0.16)(s^2+0.8s+4)} \tag{4.11}$$

$$W_m(s) = \frac{Z_m(s)}{R_m(s)} = \frac{5(s+1)}{(s^2+1.04s+0.54)(s^2+2.96s+10.83)} \tag{4.12}$$

where $W_p(s) = W_{p_1} \times W_{p_2}$ and $W_{p_1}$ and $W_{p_2}$ respectively are

$$W_{p_1} = \frac{3.75(s+0.6)}{(s^2+0.4s+0.16)}, \quad W_{p_2} = \frac{1.0667}{(s^2+0.8s+4)} \tag{4.13}$$

and at the same time, where $W_m(s) = W_{m_1} \times W_{m_2}$ and $W_{m_1}$ and $W_{m_2}$ respectively are

$$W_{m_1} = \frac{2.02(s+1)}{(s^2+1.04s+0.54)}, \quad W_{m_2} = \frac{2.475}{(s^2+2.96s+10.83)} \tag{4.14}$$

respectively.

The fixed control parameters for each stage are

$$\Lambda_1 = [-1], \quad l_1 = [1], \quad \Lambda_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -4 & -8 & -5 \end{bmatrix}, \quad l = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.15}$$

or

$$\Delta_1(s) = s+1, \quad \Delta_2(s) = (s^3+5s^2+8s+4) \tag{4.16}$$

By selecting $\Gamma_{max}$ to be $10^o$ from the use of the small angle approximation as suggested in the algorithm, we can then estimate $\omega_s$ to be $0.7r/s$ and $\omega_H$ to be $1.1r/s$. Then eight control parameters $k$, $\theta_1 = [\theta_{11},\ \theta_{12},\ \theta_{13}]^T$, $\theta_2 = [\theta_{21},\ \theta_{22},\ \theta_{23}]^T$ are adjusted using the gradient descent method. The first stage tuned control parameters, $\theta_{t_1}$, are $[k_{t_1}\quad \theta_{t_1}\quad \theta_{t_0}\quad \theta_{t_2}] = [0.5061\quad -0.2774\quad -0.1280\quad -0.2478]$. As we switch to augment the second stage observer polynomial, we expand the parameter space to its tuned values as $[k_{t_1}] = 0.84184, \theta_{t_1} = [-0.008\quad -1.4633\quad -0.2389]$, $\theta_{t_0} = -0.7252, \theta_{t_2} = [1.8103\quad 2.6827\quad 2.7732]$. Finally, after augmenting the second reference model dynamics, the tuned control parameters will be $[k_{t_2}] = 1.25, \theta_{t_1} = [-0.91\quad -17.9\quad -2.4], \theta_{t_0} = -7.07, \theta_{t_2} = [25.93\quad 48.38\quad 34.12]$.

*B. Initialization and Instability Mechanism*

The feedforward gain $k$ was initialized to 0.5 and the other control parameters were initialized to zero. Figure 4.1 shows the phase angle curves of both the initial plant closed-loop transfer function as well as the final reference model.

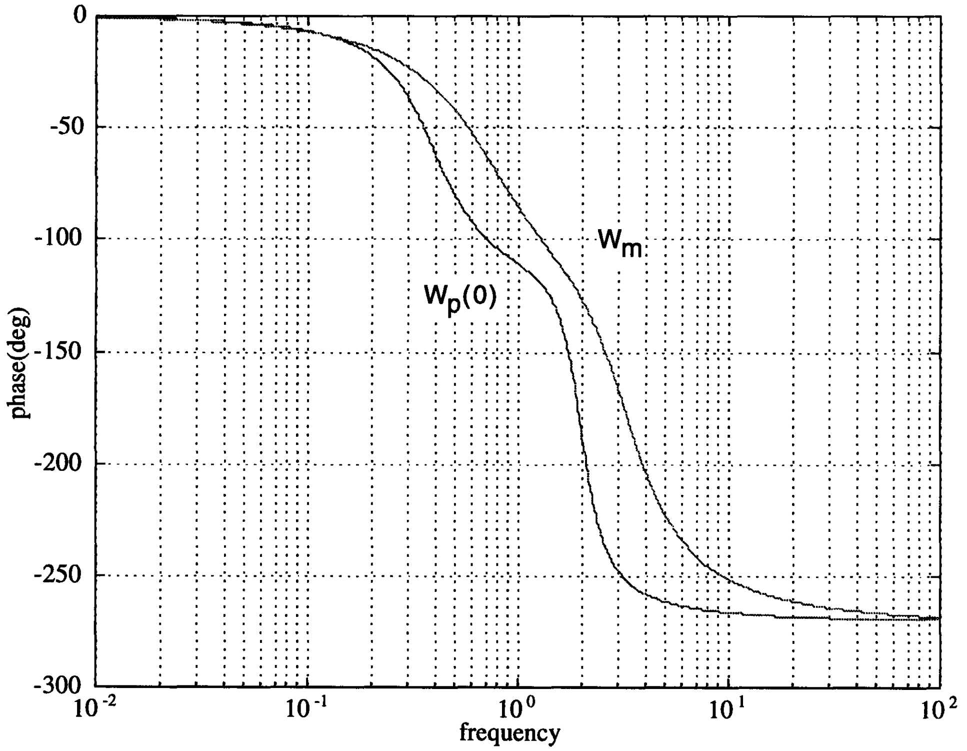


Figure 4.1: Initial and desired phase plot

### 4.5.1 Simulation Setup

According to the model augmentation presented earlier, we have four scenarios to compare. As described earlier, the system is a fourth order system. Our first scenario is the case when the controller is designed based on the reduced order system or a second order system. The second scenario is the full-order controller when the controller was designed and tuned based on the full-order system or a 4th order system. The third scenario is the case that we start like the first scenario with a second order controller, and then upgrade to the fourth order controller directly. The last scenario is similar to the third scenario except that we add the intermediate stage to have the controller upgrade the observer dynamics before we augment to the higher order controller. The summary of all four scenario setups are shown in Table 4.1.

Table 4.1: Simulation Setup in terms of the order of the controller

| Scenario No. | 2nd order | 2nd order + higher observer | 4th order |
|---|---|---|---|
| 1 | all | n/a | n/a |
| 2 | n/a | n/a | all |
| 3 | 1 | n/a | 2 |
| 4 | 1 | 2 | 3 |

The geometric trajectory for the system to follow is shown in Figure 4.2. The trajectory used for simulation only covered $x - axis$ trajectory data. Based on the *trajectory synthesis*, the frequency is selected to be equally spaced between the maximum frequency and the zero frequency or the D.C. frequency. The maximum excitation frequency for each trial and each scenario is plotted in Figure 4.3. The trend of the graph suggests how the bandwidth has been expanded for each case in its 10-trial history.

Next we are going to show the simulation results.

### 4.5.2 Simulation Results and Discussions

Simulations based on the these four scenarios are then performed. Figure 4.4-Figure 4.7 show the parameter history plot, the error plot, the phase plot, and the phase lag plot for the 4th scenario which was based on the complete implementations of the model augmentation. The aug-

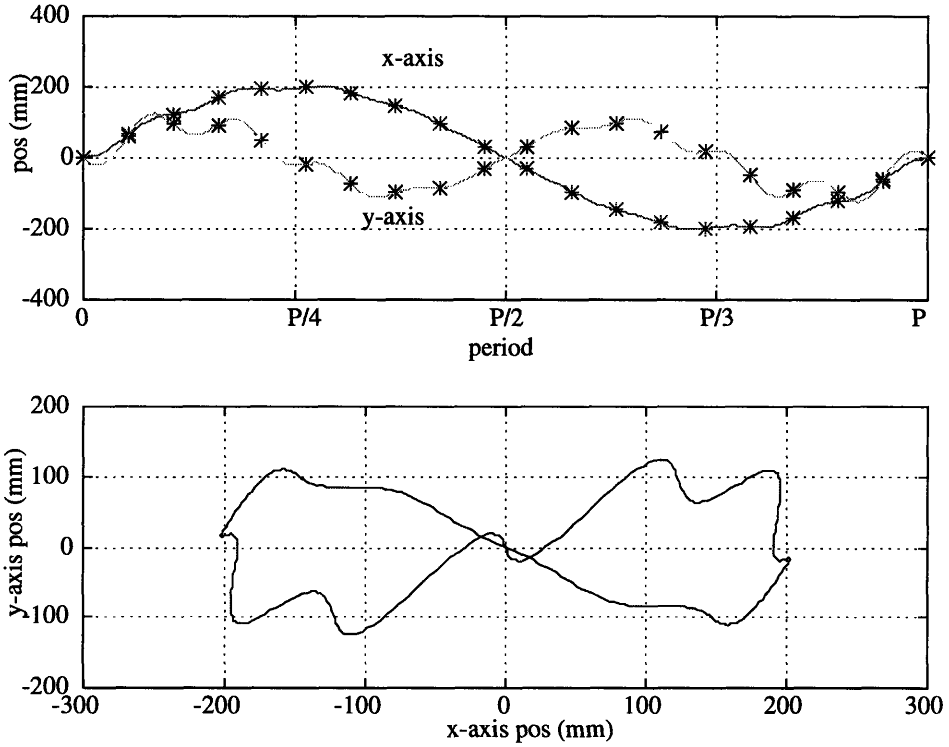


Figure 4.2: Trajectory profile for the simulation

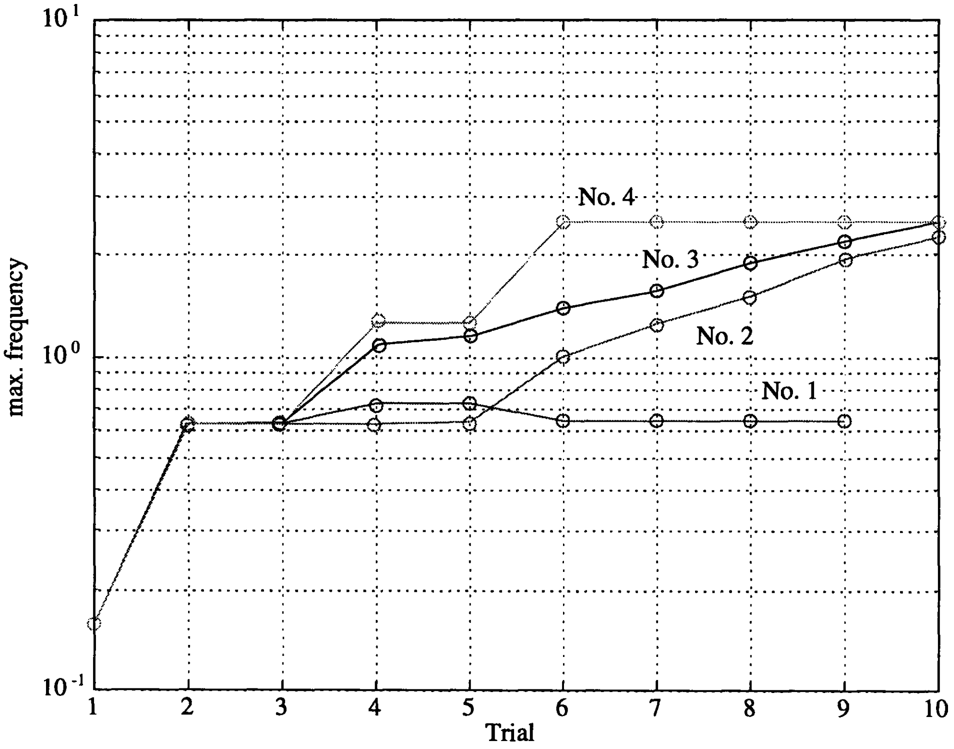


Figure 4.3: Maximum frequency of each trials for varies system dynamics

mentation starts when $\omega_s$ was determined to be around $6r/s$ on-line. Thus, the expansion of the observer dynamics starts after Trial No. 2. Then, to demonstrate the robustness of the system, the second stage reference model is not added until Trial No. 4.

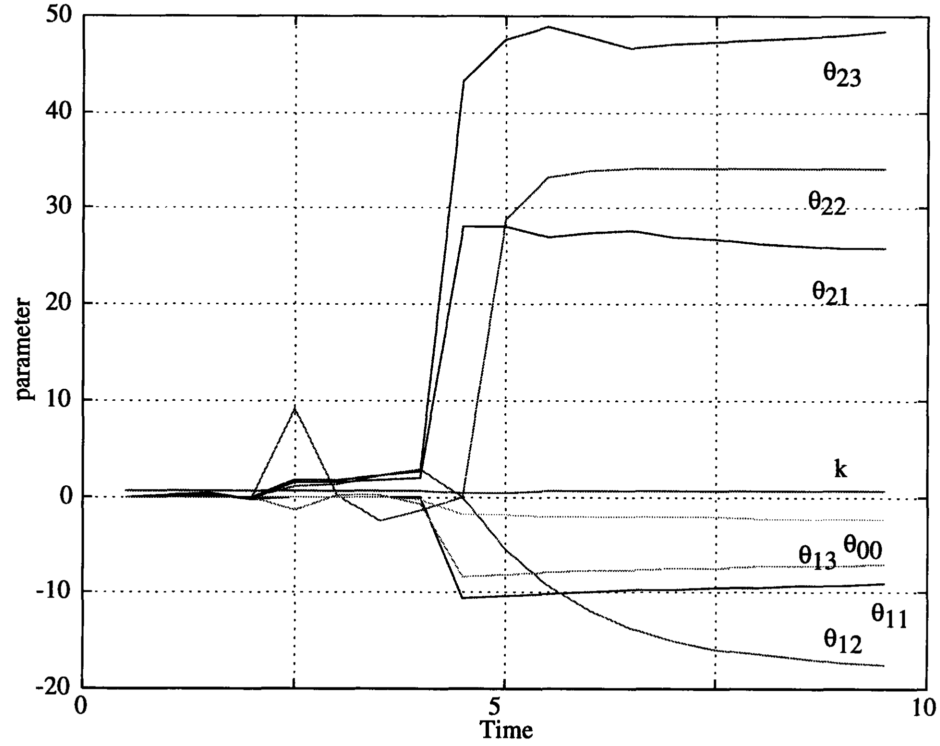


Figure 4.4: Control parameter plot based on proposed complete algorithm

Figure 4.8 and Figure 4.9 show the parameter history plot, the error plot, the phase plot, and the phase lag plot for the 2nd scenario which was based on only the full-order controller tuning.

Figure 4.10 and Figure 4.11 show the parameter history plot, the error plot, the phase plot, and the phase lag plot for the 3rd scenario which is similar to the full scaled implementation of the model augmentation, except not upgrading the observer ahead of the time.

Finally, Figure 4.12 shows the error plot, and the phase lag plot for the Scenario No. 1, which was based on the reduced order controller only. As expected, the performance can be maintained at low frequency and parameter can be tuned to the projected lower order controller parameter setting; however, it can not be used to track at higher speed or higher frequency bandwidth.

Clearly, we see that the fourth scenario, which was based on complete implementation of the augmentation algorithm, and the third scenario, which augments to the new stage all at once, show

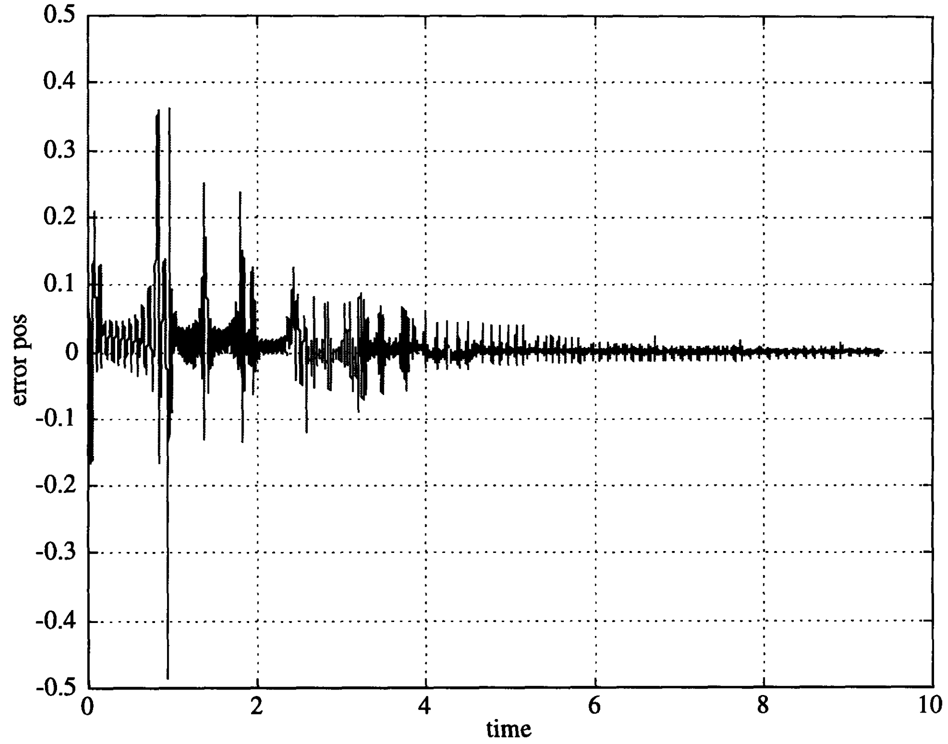


Figure 4.5: Error plot based on proposed complete algorithm

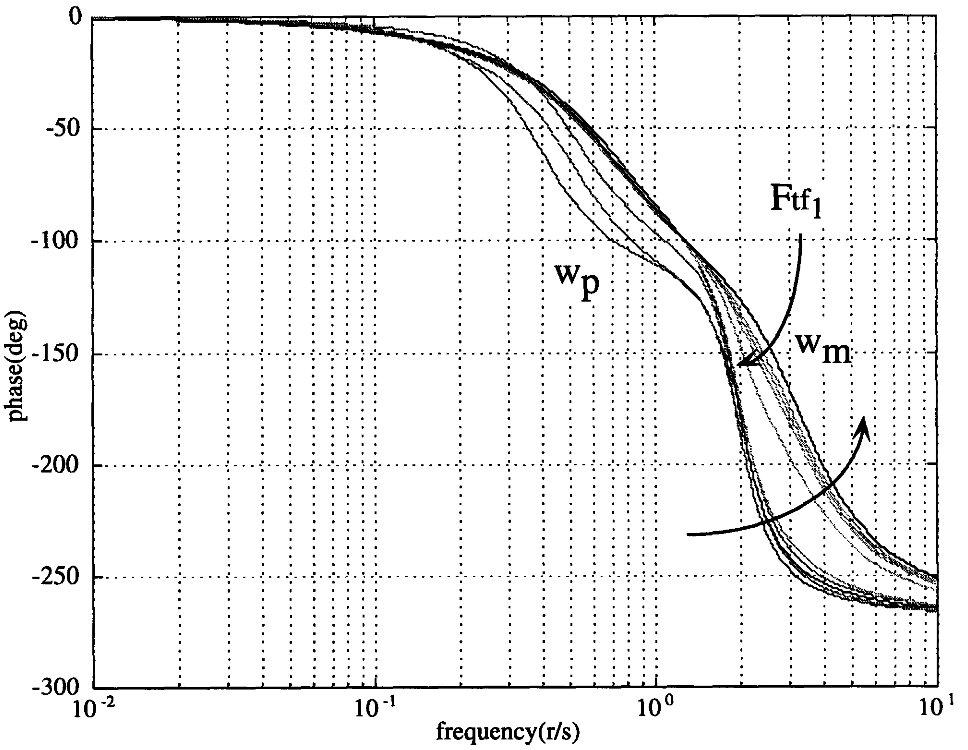


Figure 4.6: Phase plot based on proposed complete algorithm

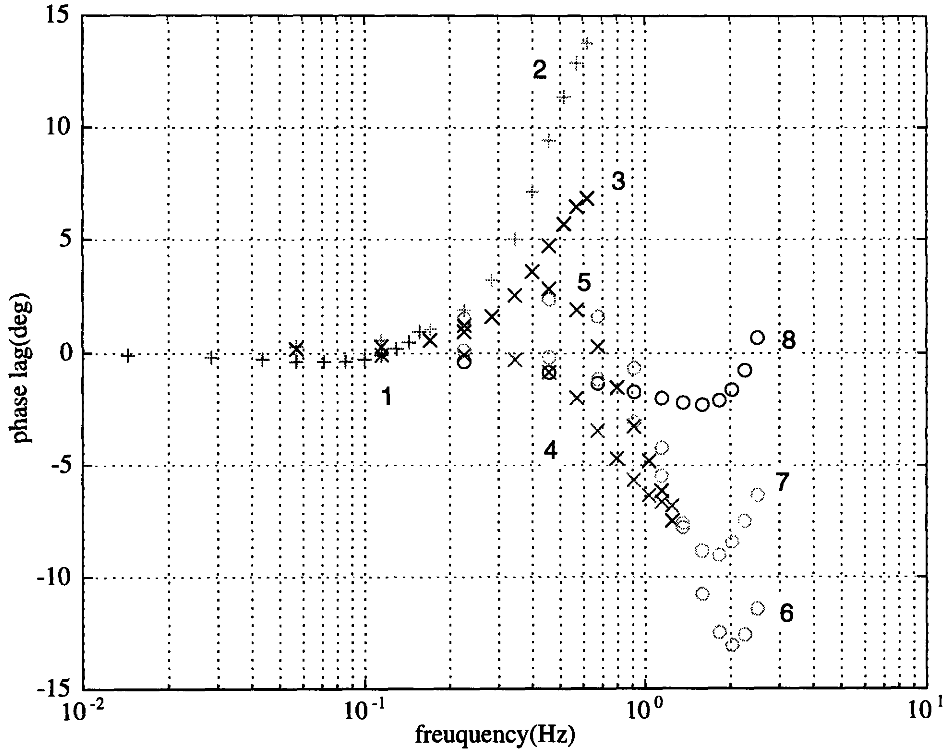
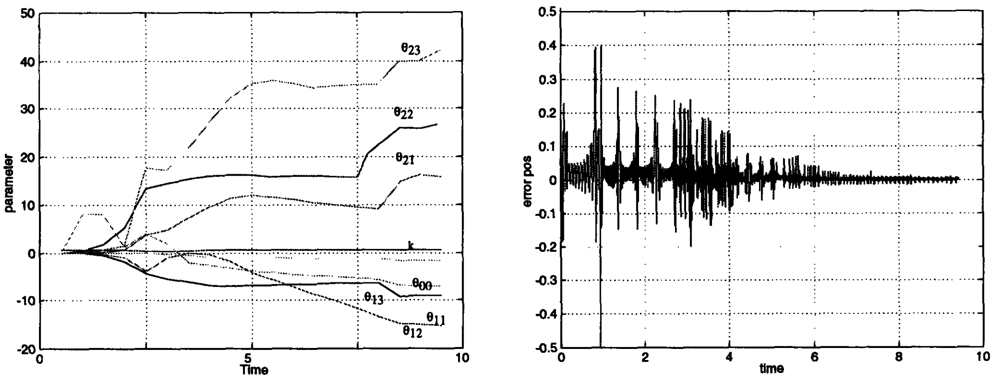


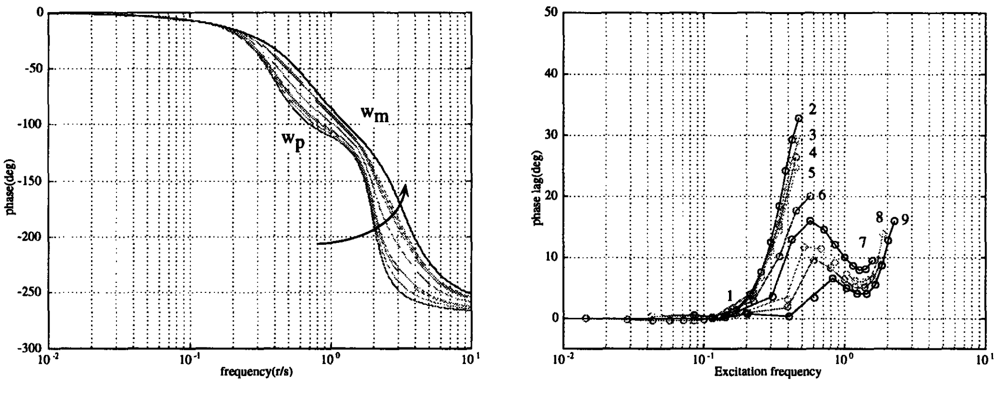
Figure 4.7: Phase lag plot based on proposed complete algorithm



(a) Control parameter

(b) Tracking error

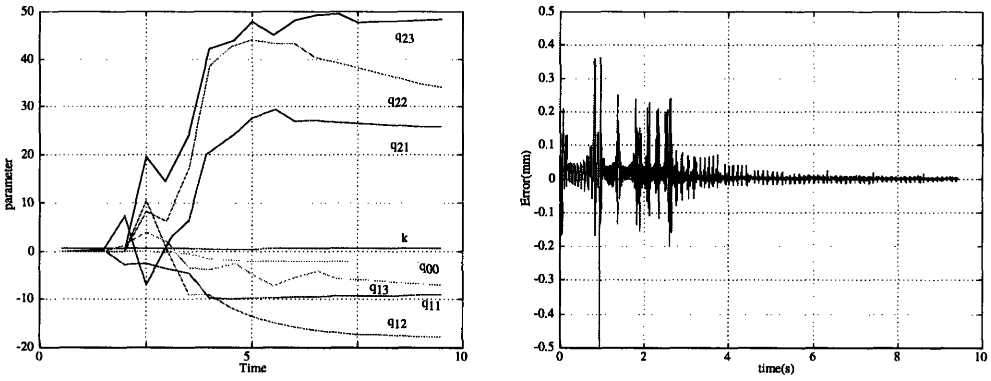
Figure 4.8: Results based on full-scaled controller implementation (error and parameter plot)



(a) Phase plot

(b) Phase lag plot

Figure 4.9: Results based on full-scaled controller implementation (phase angle plot)

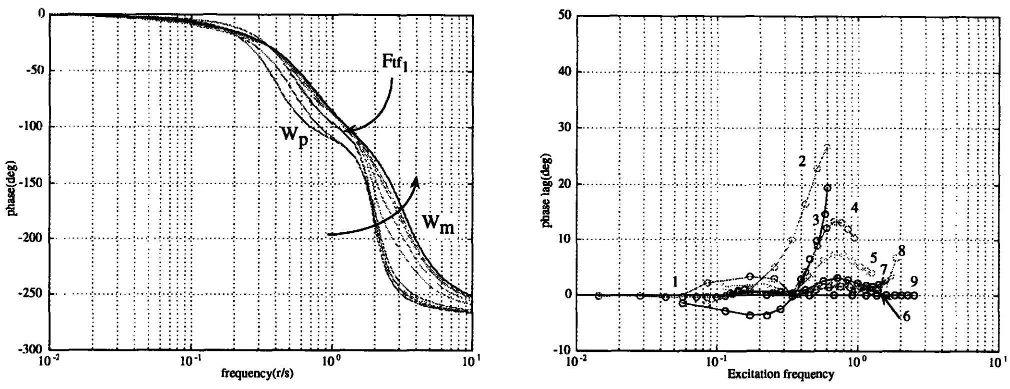


(a) Control parameter

(b) Tracking error

Figure 4.10: Results based on model augmentation without augmenting the observer in advance (parameter and error plot)

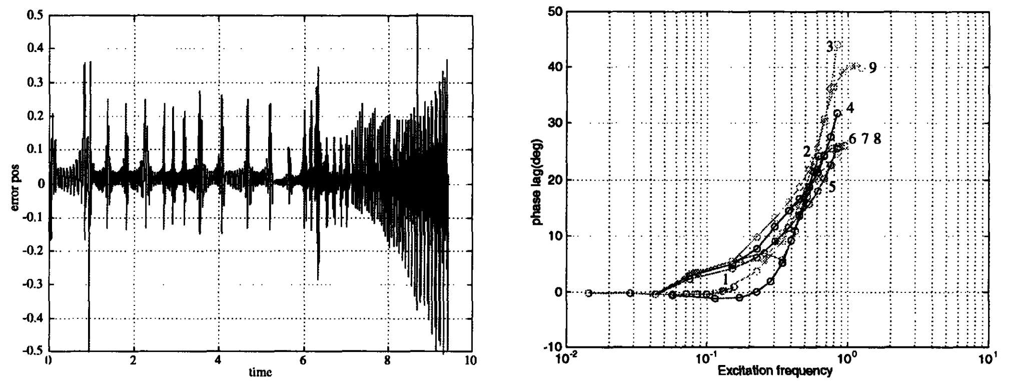
far superior results than the rest two scenarios. It is clearly to see that the parameters converge faster and most important of all, the performances during the all of the trials have better tracking or lower tracking error, which was the primary concern for the research. The reason for the complete implementation of the model augmentation to have slightly better results than the third scenario is due to the fact the observer dynamics is expanded earlier before the higher order controller has fully been excited. The complete implementation uses the higher-order observer dynamics implemented to observe the higher order unmodeled dynamics by initializing the higher order control parameters, namely, $\theta_{12}$, $\theta_{13}$, $\theta_{22}$, and $\theta_{23}$, ahead of time, while the third scenario does not. Though in both cases



(a) Phase plot

(b) Phase lag plot

Figure 4.11: Results based on model augmentation without augmenting the observer in advance (phase angle plot)



(a) Tracking error

(b) Phase lag plot

Figure 4.12: Results based on reduced-order controller implementation

the parameters converge much faster than Scenario No. 2; nevertheless, parameter history plot for the Scenario 3 does show some oscillation during the transition.

To confirm the results, we simulate an unit step input for the closed loop system based on the full implementation of the model augmentation with tuned parameters at various stages as shown in Figure 4.13.

Clearly, we see the step response improve as compared to each of its prior stage. We can see both the settling time and rise time have been improved significantly.

Based on the simulation results, we see clearly that the proposed *extended progressive learning*

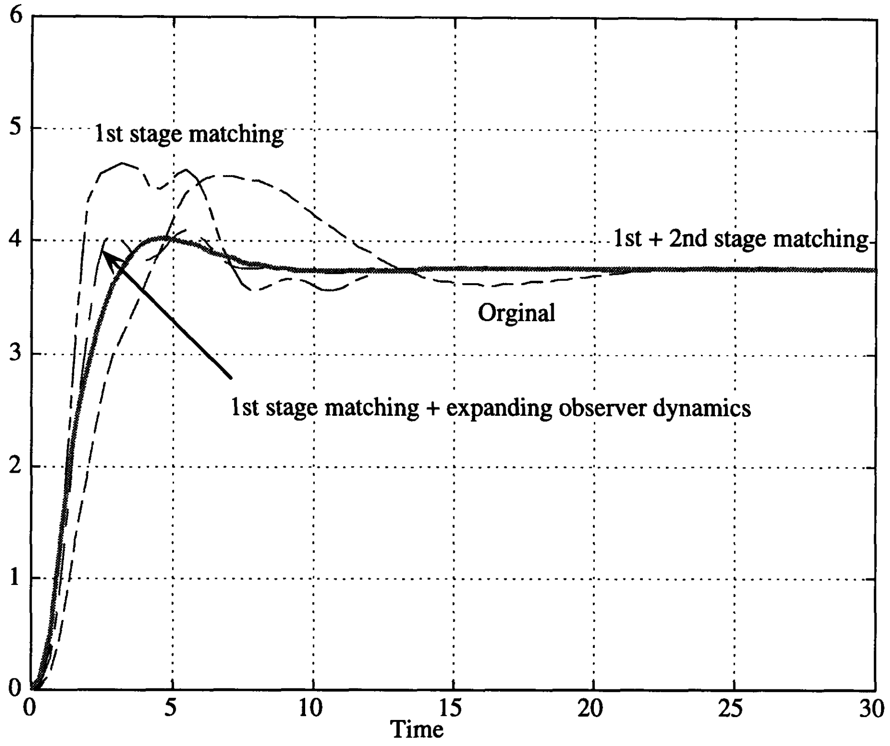


Figure 4.13: Step response based on simulation results

algorithm shows a better results than the conventional tuning. The main purpose of this chapter is to suggest a procedure for the user who is interested the progressive learning algorithm with model augmentation and trajectory synthesis to apply. Thus, some of the steps can be alter depending on the user's preference and the actual application may be. Nevertheless, with this way of progressively augmenting the observer dynamics, model dynamics, and expanding the parameter space, we can guarantee the overall stability while maintaining fast convergence since at any instance, only one mode is been "learned" or "adapted."

With this, we implement the *extended progressive learning* algorithm to three actual applications: direct-drive robot, linear slider, finally, and a two-dimensional coordinate measuring machine.

# Chapter 5

# Implementation

In this chapter, the proposed research is implemented to three different applications: a ultra-high speed direct-drive robot, a single degree-of-freedom linear slider, and a two degree-of-freedom coordinate measuring machine. Within each section, experimental setup and results are presented.

## 5.1 Implementation I: Ultra-High Speed Chip-placement Machine

In this section, we are going to show our first implementation of the model augmentation and trajectory synthesis through the use of progressive learning to an ultra-high speed chip-placement machine that we designed here. The detail of the design concept and methodology together with the priliminary experimental results for the chip-placement machine can be seen in [Li, et al., 1994], thus are ommitted here. In this section, the experimental setup together with the results from the chip-placement machine after we applied the *extended progressive learning* are presented.

### 5.1.1 Experimental Setup

From earlier section, we have shown our hardware in Figure 5.1. As shown in the figure, the second link is driven through a steel belt. An encoder is mounted on each joint and a PSD laser pointer is mounted at the tip of the second link for the use of a 2D PSD sensor that is located on a target trajectory. For accurate trajectory control, we feedback the end-point sensor signals to control the direct-drive robot. Therefore, our system results in a non-collocated system.

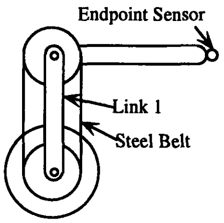


Figure 5.1: Experimental setup

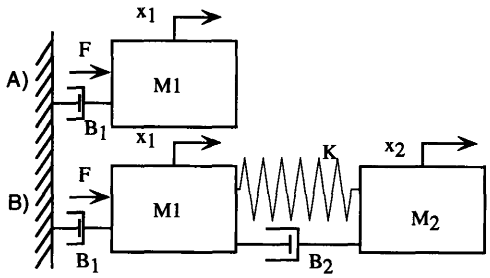


Figure 5.2: Lumped parameter models with different system orders

For simplicity, we assume the following conditions to the experimental setup:

1. All parameters involved are time-invariant,
2. The maximum operation speed is less than the excitation level of the flexible mode of the link but high enough to excite the transmission belt dynamics,
3. The first link is maintained immobile,
4. The sensor noises are small, and
5. The assembly process and the system are all deterministic.

Based on these assumptions and simplifications, Figure 5.2 shows possible lumped parameter models of the second joint control system. At low frequencies, the phase lag between the motor rotor and the joint axis due to the belt compliance is negligible. Therefore, the whole movable part can be modeled as a single rigid body, as shown in Figure 5.2A. Both the joint encoder and the endpoint sensor provide basically the same position information, as long as the robot tracks a trajectory at a low speed and the frequency spectrum of the trajectory is within a low frequency range. Tuning of control parameters is rather straight forward for this collocated system. As the frequency spectrum expands, the system model must be updated to the one involving the belt compliance. A shown in Figure 5.2B, the single rigid body is now splitted into the motor rotor side inertia, $m_1$, and the arm link side inertia, $m_2$. Both masses are connected by the belt stiffness $k$, whereas $b_1$ represents the viscous damping at the base of the second motor and $b_2$ represents the viscous damping of the belt structure.

Based on this simplified model, one can easily derive the following transfer functions:

$$\frac{x_2}{F} = W_{p_1} =$$

$$\frac{\left(\frac{b_2}{m_1 m_2}\right)\left(s+\frac{k}{b_2}\right)}{s^4+\left[\frac{b_1+b_2}{m_1}+\frac{b_2}{m_2}\right]s^3+\left[\frac{k}{m_1}+\frac{k}{m_2}+\frac{b_1 b_2}{m_1 m_2}\right]s^2+\frac{b_1 k}{m_1 m_2}s} \tag{5.1}$$

### 5.1.2 Plant and Control Structure

At low frequency, the transmission dynamics can be ignored so the system is simplified to be a second order system. At high frequency, however, the transmission dynamics can no longer be ignored. Thus, the system model has to be changed into a fourth-order system model. Using a system identification method based on the actual data provided from various data sheets and experiments, we obtained the plant model using the second-order approximation and the fourth-order approximations follows:

$$W_{p_1}(s) = \frac{26418}{s^2+301.13s} \tag{5.2}$$

$$W_{p_2}(s) = \frac{2.46\times 10^7(s+0.76)}{(s^4+391.320s^3+57289s^2+980350s)} \tag{5.3}$$

A reference model for the adaptive control has to be chosen so that it represents a desired dynamics while the relative order of the model is equal to that of the plant. To meet these requirements, we chose the following second-order and fourth-order reference models, which are used for the second order and fourth order approximations of the plant respectively:

$$W_{m_1}(s) = \frac{101710}{s^2+314s+2467} \tag{5.4}$$

$$W_{m_2}(s) = \frac{9.94\times 10^7(s+10)}{(s^2+314s+2467)(s^2+120s+11700)}. \tag{5.5}$$

The switching frequency is determined when $\omega_s$ is equal to $14hz$ or when $F_{r_t}$ is equal to $10^o$. The pre-determined feedback component $\Delta(s)$ is chosen for each of the plant approximations as:

$$\Delta_1(s) = (s+1),\ \Delta_2(s) = (s+1)(s+1)(s+10). \tag{5.6}$$

The closed-loop plant characteristic equation are obtained for the second-order and fourth-order MRAC in terms of control parameters as follows:

$$\Phi_{p_1}(s) \;=\; (\Delta_1(s)-\theta_{11})R_p(s)-k_p Z_p\left(\theta_2+\Delta_1(s)\theta_0\right) \tag{5.7}$$

$$\begin{aligned}\Phi_{p_2}(s) &= \left[\Delta_2(s) - (\theta_{11} + \theta_{12}s + \theta_{13}s^2)\right] R_p(s) \\ &\quad - k_p Z_p \left[(\theta_{21} + \theta_{22}s + \theta_{23}s^2) + \Delta_2(s)\theta_0\right] \qquad (5.8)\end{aligned}$$

where the control parameter vector for the second-order approximated plant is defined as:

$$\theta_{2nd} \overset{\text{def}}{=} [\theta_1,\ \theta_2,\ \theta_0,\ k_1]^T \qquad (5.9)$$

and the control parameter vector for the fourth-order approximation plant is defined as:

$$\theta_{4th} \overset{\text{def}}{=} [\theta_{11},\ \theta_{12},\ \theta_{13},\ \theta_{21},\ \theta_{22},\ \theta_{23},\ \theta_0,\ k_2]^T \qquad (5.10)$$

The parameters are tuned based on the progressive learning method. The tuned parameters based on the estimated second-order system parameters are: $k_{1_t} = -3.86$, $\theta_{1_t} = -13.0328$, $\theta_{2_t} = -0.1461$, $\theta_{t_0} = -0.7752$. The tuned parameters based on the estimated 4th-order system parameters are: $k_{2_t} = -4.07, \theta_{1_t} = [9.1827, -86.5461, -32.8461]^T, \theta_{2_t} = [-10.4583, -16.7737, -8.9612]^T$, $\theta_{0_t} = -0.1301$. The feedforward gain $k_1$ and $k_2$ were all initialized to 1 whereas the other control parameters were initialized to zeros. Figure 5.3 shows the phase shift plots of the reference model characteristic functions and the initial closed-loop characteristic functions.

### 5.1.3 Learning Procedure

The desired trajectory is generated based on the following set of target points in $radius$.

$$\begin{aligned}\text{targ pts} &= [0,\ 0.106,\ -0.073,\ 0.439,\ -0.437,\ 0.443, 0.576, \\ &\quad 0.537,\ -0.652,\ 0.416,\ -0.365,\ 0.319,\ -0.253, \\ &\quad 0.071,\ -0.114,\ 0.012,\ -0.167,\ 0.012] \qquad (5.11)\end{aligned}$$

In the experiments, a friction compensation was also employed independently from the MRAC control loop in order to alleviate the non-linearity of the frictional effect as much as possible.

In order to demonstrate the effectiveness of the progressive learning, we ran the experiments in three series: 1) uses only the second-order reference model, 2) uses the fourth-order reference

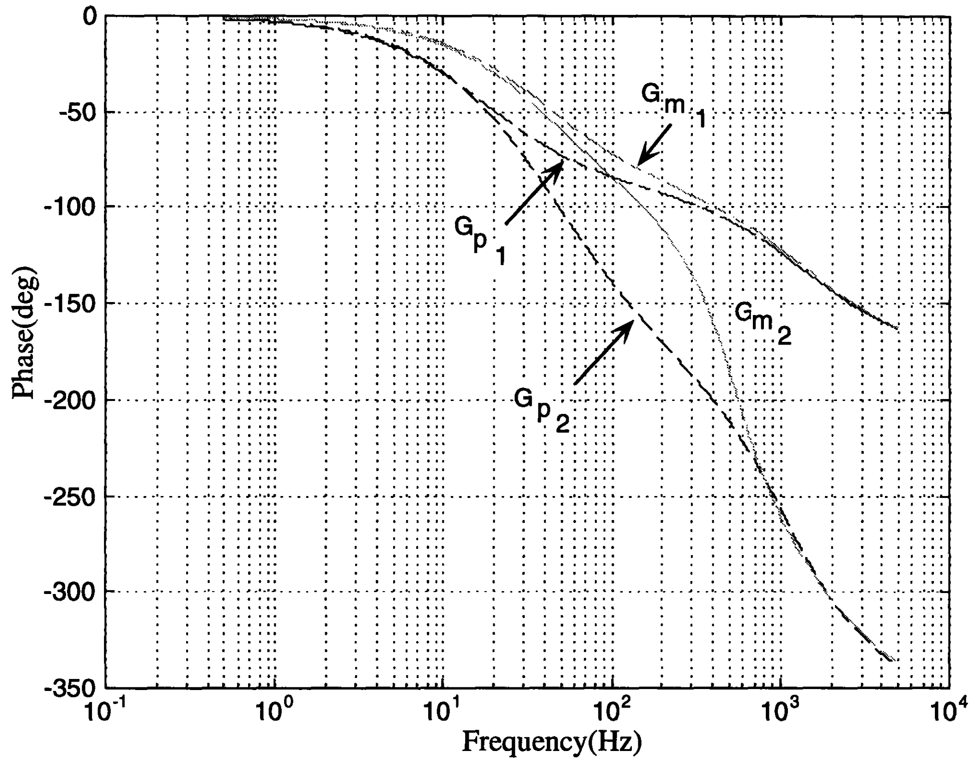


Figure 5.3: Phase plot of various transfer functions

model; and 3) starts with the second-order reference model and switches to the fourth-order reference model after the excitation level reaches the bandwidth of the fourth-order dynamics. For each series, the desired trajectory generated from the target points as presented earlier increases its speed. This, in turn, increases the excited frequency bandwidth. The notation we used to denote the speed is specified by its numerator; i.e., $10s/P$ denotes that each period is completed in 10 sec. Figure 5.4 shows the excitation frequency bandwidth with various speed and Table 5.1 shows how each series is run.

Table 5.1: History of the Experiments for DD-Robot

| Series No. | 10s/P | 8,6,5 s/P | 4 s/P | 3,2 s/P |
|---|---|---|---|---|
| #1:1:2nd-od | 4x | 2x | 2x | n/a |
| #2:4th-od | 4x | 2x | 2x | 2x |
| #3:2,4th | 4x | 2x | 4x | 2x |

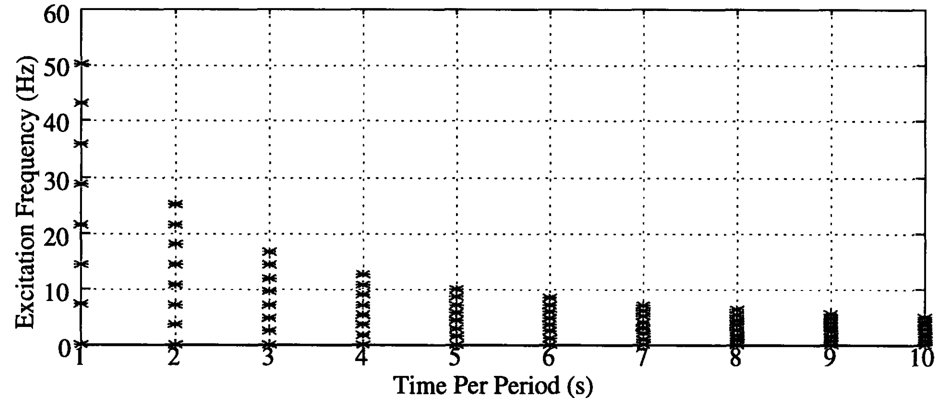


Figure 5.4: Excited frequency bandwidth for various speed

### 5.1.4 Results and Discussions

Figures 5.5- Figure 5.10 show the results for the three sets of experiments that we described earlier. Based on these results, we can clarify three major points: bound on tracking error, stability of the progressive learning, and parameter convergent speed with the use of gradual model augmentation method.

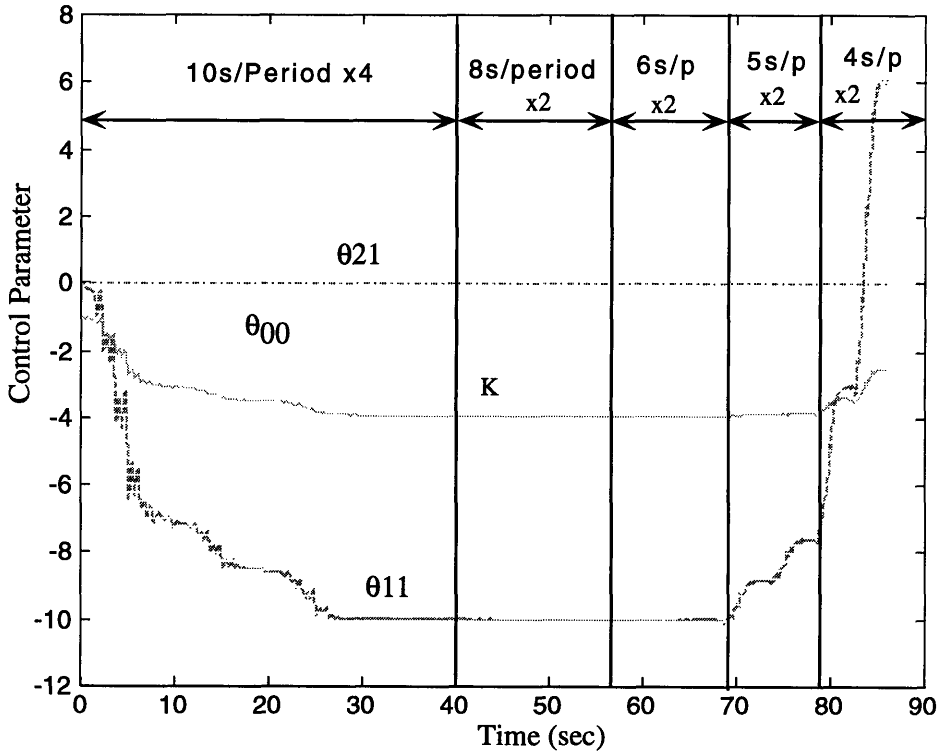


Figure 5.5: Control parameter history based on a 2nd-order MRAC formulation

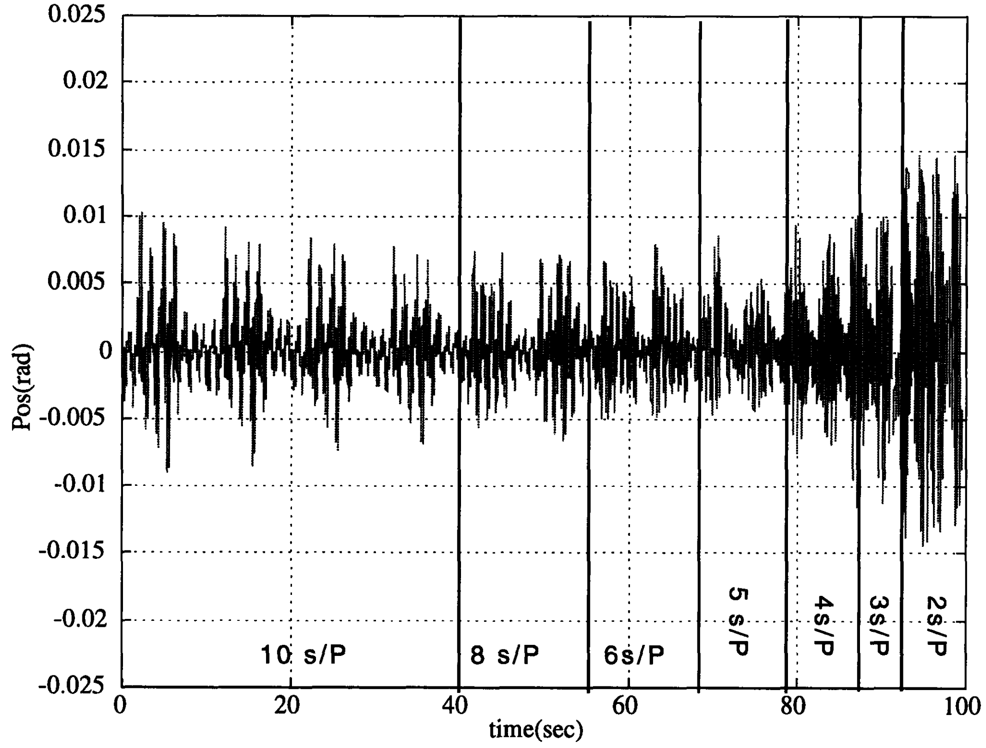


Figure 5.6: Error plot based on a 2nd-order MRAC formulation

Figure 5.6 shows the error plots for the first set of experiments which were conducted under the assumption that the plant is a second-order system, and Figure 5.8 shows the error plots of the second set of experiments which assumes that the plant is a fourth-order system, while Figure 5.10 exhibits the errors of the third set of experiments which started with the second-order plant model and later augmented to the fourth-orderplant model. Our primary concern for this research is the tracking accuracy. Although the desired trajectory was synthesized based on the inverse reference model transfer function independently of the plant, the tracking errors quickly converged to within the tolerance. Small errors, however, remain due to other non-linearity effect of the system and sensor noise that we ignored in modeling.

Figure 5.5 shows the learning curves of the control parameters for the first set of experiments which assume the plant is a second-order system. As shown in this figure, the stability and parameter convergence were acquired by the progressive learning. By gradually exciting the system, we have achieved the stable adaptive control even though the excitation signals are band-limited with

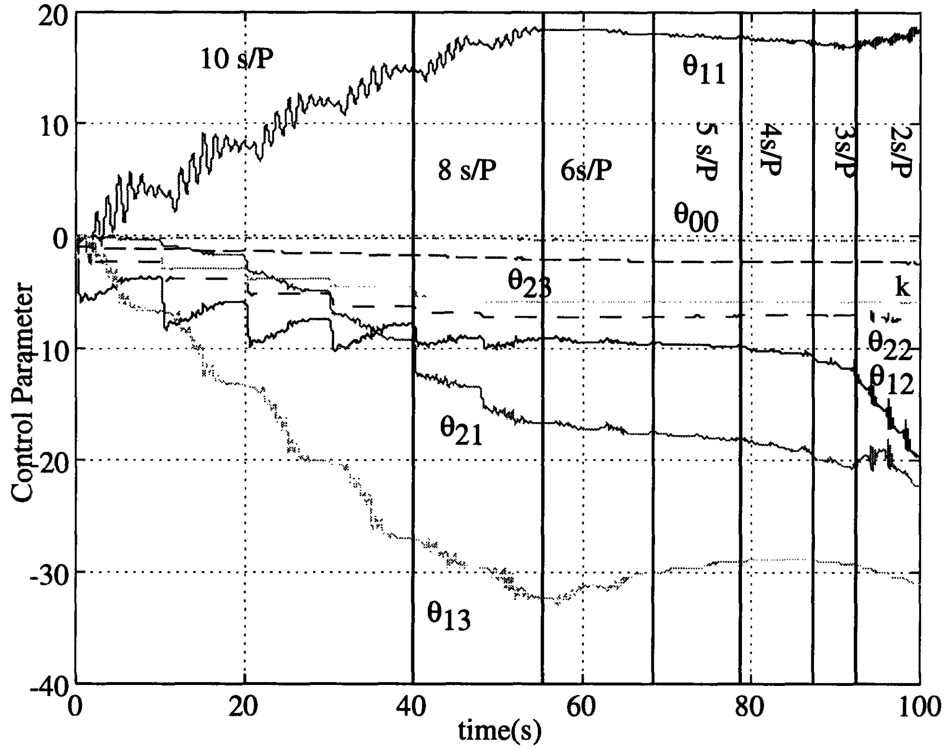


Figure 5.7: Control parameter history based on a 4th-order MRAC formulation

respect to the overall system bandwidth and the relative order of the plant is two or higher.

Figure 5.7 shows the learning history of the eight control parameters for the second set of experiments where the controller were constructed based on the fourth-order plant assumption. Due to the fact that the phase discrepancy between the desired and actual system is large as shown in Figure 5.3, the control parameter convergence is quite slow. This result is consistent with the stability analysis we have shown earlier. Figure 5.9 shows the history of the control parameters for the third set of experiments. In this particular set, we constructed a low-order controller based on the assumption that the plant is a second-order system, and, therefore, only four control parameters were adapted. As we increased the excitation level through the increased speed of the trajectory tracking command, the belt dynamics was excited. This happened when the excitation exceed $14hz$ or when the period of the tracking reaches 4 second per period. Once the controller sensed this change, the original second-order reference model was augmented to a fourth-order reference model. Thus, the number of the control parameters to be adapted increased from 4 to 8. By comparing the results shown in Figures 5.7 and Figure 5.9, it is found that the convergence was accelerated by this

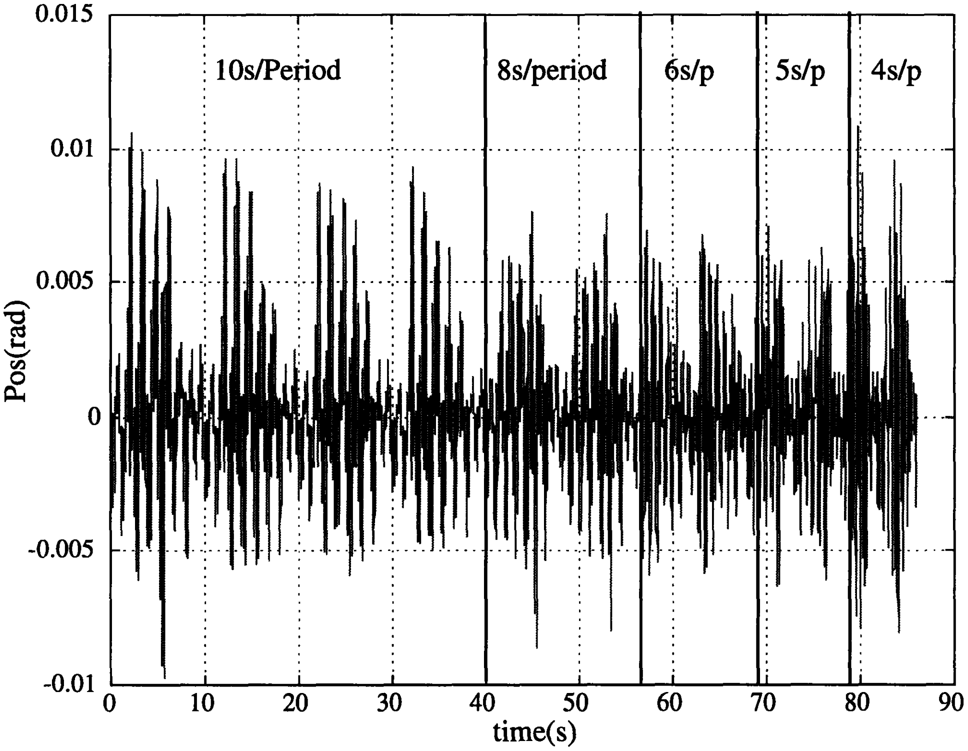


Figure 5.8: Error plot based on a 4th order MRAC formulation

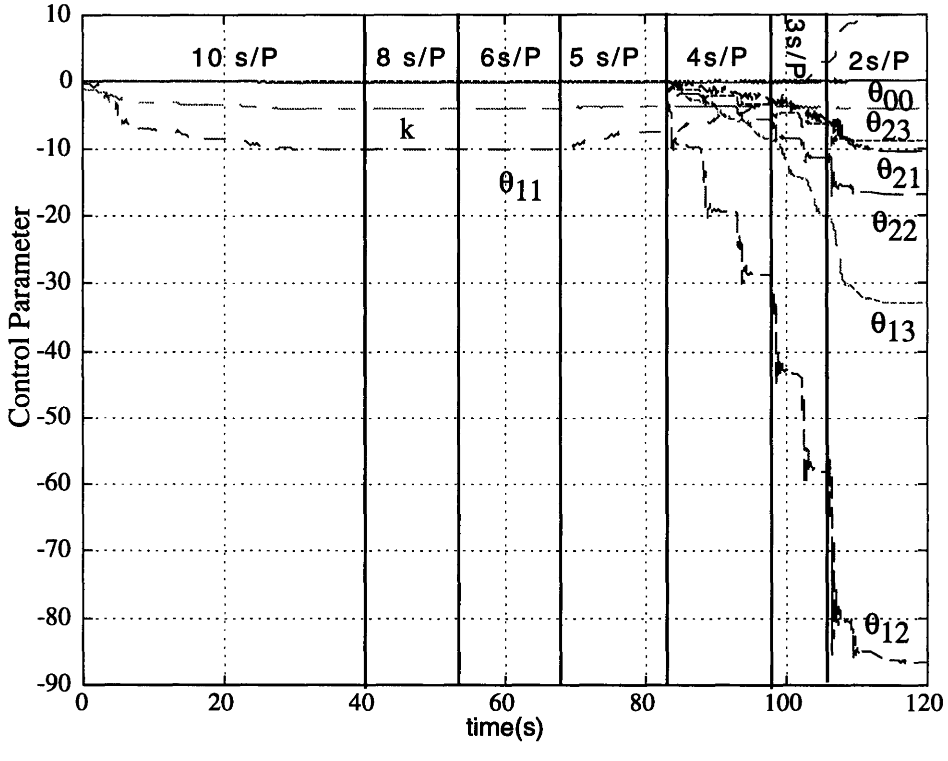


Figure 5.9: Control parameter history based on both a 2nd-order and a 4th-order MRAC formulation

method of augmenting the controller as well as the reference model. One explanation for this observation is that after learning a trajectory at a slow speed, the stable range of excitation frequency has been expanded. More specifically, in the beginning of experiments, since the number of the control parameters is much less with a low-order assumption, it is easier for the parameter to converge. Once the switching is made, the controller has already some information about the lower-order system dynamics; thus, adjustment for the controller to expand from four control parameters to eight is quite easy. Graphically, as seen in Figure 5.3, the phase discrepancy between the desired and actual plant model is below 90 degrees, the maximum allowed phase shift to maintain parameter adaptation stability and convergence. Thus, we can tune the system by fully exciting the system up to the system bandwidth while parameter convergent stability is guaranteed and tracking performance is maintained. Figure 5.9 shows clearly that eight control parameters converged smoothly and quickly to their true values as compared with the results in Figure 5.7. These results clearly demonstrated the effectiveness of the model augmentation method. Consequently, we achieved both trajectory control and control parameter convergence by using progressive learning and trajectory synthesis as demonstrated in these three sets of experiments. Finally, to have a fair comparison between the traditional PD control and adaptive control, we performed a step response test as shown in Figure 5.11. As shown in [Li, et al., 1994], the best PD control step response we were able to achieve is around 50ms. With the fourth-order adaptive control tuned using model augmentation method, we are able to achieve 35ms. The position error is 0.0015rad. The step response result also shows that adaptive control that assumed second-order system is better than the PD control due to the fact that it has compensator built in as part of its control parameter space.

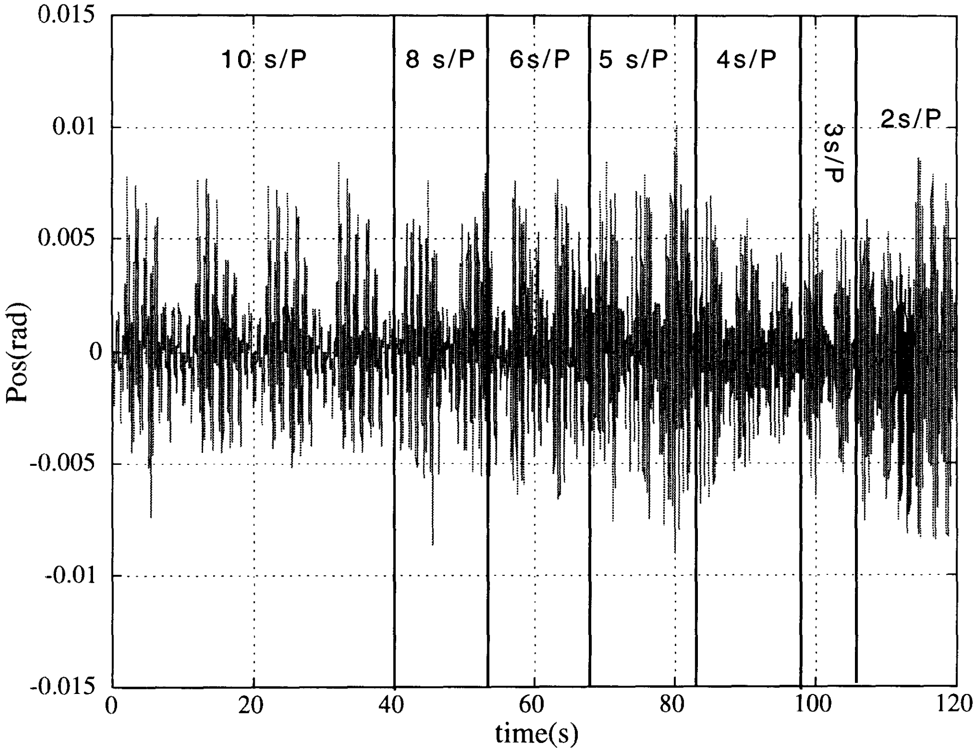


Figure 5.10: Error plot based on both a 2nd-order and a 4th-order MRAC formulation

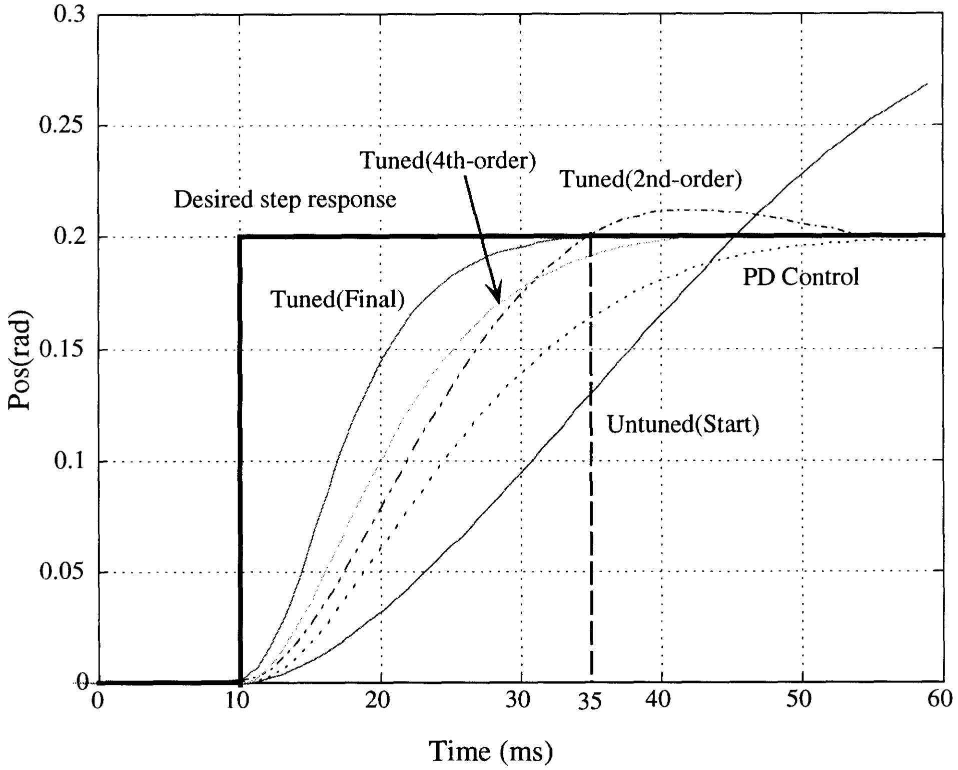


Figure 5.11: Step response plot

## 5.2 Implementation-II: Single Degree-of-Freedom Linear Slider

In this section, we will describe how we implemented the *extended progressive learning* algorithm to a single degree of freedom linear slider and the results we obtained. In this particular experiment, our main point here is to show the effect of our new control technique with the use of trajectory synthesis as described earlier. Thus, we choose a single dimension linear slider as our experimental setup. The experimental setup and goal are first introduced, then they are followed by the implementation results.

### 5.2.1 Experimental Setup

The high-speed single-axis bi-directional linear slider can be seen in Figure 5.12. It was specially designed and built to replace the leadscrew type linear drive mechanism to minimize the manufacturing cost. As shown in the figure, the leadscrew is replaced by a twin-gear drive and a single reinforced rubber belt. The twin-gear drive mechanism was designed to have a large gear reduction without a large physical gear ratios to save space and to have almost a zero bi-directional backlash. The large transmission ratio helps to achieve high accuracy without using an expensive high resolution encoder. One draw-back of using the reinforced rubber belt and twin-geartrain is that the stiffness is much lower as compared to the leadscrew type drivetrain. At such high speed, the belt dynamics can not be ignored. Since the feedback is based on the carrier's motion, this system is in fact a non-collocated actuator/sensor system. Our purpose here is to demonstrate the use of the trajectory synthesis. The complete setup can be seen in Figure 5.13. An IBM compatible 486/50Mhz computer and a torque controlled PWM servo amplifier are used for the experiments.

The targeted settling time after completing a full travel length of 50cm at full speed capacity is 30ms with accuracy of 50 $\mu$m and minimal overshoot. The prior experimental results show that the slider can best achieve 50 $\mu$m with the settling time of 50ms.

### 5.2.2 Plant and Control Structure

As shown earlier, it uses a plastic belt to transmit the power to the carrier. Since the feedback signals are obtained based on the carrier position information, the system is in fact a non-collocated

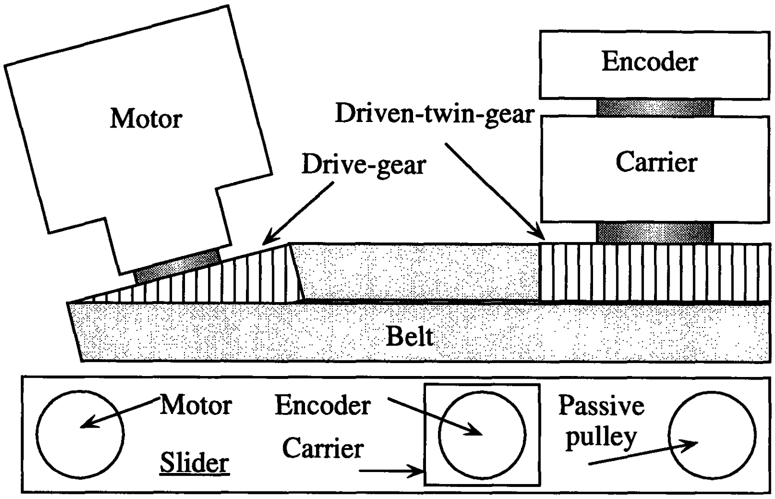


Figure 5.12: The single directional linear slider

system. At low frequency, the transmission dynamics can be ignored so the system is simplified to be a second order system. At high frequency, however, the transmission dynamics can no longer be ignored. Thus, the system model has to be changed into a fourth-order system model.

Using a system identification method based on the actual data provided from various data sheets and experiments, we obtained the plant model using the second-order approximation and the fourth-order approximations follows:

$$G_{p_1}(s) = \frac{785.3982(s+5)}{s^2+30s} \tag{5.12}$$

$$G_{p_2}(s) = \frac{3.9243 \times 10^6(s+5)}{(s^4+118.1139s^3+2.9743 \times 10^4 s^2+75s)} \tag{5.13}$$

One thing needs to be clarify is that though we have obtained these coefficients for the two models, we still assume that plant model is unknown. We never use these coefficients at any time during the experiments. The models presented here are only for the purpose of validation of experimental

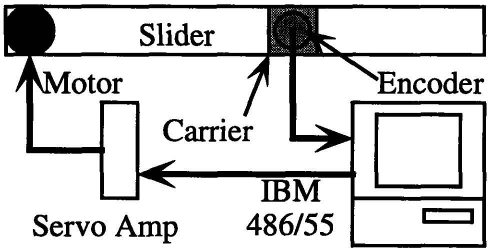


Figure 5.13: Experimental setup of the linear slider

results.

A reference model for the adaptive control has to be chosen so that it represents a desired dynamics while the relative order of the model is equal to that of the plant. To meet these requirements, we chose the following second-order and fourth-order reference models, respectively:

$$G_{m_1}(s) = \frac{4.7124 \times 10^3(s+5)}{s^2+179s+8826} \tag{5.14}$$

$$G_{m_2}(s) = \frac{6.2788 \times 10^7(s+5)}{(s^2+179s+8826)(s^2+219s+13324)}. \tag{5.15}$$

The pre-determined feedback component $\Delta(s)$ is chosen for each of the plant approximations as:

$$\Delta_1(s) = (s+5),\ \Delta_2(s) = (s+5)(s+5)(s+10). \tag{5.16}$$

The closed-loop plant characteristic equation are obtained for the second-order and fourth-order MRAC in terms of control parameters as follows:

$$\begin{aligned} \Phi_{p_1}(s) &= (\Delta_1(s) - \theta_{11})R_p(s) - k_p Z_p\,(\theta_2 + \Delta_1(s)\theta_0) && (5.17)\\ \Phi_{p_2}(s) &= \left[\Delta_2(s) - (\theta_{11} + \theta_{12}s + \theta_{13}s^2)\right] R_p(s) \\ &\quad -k_p Z_p\left[(\theta_{21} + \theta_{22}s + \theta_{23}s^2) + \Delta_2(s)\theta_0\right] && (5.18) \end{aligned}$$

where the control parameter vector for the second-order approximated plant is defined as:

$$\theta_{2nd} \stackrel{\text{def}}{=} [\theta_1,\ \theta_2,\ \theta_0,\ k_1]^T \tag{5.19}$$

and the control parameter vector for the fourth-order approximation plant is defined as:

$$\theta_{4th} \stackrel{\text{def}}{=} [\theta_{11},\ \theta_{12},\ \theta_{13},\ \theta_{21},\ \theta_{22},\ \theta_{23},\ \theta_0,\ k_2]^T \tag{5.20}$$

The parameters are tuned based on the progressive learning method. The tuned parameters based on the estimated second-order system parameters are: $k_{1_t} = 6.01$, $\theta_{1_t} = 0.0001$, $\theta_{2_t} = -10.9307$, $\theta_{0_t} = -0.0758$. The true parameters based on the estimated 4th-order system parameters are: $k_{2_t} = 16.002$, $\theta_{1_t} = [-18.172, -41.88, -11.11]^T$, $\theta_{2_t} = [-14.408, -36.74, -25.4]^T$, $\theta_{0_t} = -0.3$. The feedforward gain $k_1$ and $k_2$ were all initialized to 1 where as the other control parameters were initialized to zeros. The $\Gamma_{max}$ is once again set to be $10^o$, thus projected $\omega_s$ is $6hz$.

### 5.2.3 Learning Procedure

The desired trajectory as shown in Figure 3.13 is generated based on the following set of target points with unit of $cm$.

$$\text{targ pts} = [0,\ 1,\ 2,\ 2,\ 0,\ 0,\ 4,\ 4,\ 0,\ 0,\ -3,\ -3,\ 0,\ 0,\ -2,\ -2,\ -1,\ 0] \qquad (5.21)$$

In order to demonstrate the effectiveness of the progressive learning, we ran the experiments in two series: 1) uses the fourth-order reference model and 2) starts with the second-order reference model and switches to the fourth-order reference model after the excitation level reaches the bandwidth of the fourth-order dynamics. For each series, the desired trajectory generated from the target points as presented earlier increases its speed. This, in turn, increases the excited frequency bandwidth. The notation we used to denote the speed is specified by its numerator; i.e., $10s/P$ denotes that each period is completed in 10 sec. Table 5.2 shows how each series is run.

Table 5.2: History of the Experiments for the Linear Slider

| Series No. | 10s/P,7.5s/P | 5, s/P | 2.5 s/P | 2. s/P | 1 s/P |
|---|---|---|---|---|---|
| #1:4th-od | 2x | 2x | 3x | 2x | 18x |
| #2:2,4th | 2x | 3x | 3x | 2x | 13x |

### 5.2.4 Results and Discussions

From the error plots for both experiment as shown in Figure 5.14 and Figure 5.15, we see that both setups have surpassed the accuracy of 50$\mu$m requirements at the end of experiments. As the system is tuned to the desired system, the tracking error between the actual system and desired system decreases as shown in these two figures. As shown in Figure 5.16 and Figure 5.17 where the pole-zero plots for each set and Figure 5.18 and Figure 5.19 where the phase plots for each

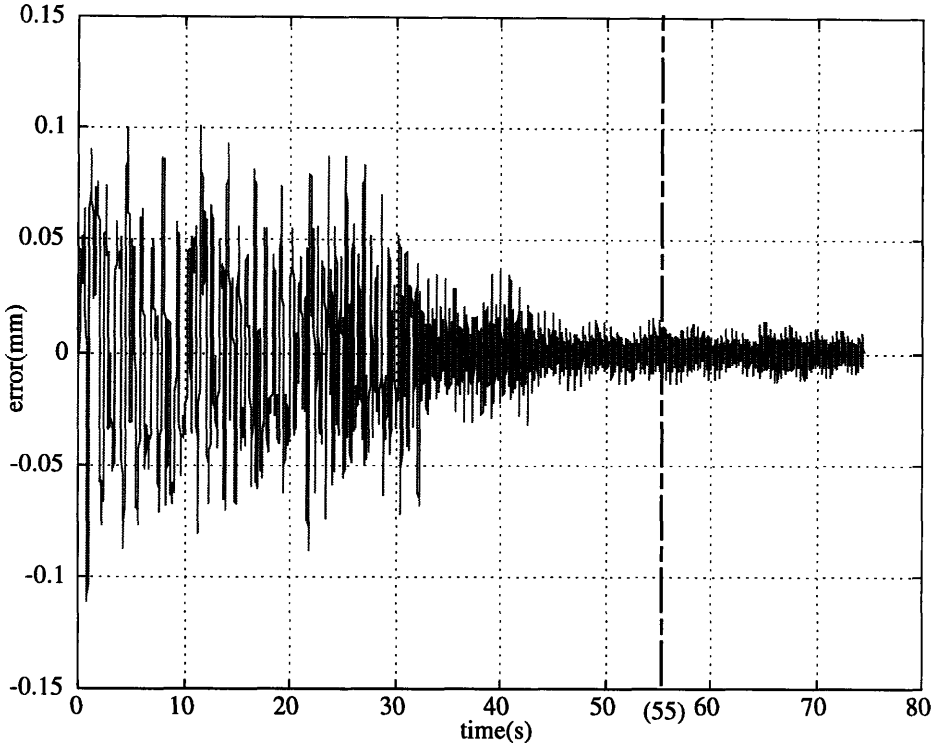


Figure 5.14: Error plot based on a 4th order MRAC formulation

set, the system is always stable throughout the whole duration of training. More importantly, the parameters all converged toward the desired parameters as shown in Figure 5.21 and Figure 5.22. The pole-zero plots obtained earlier, represents the snap shot of the closed-loop system pole-zero locations as we varied the controller parameters. Figure 5.20 show experimentally, the existence of $\varepsilon$ and gradual improvement as proposed by the progressive learning together with the trajectory synthesis.

Earlier, we proposed that we can progressively learn the system at a faster pace by gradually increasing the assumed model complexity as we gather more information about the system. By comparing the results shown in Figures 5.21 and Figure 5.22, we have found that the convergence rate was accelerated by this method of augmenting the controller as well as the reference model. One explanation for this observation is that after learning a trajectory at a slow speed, the stable range of excitation frequency has been expanded. More specifically, in the beginning of experiments, since the number of the control parameters is much less with a lower order assumption, it is easier for parameters to converge. Graphically, from Figure 5.19, we see the use of reduced-order

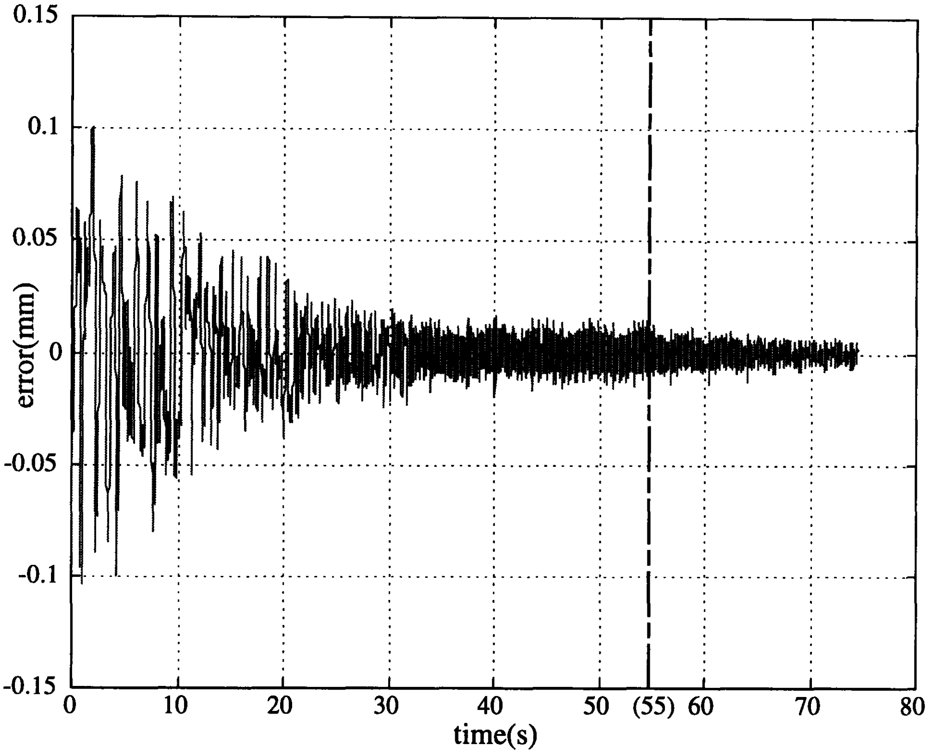


Figure 5.15: Error plot based on both a 2nd-order and a 4th-order MRAC formulation

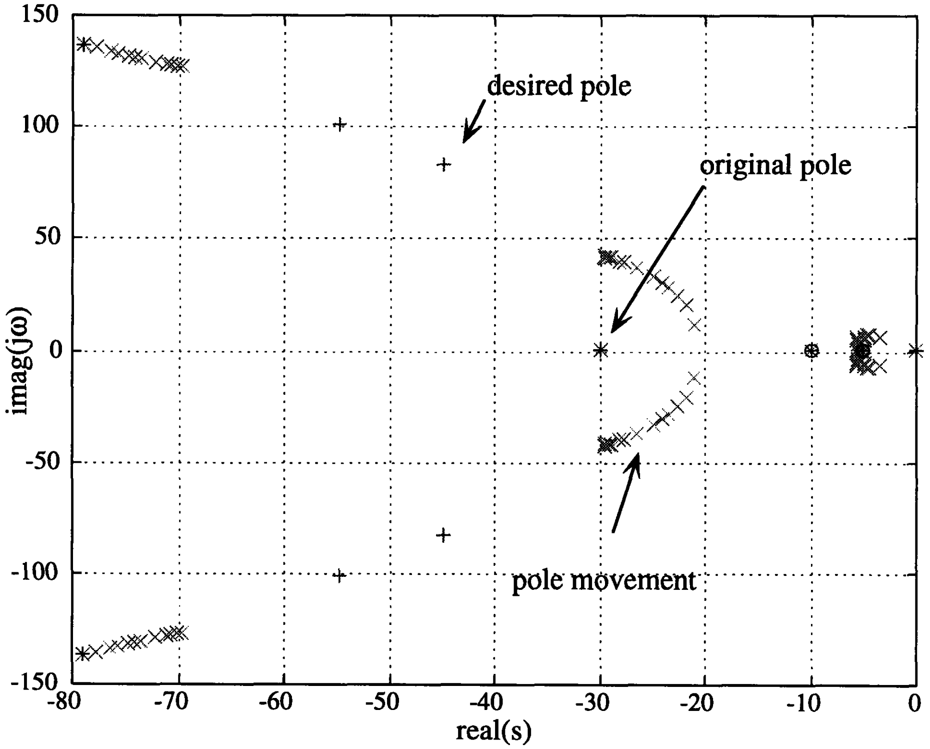


Figure 5.16: Pole-zero plot based on a 4th-order MRAC formulation

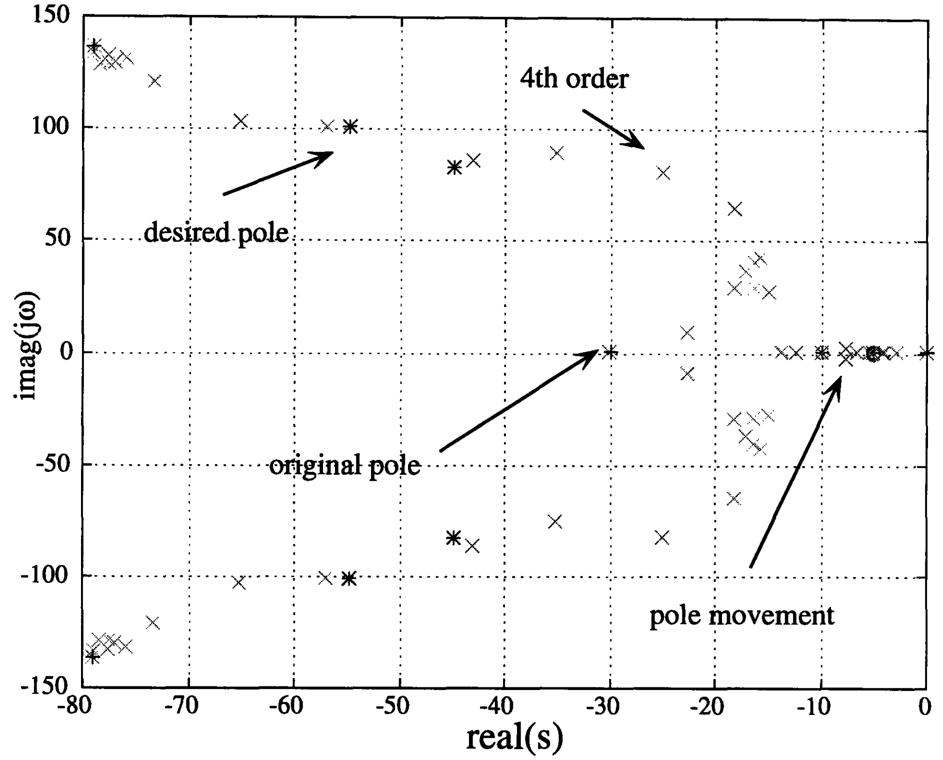


Figure 5.17: Pole-zero plot based on both a 2nd-order and a 4th-order MRAC formulation

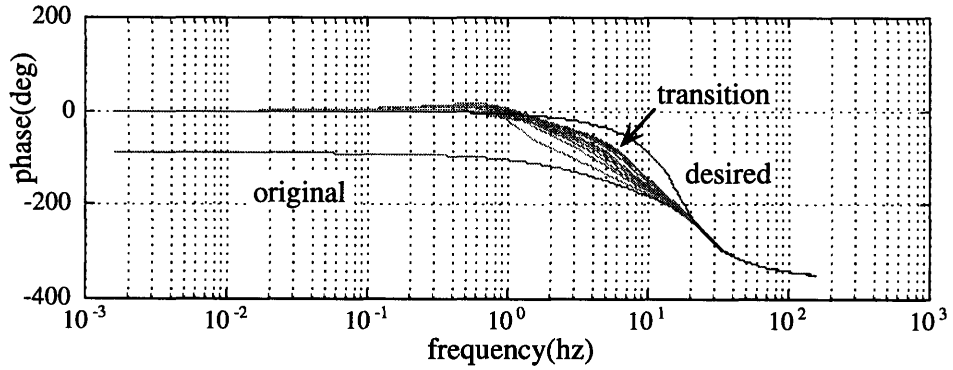


Figure 5.18: Phase plot based on a 4th-order MRAC formulation

reference model assumption allows the system to adapt to an intermediate state. As the excitation reaches $5.5Hz$, the experimental validated $\omega_s$, we performed the augmentation when the tracking period is equal to 5 sec. Once the switching is made, the controller has already some information about the reduced-order system dynamics; thus, adjustment for the controller to expand from four control parameters to eight or from reduced-order system to full-order system is quite easy. Graphically, as seen in Figure 5.19, the phase discrepancy between the desired and actual plant model is

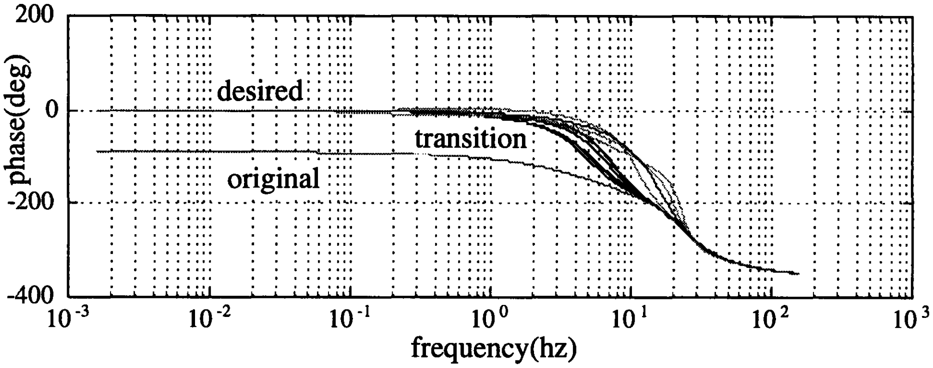


Figure 5.19: Phase plot based on both a 2nd-order and a 4th-order MRAC formulation

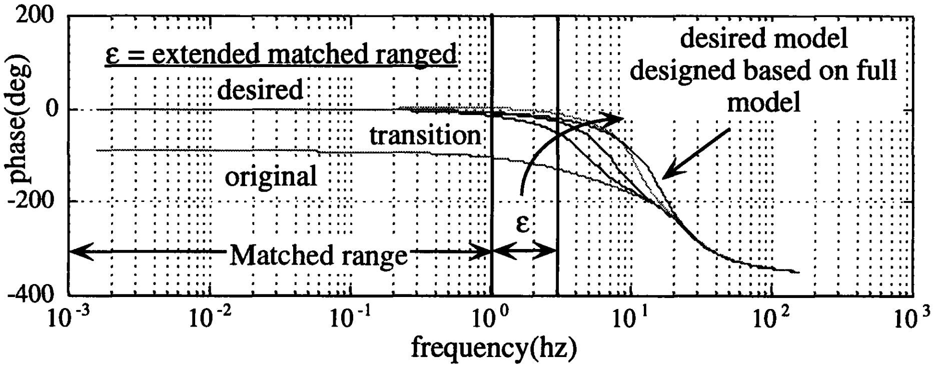


Figure 5.20: Experimental validation of the transfer function matching

less, thus, we can tune the system by fully exciting the system up to the system bandwidth while parameter convergent stability is guaranteed and tracking performance is maintained. Figure 5.22 shows clearly that eight control parameters converged smoothly and quickly to their true values as compared with the results in Figure 5.21. These results clearly demonstrated the effectiveness of the model augmentation method. Consequently, we achieved both trajectory control and control parameter convergence by using progressive learning and trajectory synthesis as demonstrated in these two sets of experiments.

Finally, in order to demonstrate the improvement we have made by implementing the trajectory synthesis to the slider, we perform a tracking test. The length of travel is 50cm. A laser proximity sensor is mounted at the end of the course to detect the settling time and tracking error. The results between the prior result using the PD controller as compared to the newly improved system using extended progressive learning method is shown in Figure 5.23. The x-axis represents time where each grid represents 50ms. The y-axis represents the position of the end-point detector with each

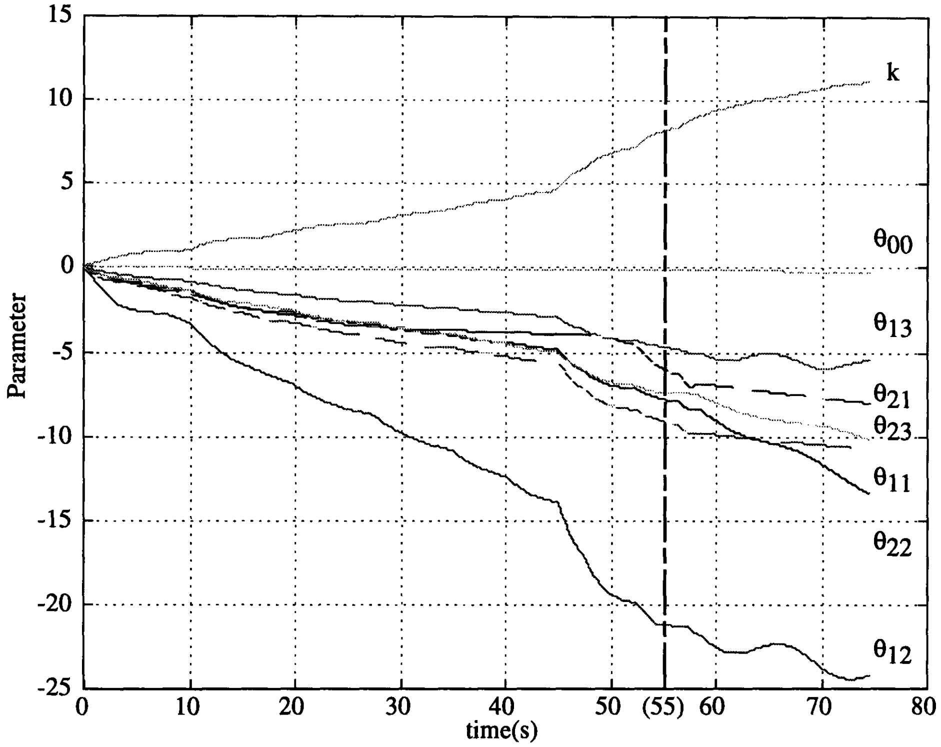


Figure 5.21: Control parameter history based on a 4th-order MRAC formulation

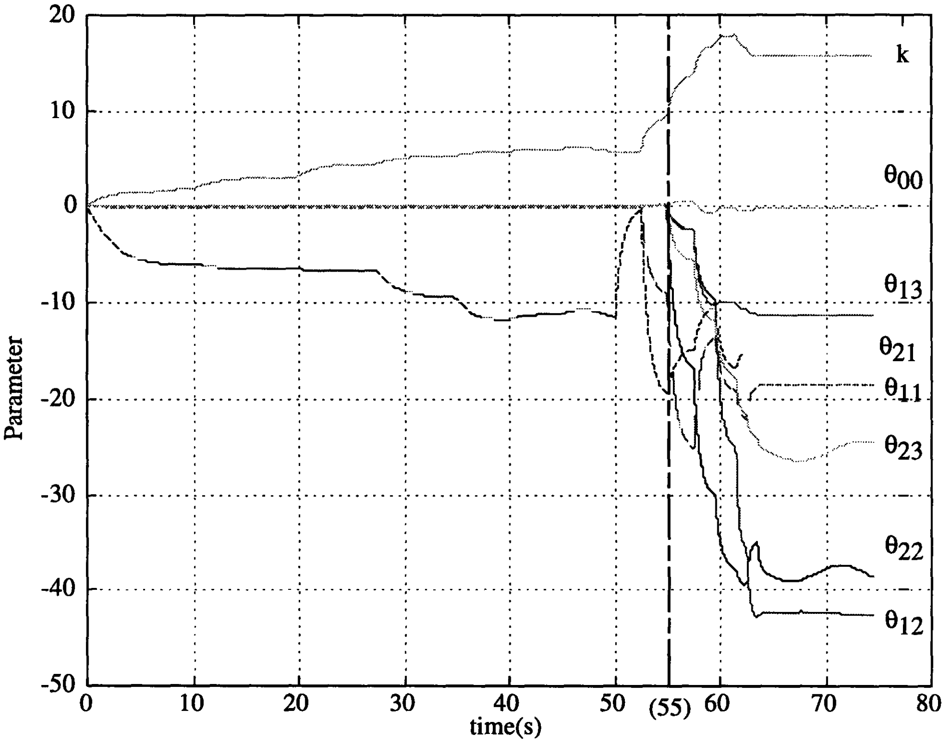


Figure 5.22: Control parameter history based on both a 2nd-order and a 4th-order MRAC formulation

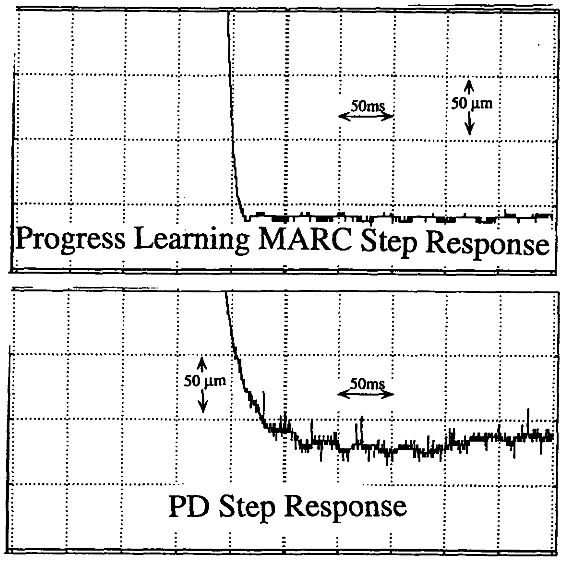


Figure 5.23: Step response plot

grid represeting 50 $\mu$m. We can see clearly that we have surpass our goal by achieving the settling time of 30ms while maintains the accuracy bound of $25\mu m$.

### 5.2.5 Conclusion

In this section, we showed through experiments that progressive learning together with the trajectory synthesis is an efficient method for tuning a high order, non-collocated robotic system. The main idea of progressive learning is to excite the system gradually in accordance with the progress of the adaptation. By incorporating a trajectory synthesis, we developed a method of generating a series of tracking trajectories that satisfy the stability conditions of progressive learning. As part of future works, our next step is to explore the controller augmentation method in association with the progressive learning and trajectory synthesis. We need to optimize the time to perform model augmentation and the way to transform the knowledge from the lower order model to higher model with a better mathematical theory. Nevertheless, the results clearly show that the convergence is much faster and tracking error is much smaller when the progressive model augmentation is used.

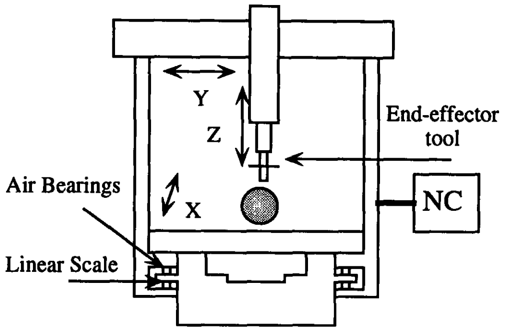


Figure 5.24: 3-D coordinate measurement machine

## 5.3 Implementation III: 2-Dimensional Coordinate Measurement Machine

The previous two experimental validation cases have focused on implementations of the endpoint feedback control of single dimensional systems using *progressive learning*. In this section, we have extended the study to control a 2 DOF coordinate measuring machine (CMM)). The CMM is first introduced before we described the experimental setup and how we achieve the endpoint control to this 2DOF CMM. Finally, we are going to show the single axis tuning results followed by the two-axes tuning results.

### 5.3.1 Control Problem of a Coordinate Measuring Machine

In this section, we apply the new adaptive control method to a three-dimensional (X-Y-Z) Cartesian coordinate measurement machine(CMM) as shown in Figure 5.24. Though the machine has three degrees of movements, we only focus on two dimensional movement, namely, the planar movement that involves $x$ and $y$ axis. We will first describe the CMM and the experimental setup.

#### 5.3.1.1 Experimental Setup

As shown in Figure 5.24, XYZ has three DOF. The x-axis has 2.0m travel length, y-axis has 1.6m travel length, and z-axis has 70cm travel length. Each linear axis is driven by an AC motor through a leadscrew. Instead of conventional ball-screw typed rails, all axes are run on air bearings. Each AC motor is equipped with a tachometer, and each axis has a Heidenhein linear encoder along the rails. The linear encoder has 1 $\mu$m resolution after the quadrature counter. Currently, the machine is running under two free-move fixed speeds: 6m/min and 12m/min. The maximum speed is limited by the maximum speed of the AC motor used. The AC motors are driven by a servo amplifier that has a PI controller built-in for the motor-side tachometer feedback signals. The outer control loop has a P control on the position that is based on the differentiation of the tachometer signals. The overall control block-diagram can be seen in Figure 5.25. Due to the assumption of the SISO system and the internal PI controller between the servo and servo tachometer, a low acceleration has been obtained; thus, during the actual scanning, the machine can best achieve scanning speed of 4 m/min with accuracy of ±25$\mu$m. To gain more physical insight of the system, we introduce a lumped parameter model for the 2DOF CMM machine.

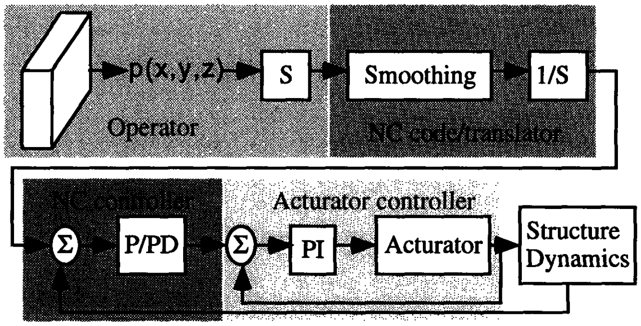


Figure 5.25: Current NC machine operation

### 5.3.1.2 Modeling of the CMM

Our main objective is to design a stable control algorithm to control the position of the tip of the CMM. Before designing the control algorithm, we will first study the effect of placing the sensors and actuators on the overall performance of the system. We begin by describing the model and the relevant assumptions we made to simplify the problem.

A simplified two-dimensional model of the CMM is shown in the Figure 5.26. The mass $m$ is attached to the rigid body $M$ which is free to translate and rotate about its center of mass. The following are the assumptions we made in formulating the equations of the model:

- The angle of rotation $\theta$ of the mass about the center of mass is small so that $\sin\theta \approx \theta$ and $\cos\theta \approx 1$.

- The actuator moves horizontally and does not rotate. The inputs to the system are the force $F_x$ and $F_y$ applied on actuators in the x-axis and y-axis respectively.
- The motor inertia $m_x$ and $m_y$ are small compared to the mass $M$ and $m$ and are thus ignored in considering the center of the mass and moment of inertia, $I_{cm}$
- The rotation due to the movement of y-axis is negligible.

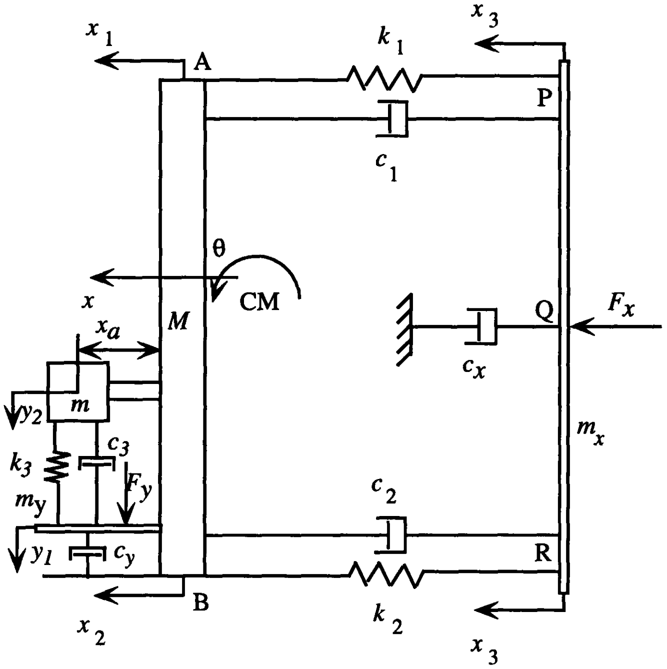


Figure 5.26: Two-dimensional model of a CMM with the force $F$ acting at $Q$

The number of degrees of freedom for this system are five and the generalized coordinates can be taken as $x, \theta, x_3, y_1$ and $y_2$ or equivalently $x_1, x_2, x_3, y_1$ and $y_2$. Physically, $x_1$ represents the x-axis linear encoder reading, $x_3$ represents the x-axis motor position reading, $y_1$ represents the y-axis motor position reading, and $y_2$ represents the y-axis linear encoder reading.

Based on these generalized coordinates, assumptions and model, we have obtained the following equations:

$$\begin{aligned}
(M+m)\ddot{x} &= -(c_1+c_2)\dot{x}-(k_1+k_2)x-(c_1y_2-c_2(L-y_2))\dot{\theta} \\
&\quad -(k_1y_2-k_2(L-y_2))\theta+(c_1+c_2)\dot{x}_3+(k_1+k_2)x_3, \\
I_{cm}\ddot{\theta} &= -\frac{Mm}{M+m}y_2\dot{y}_2\dot{\theta}-(c_1y_2^2+c_2(L-y_2)^2)\dot{\theta} \\
&\quad -(k_1y_2^2+k_2(L-y_2)^2)\theta-(c_1y_2-c_2(L-y_2))\dot{x}
\end{aligned}$$

$$\begin{aligned} & -(k_1y_2 - k_2(L-y_2))x + (c_1y_2 - c_2(L-y_2))\dot{x}_3 \\ & +(k_1y_2 - k_2(L-y_2))x_3, \qquad (5.22) \\ m\ddot{y}_2 &= -k_3(y_2-y_1) - c_3(\dot{y}_2-\dot{y}_1) + \frac{Mm}{M+m}\dot{\theta}^2y_2 \\ & -k1(x - x_3 - (L-y_2)\theta)\theta - k2(x - x_3 + y_2\theta)\theta, \qquad (5.23) \\ m_y\ddot{y}_1 &= -k_3(y_1-y_2) - (c_3+c_y)\dot{y}_1 + c_3\dot{y}_2 + F_y, \qquad (5.24) \\ m_x\ddot{x}_3 &= F - (M+m)\ddot{x} - c\dot{x}_3. \qquad (5.25) \end{aligned}$$

where $I_{cm} = \frac{Mm}{M+m}(x_a^2 + y_2^2)$, $L = 2 \times l$ is the point to point distance between the two linear screws.

Clearly, these five equations shown here are highly coupled with non-linear terms, thus, any SISO-type control may not work properly. This is also true even when we run only single axis at a time while inmobilizing the other axis. For the case that y-axis is immobilize, we can simplified these dynamics equations into the following:

$$\begin{aligned} & (M+m)\ddot{x} + (c_1+c_2)\dot{x} + (k_1+k_2)x \\ & \qquad + (c_1l_1 - c_2l_2)\dot{\theta} + (k_1l_1 - k_2l_2)\theta \\ & \quad = (c_1+c_2)\dot{x_3} + (k_1+k_2)x_3, \\ & I_{cm}\ddot{\theta} + (c_1l_1^2 + c_2l_2^2)\dot{\theta} + (k_1l_1^2 + k_2l_2^2)\theta \qquad (5.26) \\ & \qquad + (c_1l_1 - c_2l_2)\dot{x} + (k_1l_1 - k_2l_2)x \\ & \quad = (c_1l_1 - c_2l_2)\dot{x_3} + (k_1l_1 - k_2l_2)x_3, \text{ and} \\ & m_a\ddot{x_3} = F - (M+m)\ddot{x} - c\dot{x_3}. \end{aligned}$$

where $l_1$ is the distance measuring from $A$ to the endpoint mass $m$ and $B$ is the distance measuring from $x_2$ to the end point mass $m$. The position $x_1$ of the point $A$ is given by

$$x_1 = x + l_1\theta.$$

By taking the Laplace transform we get the transfer function from actuator $Q$ to sensor $A$ as

$$\frac{\tilde{X}_1(s)}{\tilde{F}(s)} = \frac{Z_{11}(s)}{R_1(s)}, \qquad (5.27)$$

where

$$
\begin{aligned}
Z_{11}(s) \quad = \quad & [I_{cm}(c_1 + c_2) + l_1(M + m)(c_1 l_1 - c_2 l_2)]\, s^3 \\
& +[I_{cm}(k_1 + k_2) + c_1 c_2 (l_1 + l_2)^2 \\
& \quad +l_1(k_1 l_1 - k_2 l_2)(M + m)] s^2 \\
& + (c_1 k_2 + c_2 k_1)(l_1 + l_2)^2 s \\
& + k_1 k_2 (l_1 + l_2)^2, \text{ and} \\
R_1(s) \quad = \quad & I_{cm}(M + m) m_a s^6 + [I_{cm}(c_1 + c_2)(M + m + m_x) \\
& + (c_1 l_1^2 + c_2 l_2^2)(M + m) m_a + I_{cm} c(M + m)\big]\, s^5 \\
& + \big[(M + m + m_x)\{c_1 c_2 (l_1 + l_2)^2 + I_{cm}(k_1 + k_2)\} \\
& + \ (k_1 l_1^2 + k_2 l_2^2)(M + m) m_a + c(c_1 + c_2) I_{cm} \\
& +c(c_1 l_1^2 + c_2 l_2^2)(M + m)\big]\, s^4 \\
& + \big[(l_1 + l_2)^2 (c_1 k_2 + c_2 k_1)(M + m + m_x) \\
& + c\{I_{cm}(k_1 + k_2) \\
& + (k_1 l_1^2 + k_2 l_2^2)(M + m) + c_1 c_2 (l_1 + l_2)^2\}\big]\, s^3 \\
& + [k_1 k_2 (l_1 + l_2)^2 (M + m + m_x) \\
& \quad +c(c_1 k_2 + c_2 k_1)(l_1 + l_2)^2] s^2 \\
& + c k_1 k_2 (l_1 + l_2)^2 s.
\end{aligned}
\tag{5.28}
$$

### 5.3.1.3 Simulation of the open loop plant

The values of the various parameters of the model are tabulated in Table 5.3. We are interested in examining the effect of variation of the location of the mass $m$ in the $y$-direction on our model. Figures 5.27 and 5.28 show the movement of poles and zeros of the plant as well as the bode plot of the plant as the location of the mass $m$ is varied from the bottom ($x = 0$) to the top ($x = 2l$). From Figure 5.27, we find that the second pair of poles and a pair of transmission zeros move together as the location varies from bottom to top, and they almost cancel one another as well. A similar effect is also observed in Figure 5.28. However, interesting point here is that at low frequency range the plant dynamics is independent of the mass location since we have the high order pole-zero cancellation. Even if we assume all the dynamics are active, i.e., the original five dynamics equations, there exist a range they can be treated independently and separately, i.e., as a SISO system instead of as a MIMO system. This is specially true when the endpoint mass, $m$ is closer to the center of the x-axis. Within the vicinity of this range, we observe the position-dependent pole-zero cancellation and domination of four poles and one zero position-independent dynamics even at higher frequencies range.

Based on this simulation, we see that we can train each axis independently at its nominal position

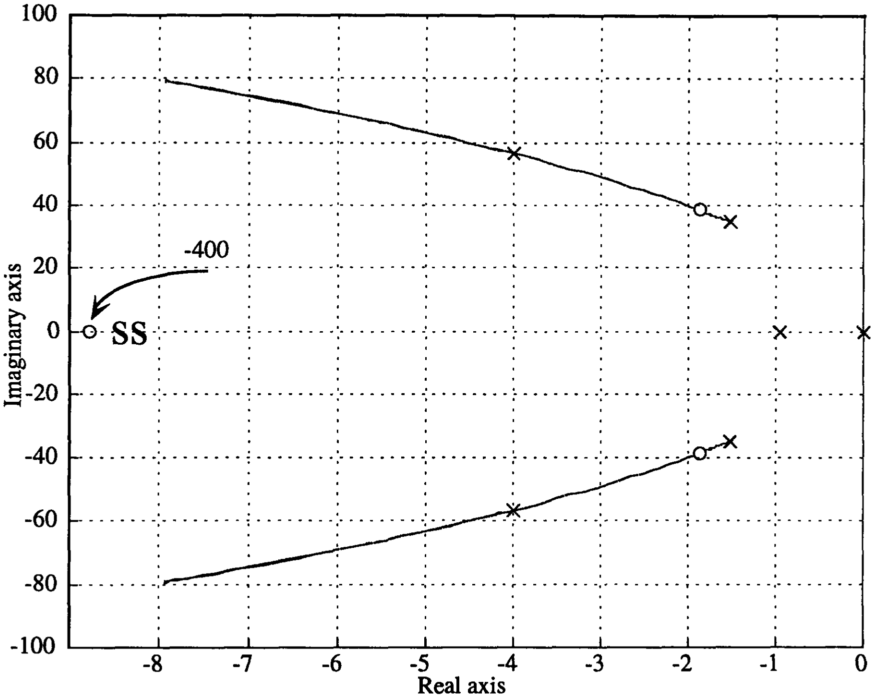


Figure 5.27: Pole-zero plot of the CMM dynamics

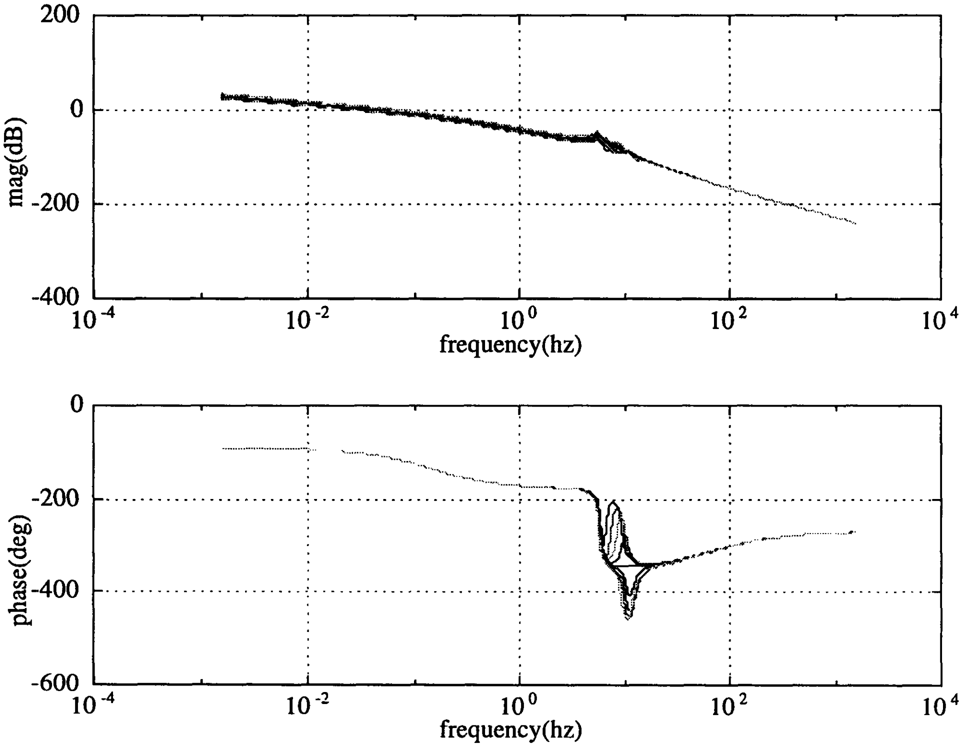


Figure 5.28: Bode plot of the CMM dynamics

Table 5.3: Parameter Values Used for the CMM

| | | | |
|---|---|---|---|
| $x_a$ | 0.2 m | $c$ | 50 N-sec/m |
| $l$ | 1.0 m | $c_1$ | 50 N-sec/m |
| $M$ | 200 Kg | $c_2$ | 45 N-sec/m |
| $m$ | 40 Kg | $k_1$ | 2000 N/m |
| $m_x$ | 10 Kg | $k_2$ | 2000 N/m |
| $a$ | 0.4 m | $k$ | 5000 N/m |

at low speed where this multiple DOF system behaves as separated independent SISO sub-systems. Thus, the number of the parameters to be trained can be reduced significantly so that it is possible for actual implementation of such complex multi-degree system. We can add the adaptive gains to compensate the cross-coupling after the independent tuning is completed. Due to this MIMO conversion to the SISO at the nominal position, we can effectively train a complex MIMO system with much lesser parameters more efficiently.

### 5.3.2 Experiments

#### 5.3.2.1 Plant and Control Structure

Based on the model we obtained earlier and because of the PI inner loop in the servo amplifier as described earlier, the structure of the plant after combining the inner control loop to the system, the system is a fourth-order system with relative degree of 3.

The estimated plant equations for both x-axis and y-axis are obtained from the experimental bode-plots. The plant transfer equation for the x-axis is shown in Eqn.(5.29) and that for the y-axis is shown in Eqn.(5.30). Figure 5.29 shows the final computer-based NC controller based on *progressive learning* and MRAC adaptive control.

$$\frac{\dot{x}_a}{\dot{x}_r} = \frac{Kp_x\ (S + b_x)}{S^4\ +\ a_{x_3}S^3\ +\ a_{x_2}S^2\ +\ a_{x_1}S\ +\ ax_0} \tag{5.29}$$

$$\frac{\dot{y}_a}{\dot{y}_r} = \frac{Kp_y\ (S + b_y)}{S^4\ +\ a_{y_3}S^3\ +\ a_{y_2}S^2\ +\ a_{y_1}S\ +\ ay_0} \tag{5.30}$$

The description of the above plant is under the assumption of single axis operation. When we extend to the multi-axes movement, cross-coupling effect needs to be incorporated. Instead of full MIMO system based on the simulation and modeling described earlier, a simplified MIMO system is considered and implemented. Here, we treat the input from the other axis as a disturbance to the system. The modified input is shown in Eqn.(5.31).

$$\hat{u}_{tot} = \hat{u}_c + \hat{u}_o, \tag{5.31}$$

where $\hat{u}_{tot}$ represents the final total input for each axis, $\hat{u}_c$ represents the estimated input with the single axis assumption, and $\hat{u}_o$ represents the estimated input results from the cross-coupling effect. Further,

$$\hat{u}_{tot} = \theta^T \omega \tag{5.32}$$

where $\theta = [\theta_c^T \quad \theta_o^T]^T$ and regressor, $\omega = [\omega_c^T \quad \omega_o^T]^T$. $\theta_c$ is the control parameters based on the SISO MRAC formulation and the $\theta_o$ is the control parameters involve in the cross-coupling dynamics. $\omega_c$ is the internal dynamics of the current axis and $\omega_o$ is the internal dynamics from other axes. It can also be seen in a block diagram form as shown in Figure 5.30. Therefore, there are 32 parameters for the overall MIMO system based on the MRAC formulation. It is almost impossible to train all of them and guarantee stability for the reasons given earlier. By taking into account the

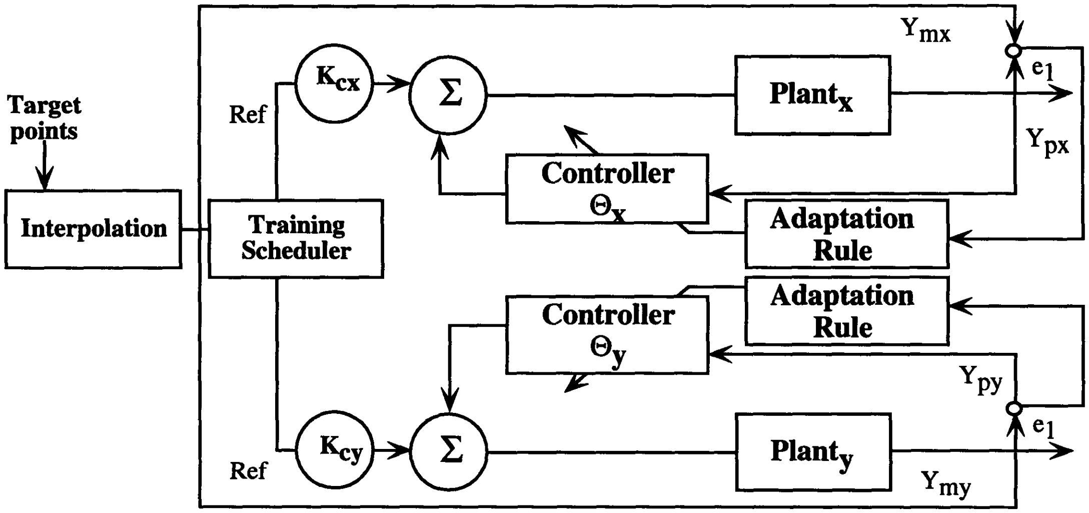


Figure 5.29: Block-diagram for the single axis under MRAC control

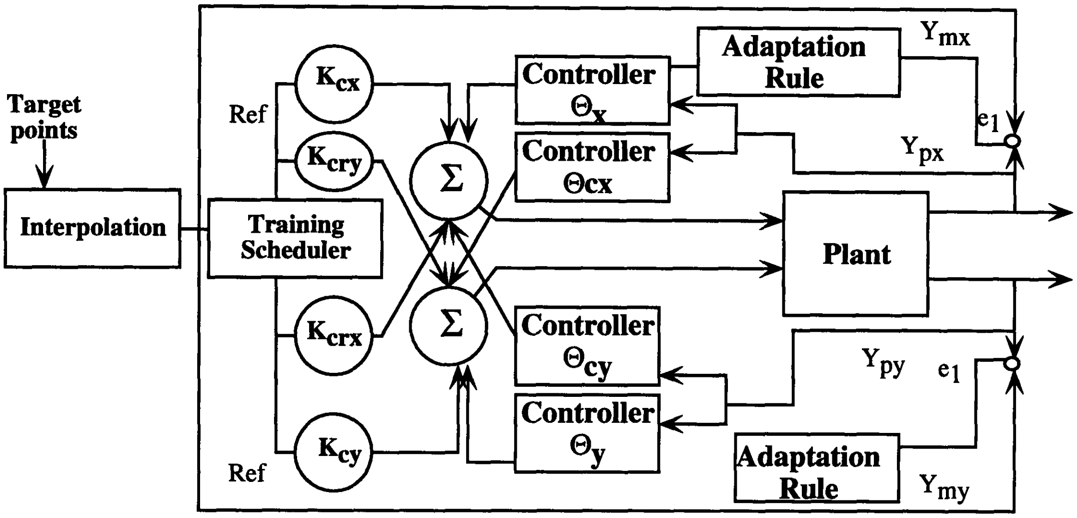


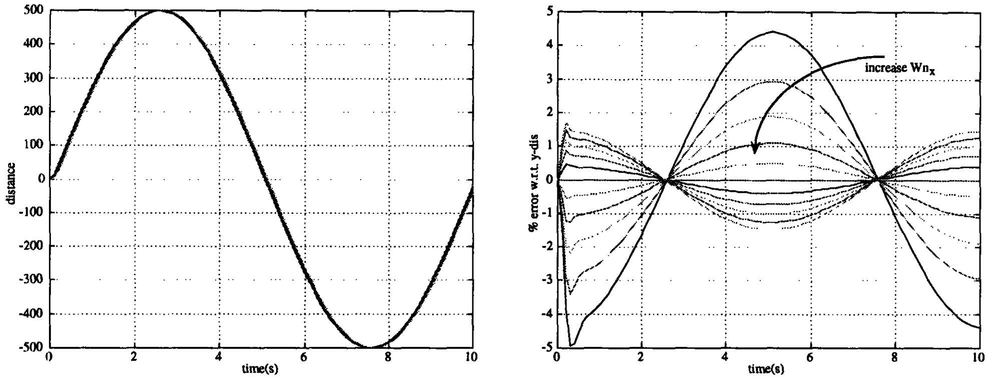
Figure 5.30: Block-diagram for the dual axes movement of CMM-XYZ under MRAC control

simplification we made earlier about converting the MIMO into a SISO system at nominal position and low speed, we can train the 8 parameters relating to the dynamics of its axis for each axis separately before we train the final 16 cross-coupling effect parameters. Next we are going to introduce the reference model before we show the tuning results.

#### 5.3.2.2 Reference Model

Based on the plant structure and the required performance criteria, a reference model is formulated. Before we start introducing the reference model we used, we would like to introduce the definition of the position error in our case here. During the dual-axes movement, the resultant error is caused by the differential movements between each axis, or the relative position difference. Thus, if the dynamics of each axis is different, then each axial output will arrive to its target point for the same given set of inputs at different times, thereby giving rise to a position error. To demonstrate the effect, we simulate a two-DOF system that has a different second-order dynamics for each axis and subject to the same sinusoid input as shown in Figure 5.31 and Figure 5.32 where the nature frequency of the x-axis dynamics varies from 50% to 150% of the y-axis dynamics. Due to the fact that the input to each axis is the same, the temporal outputs, or the geometric outputs, should be a straight 45° line; however, as shown in Figure 5.32(a), the outputs for various x-axis dynamics result in an elliptical shape. Figure 5.31(b) shows the relative error at each given instant and

Figure 5.32(b) shows the maximum percentage relative error for various x-axis dynamics.



(a) X-axis output

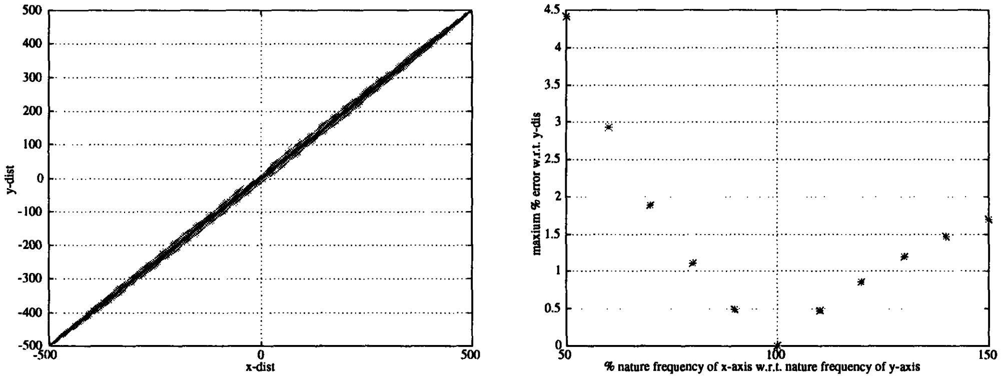
(b) % of relative position error

Figure 5.31: Relative position error resulted for system with different dynamics for each axis

Based on this simulation, we can clearly see that it is important for each axis of the multiple DOF system to have the same closed loop dynamics to minimize the relative position error.

To meet these requirements, we chose the following fourth-order reference model:

$$W_{(m)} = \frac{3.36 \times 10^7 (s + 147)}{(s^4 + 537s^3 + 1.1 \times 10^5 s^2 + 1.0 \times 10^7 s + 3.68 \times 10^8)}. \tag{5.33}$$



(a) X-axis and Y-axis outputs

(b) Maximum % of relative position error

Figure 5.32: Maximum relative position error resulted for system with different dynamics for each axis

The transfer function for the X-axis is

$$W_{p_x} = \frac{5.72 \times 10^7(s+147)}{s^4 + 159.7s^3 + 56205s^2 + 1.45 \times 10^6 s + 8.677 \times 10^7} \tag{5.34}$$

The transfer function for the Y-axis is

$$W_{p_y} = \frac{5.5465 \times 10^7(s+128.15)}{s^4 + 145.2974s^3 + 59764s^2 + 1.75 \times 10^6 s + 1.22 \times 10^8} \tag{5.35}$$

The pre-determined feedback component $\Delta(s)$ is chosen for each of the plant approximations as:

$$\Delta = s^3 + 162s^2 + 2255s^3 + 7350 \tag{5.36}$$

Based on the above transfer functions, the reference model characteristic equations are obtained as:

$$\begin{aligned} \Phi_{m_x}(s) &= s^7 + 699s^6 + 200 \times 10^5 s^5 + 2.9 \times 10^7 s^4 \\ &\quad +(0.23s^3 + 8.3s^2 + 90.6s + 270.5) \times 10^{10} \\ \Phi_{m_y}(s) &= s^7 + 680s^6 + 1.89 \times 10^5 s^5 + 2.7 \times 10^7 s^4 \\ &\quad +(0.21s^3 + 7.4s^2 + 79.2s + 235.9) \times 10^{10} \end{aligned} \tag{5.37}$$

The closed-loop plant characteristic equation is obtained for the MRAC in terms of control parameters as follows:

$$\begin{aligned} \Phi_p(s) &= \left[\Delta_(s) - (\theta_{11} + \theta_{12}s + \theta_{13}s^2)\right] R_p(s) \\ &\quad -k_p Z_p \left[(\theta_{21} + \theta_{22}s + \theta_{23}s^2) + \Delta(s)\theta_0\right] \end{aligned} \tag{5.38}$$

where the control parameter vector for the fourth-order approximation plant is defined as:

$$\theta \overset{\text{def}}{=} [\theta_{11},\ \theta_{12},\ \theta_{13},\ \theta_{21},\ \theta_{22},\ \theta_{23},\ \theta_0,\ k_c]^T \tag{5.39}$$

The tuned parameters based on the estimated 4th-order system parameters for the x-axis are: $k_{x_t} = 1.287$, $\theta_{1_{xt}} = [21.4882 \; - 55.49 \; - 37.75]^T$, $\theta_{2_{xt}} = [-17.29\, 36.74 \; - 30.34]^T$, $\theta_{0_{xt}} = 0.2$ and for the y-axis are: $k_{y_t} = 1.327$, $\theta_{1_{y_t}} = [32.1 \; - 49.95 \; - 37.31]^T$, $\theta_{2_{y_t}} = [-15.94 \; - 16.13 \; - 27.58]^T$, $\theta_{0_{y_t}} = 0.197$

The feedforward gain $k_{c_x}$ and $k_{c_y}$ were all initialized to 1 whereas the other control parameters were initialized to zeros. Thus, the closed-loop characteristic polynomial is defined as follows:

$$\begin{aligned} W_{\theta_x}(0) &= s^7 + 307.29s^6 + 8.55 \times 10^4 s^5 \\ &+ (0.0012s^4 + 0.05464s^3 + 2.418s^2 + 28.82s + 87.9) \times 10^{10} \end{aligned} \qquad (5.40)$$

and

$$\begin{aligned} W_{\theta_y}(0) &= s^7 + 321.72s^6 + 8.43 \times 10^4 s^5 \\ &+ (0.0011s^4 + 0.045s^3 + 1.775s^2 + 20.64s + 63.78) \times 10^{10} \end{aligned} \qquad (5.41)$$

### 5.3.2.3 Learning Procedure

The desired trajectory as shown in Figure 5.33 is generated using trajectory synthesis based on the following set of target points and is given in Table 5.3.2.3. During the training, the speed of the desired trajectory generated from the target points as presented earlier increases, thus increases

Table 5.4: Desired Target Points Used in the Trajectory

| Pt. No. | (x,y) | Pt. No. | (x,y) |
|---|---|---|---|
| 1 | (0,0) | 2 | (69,61) |
| 3 | (122,97) | 4 | (167,91) |
| 5 | (193,47) | 6 | (199,-16) |
| 7 | (183,-73) | 8 | (147,-99) |
| 9 | (95,-83) | 10 | (32.9,-32) |
| 11 | (-32.9,32) | 12 | (-95,83) |
| 13 | (-147,99) | 14 | (-183,73) |
| 15 | (-199,16) | 16 | (-193,-47) |
| 17 | (-167,-91) | 18 | (-122,-97) |
| 19 | (-64,-61) | 20 | (0,0) |

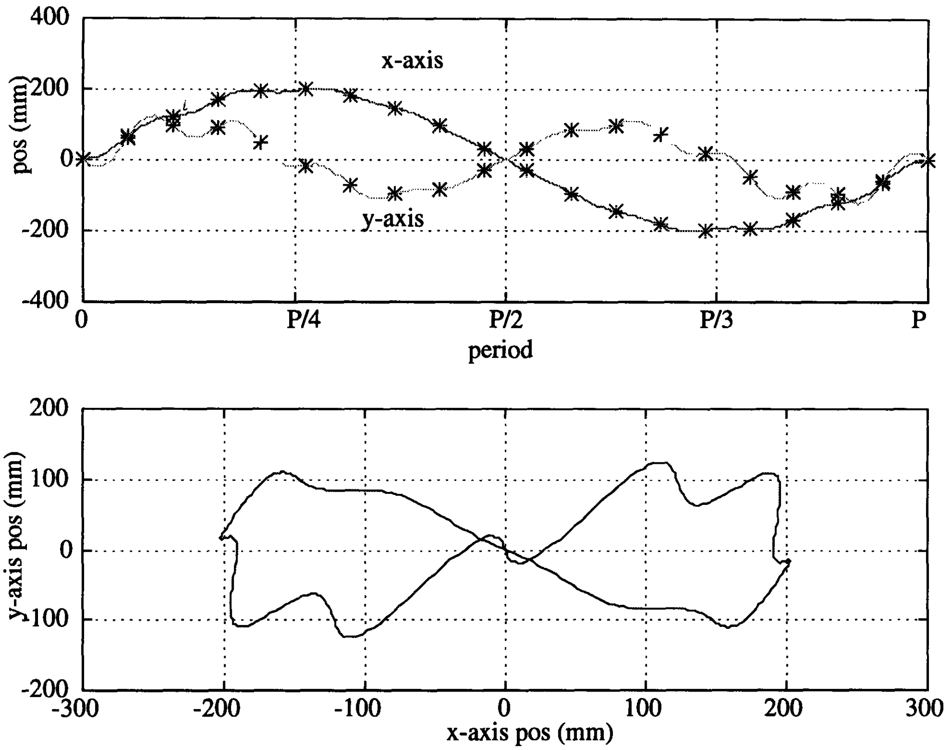


Figure 5.33: Trajectory Profile

the excited frequency bandwidth. Here $10s/P$ is used to denote that each period is completed in 10 sec. Figure 5.34 shows the excitation frequency bandwidth for various speeds.

### 5.3.3 Results and Discussions for the Tuning of Each Axis Independently

Figures 5.35 to Figure 5.38 show the results for experiments that we described earlier where the x-axis or y-axis are operating with the other axis being locked at its nominal position.

Figure 5.35(a) shows the parameter history plot, Figure 5.35(b) shows the error plot, Figure 5.36(a) shows the phase plot, and Figure 5.36(b) plot shows the pole-zero plot of the x-axis with the y-axis locked at its nominal position, or the center position. Figure 5.37(a) shows the parameter history plot, Figure 5.37(b) shows the error plot, Figure 5.38(a) shows the phase plot, and Figure 5.38(b) plot shows the pole-zero plot of the y-axis while with x-axis locked at its nominal position, or the center position. The obtained nominal parameters for the X-axis and Y-axis are shown in Table 5.3.3.

After this is obtained, we run tests to include the cross-coupling parameters. Since they are po-

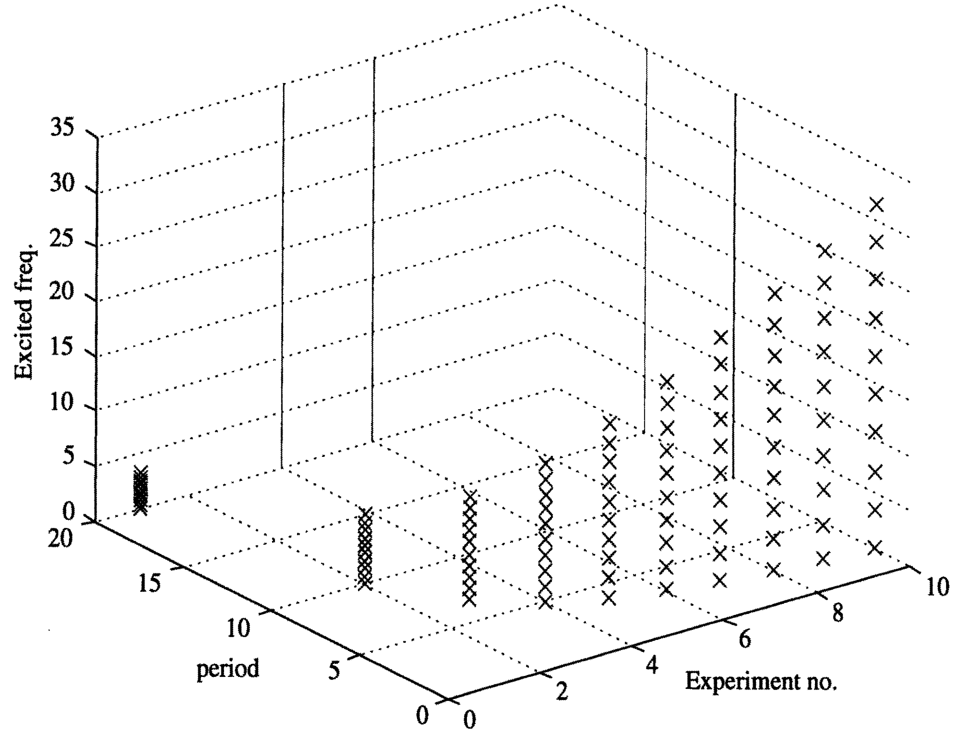
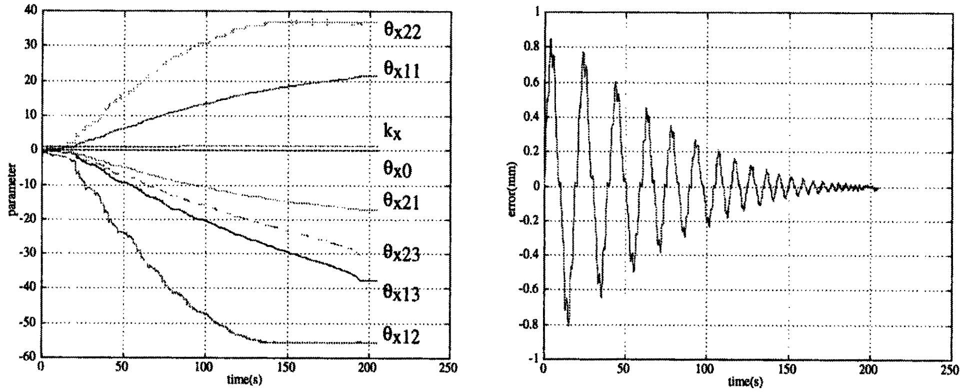


Figure 5.34: Excited frequency bandwidth for various speeds

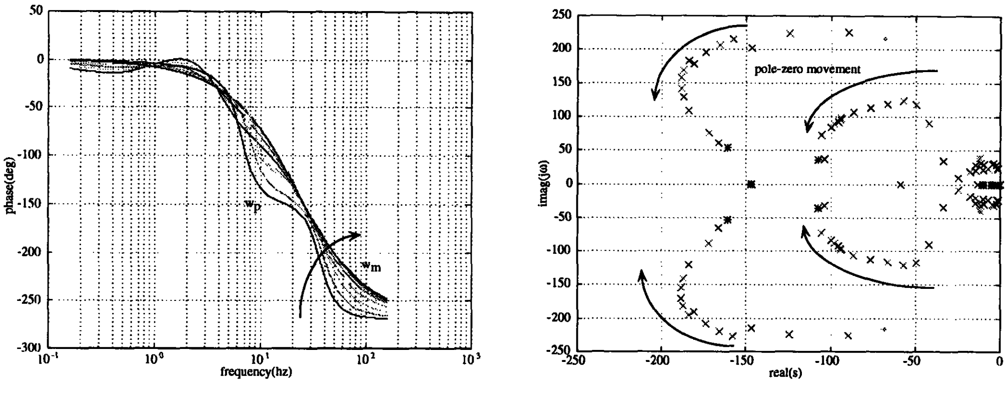


(a) Control parameter

(b) Error plot

Figure 5.35: X-axis tuning results (parameter and error plot)

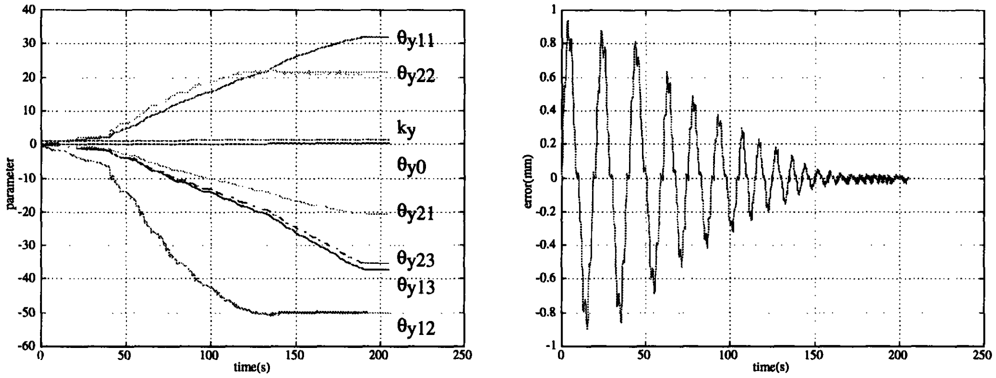
sition dependent, they do not have any fixed value as the nominal motion does. Thus we chose not to show the results of the tuning here but will show later in the validation test. Neverthelss, based on these results, we clearly see the effectiveness of *progressive learning* in tuning of complicated MIMO systems even using the simplified tuning method. More importantly, once the system is tuned, the results are task-independent which means that it will give the same relative performance



(a) Phase plot

(b) Pole-zero plot

Figure 5.36: X-axis tuning results (phase and pole-zero plot)



(a) Control parameter

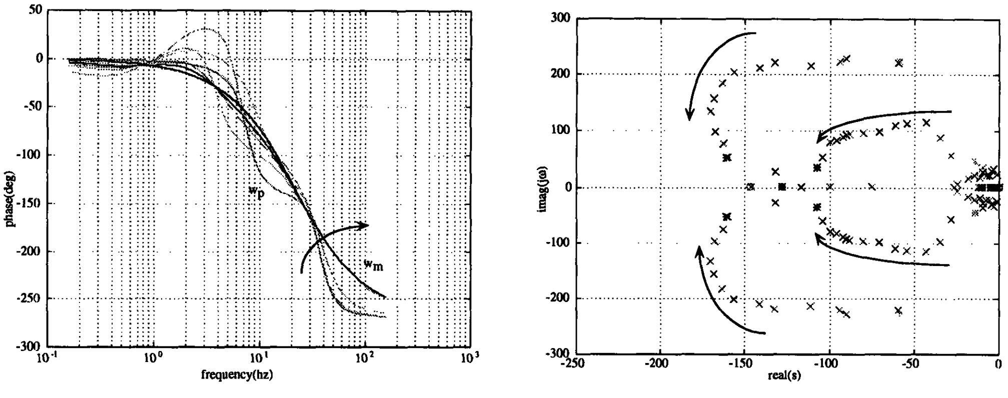
(b) Error plot

Figure 5.37: Y-axis tuning results (parameter and error plot)

regardless of the tasks. Most important of all, the final control scheme shows the computerized unified approach to the NC/servo controller and the synthesized desired trajectory. Next, we are going to validate the tuning results that include the cross-coupling effects on various test requirements that are different from the tuning task to demonstrate this "task-independence".

## 5.3.4 Validation Tests and Results Based on the Dual-axes Tuning

In order to validate the tuning results for the complete MIMO system, we ran a validation test. The test requires the endpoint to travel diagonally 50cm and to follow a trapezoidal velocity



(a) Phase plot

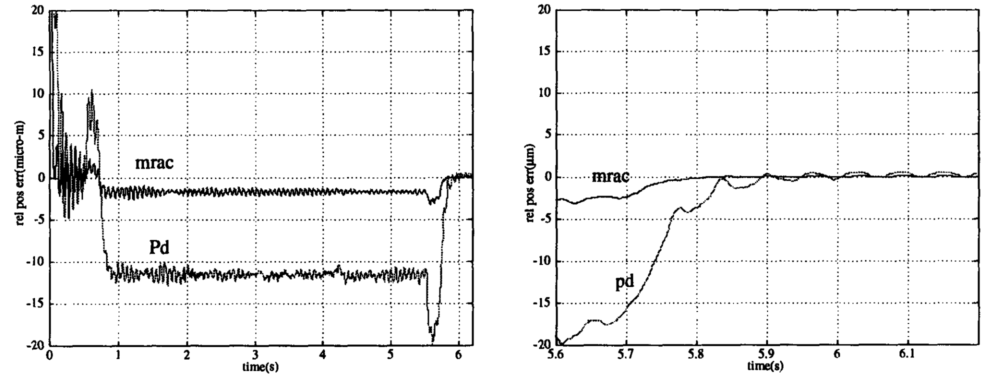
(b) Pole-zero plot

Figure 5.38: Y-axis tuning results (phase and pole-zero plot)

Table 5.5: Obtained Nominal Parameters for X- and Y-axis

| | | | |
|---|---|---|---|
| $\theta_{11_x}$ | 20.99 | $\theta_{11_y}$ | 32.0 |
| $\theta_{12_x}$ | -55.49 | $\theta_{12_y}$ | -49.95 |
| $\theta_{13_x}$ | -36.15 | $\theta_{13_y}$ | -37.2 |
| $\theta_{21_x}$ | -16.75 | $\theta_{21_y}$ | -15.94 |
| $\theta_{22_x}$ | 36.74 | $\theta_{22_y}$ | -16.7 |
| $\theta_{23_x}$ | -29.06 | $\theta_{23_y}$ | -27.58 |
| $\theta_{0_x}$ | 0.2 | $\theta_{23_y}$ | 0.197 |
| $k_{c_x}$ | 1.287 | $k_{c_y}$ | 1.327 |

profile. In the test, we mounted accelerometers both in the direction of the motion and perpendicular to the motion to monitor the cross-coupling effects, even though our tuning results are based on the sensor information from linear encoders along the x- and y- rails. The test results are compared to the improved PD controller as opposed to the P controller originally installed. The tests are repeated for two types of accelerations and speeds. Figure 5.39 and Figure 5.40 show results based on constant acceleration in the beginning and deceleration in the end at $5kgm/s^2$ for 0.1 second and constant velocity of $0.1m/s$ or $6m/min$ during the intermediate time. Figure 5.41 and Figure 5.42 show results based on constant acceleration in the beginning and deceleration in the end at $4kgm/s^2$ for 0.2 second and constant velocity of $0.2m/s$ or $12m/min$ during the intermediate time. Clearly, we see that all the results shows a great improvement as compared to the original $4m/min$ scanning speed with ± $25\mu$m accuracy. The tuned system using *progressive learning* and simplified MRAC shows even better improvement as shown in Figure 5.40 and Figure 5.42 since the idea of using the cross-coupling parameters is to decouple the motion for the off-axis dynamics which can be observed in term of minimal acceleration from the accelerometer mounted perpendicular to the motion. The accuracy we can achieved is within $8\mu$m for the fast motion and within $5\mu$m for the slower motion. Both cases show a greater improvement as compared to the old scanning speed and accuracy. The major cause of the jump in the beginning and the end of the motion is due to the use of the trapezoidal velocity profile. Since there is acceleration discontinuity present at each corner of the trapezoidal velocity profile, we will expect a jump in acceleration measurement. However, by correctly tuned the system, we can minimize the jump in acceleration.

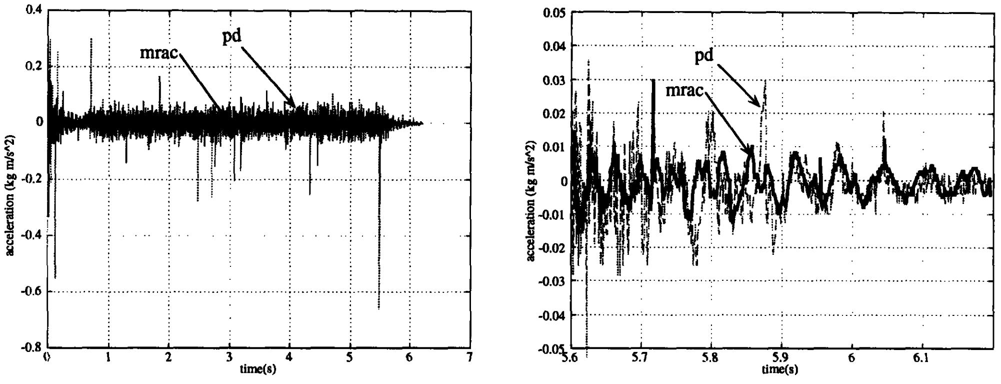


(a) Full motion

(b) Steady state

Figure 5.39: Relative position error at half speed

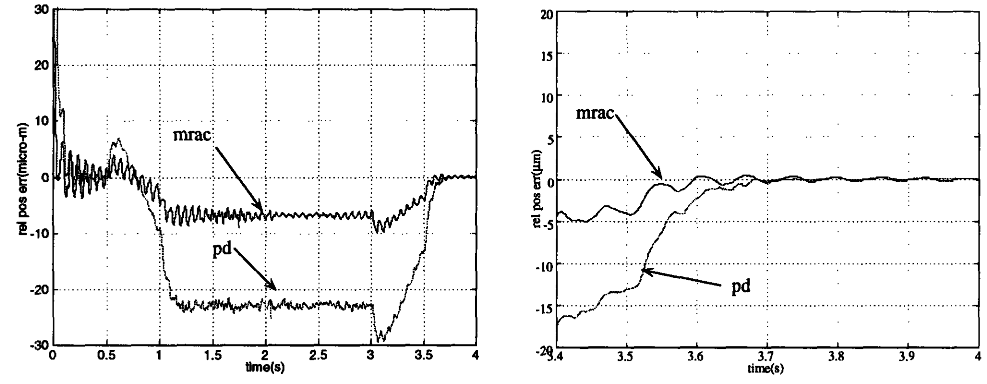
As shown in these validation test results based on the proposed dual-axis MIMO tuning method, we can see that through the use of the MIMO high-order controller we can control the dual axis motion of the CMM at much higher speed and accuracy. This also validates the assumption we made based on the simulation that the dual axis MIMO system we have here can be divided into SISO subsystem and cross-coupled subsystem and train each one of them separately. Thus, in fact we have successfully taken a first step of applying a high-order adaptive control with a large number of the parameters that need to be tuned to a real physical system. Through our modification of the MIMO tuning method by the way of the *extended progressive learning* we have in fact "progressively" stable tuned each sub-set of the total parameters at a given time to complete once impossible



(a) Full motion

(b) Steady state

Figure 5.40: Cross acceleration at half speed



(a) Full motion

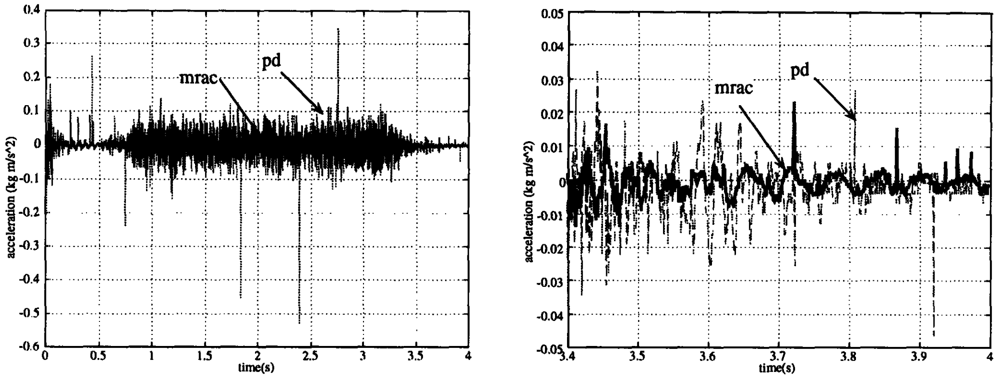
(b) Steady state

Figure 5.41: Relative position error at full speed

on-line tuning of a large number of the control parameter space.

### 5.3.5 Conclusion

In this section, we have shown through experiments that the *extendedprogressive learning* is a better method for tuning a higher order, non-collocated multiple DOF robotic system compared to the conventional method. The main idea of the *extended progressive learning* is to excite the system gradually in accordance with the progress of the adaptation. By incorporating a trajectory synthesis, we developed a method of generating a series of tracking trajectories that satisfy the sta-



(a) Full motion

(b) Steady state

Figure 5.42: Cross acceleration at full speed

bility conditions of progressive learning as well as reflect the dynamics of the system. Together, we have shown a new unified approach to replace the conventional separate NC/servo controller method. Also, we have extended the *extended progressive learning* to "progressively" tune a multiple DOF system by tuning first its dominating single axis before considering the cross-coupling effect to complete tuning of a MIMO MRAC controller which was an impossible task to do for a real complicated physical system and implemented it successfully.

# Chapter 6

# Conclusion and Future Works

In this research, we have successfully developed an algorithm to perform on-line adaptive control and tuning of endpoint feedback controlled systems. In particular, we have developed the theoretical analysis of two newly added concepts to the original work of *progressive learning* into its final form named the *extended progressive learning*. The two newly added components are the *model augmentation* which we partake a complex dynamics system into stages and train each of them individually and sequentially according to its progress of improvement, and the *trajectory synthesis* which generates stable reference signal for the system based on the desired task. Together, a dynamics adjusted reference trajectory can be generated for stable tuning of the system and for vast operating conditions. With modification of controller complexity on-line, the algorithm can adapt to vast operating conditions and perform on-line tuning automatically. The algorithm offers the first implementable high-level controller that can self-adapt both set values and its own complexity for superior performance under vast operating conditions. In this research, we have shown the theory of the *extended progressive learning* and successfully implemented to three different industrial-related systems: a high-speed chip-placement machine, a linear slider, and a dual degree coordinate measurement machine. The results have shown and demonstrated the effectiveness of the newly developed *extended progressive learning* in improving the systems' performance and adaptation speed.

## Appendix A

# Suplemented Theorems for Robustness

Within this chapter, we restated the robustness theorems originally stated in [Sastry and Bodson, 1989] in dealing with the external sensored noises and unmodeled dynamics. For complete original reference, please refer to [Sastry and Bodson, 1989].

## A.1 Exponential Convergence and Robustness

In this section, we will define a the property of a perturbed system as

$$\dot{x} = f(t, x, u) \quad x(0) = x_0 \tag{A.1}$$

and its counterpart, the unperturbed system as

$$\dot{x} = f(t, x, 0) \quad x(0) = x_0 \tag{A.2}$$

where $t \geq 0$, $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$. Here, we shall restrict the solution of the $x$ and $u$ belonging to some arbitrary ball $b_h \in \mathbb{R}^n$ and $B_c \in \mathbb{R}^m$.

**Theorem 3** *Consider the system in Eqn.(A.1) and system in Eqn.(A.2). Let $x = 0$ be an equilibrium point of Eqn.(A.2), i.e., $f(t, 0, 0) = 0$, for all $t \geq 0$. Let $f$ be piecewise continuous in $t$ and have continuous and bounded first partial derivatives in $x$, for all $t \geq 0, x \in B_h, u \in B_c$. Let $f$ be Lipschitz in $u$, with Lipschitz constant $lu$, for all $t \geq 0, x \in B_h, u \in B_c$. Let $u \in L_\infty$.*

*Assume that there exist a Lyapunov function $v(t, x)$ for the system in Eqn.(A.2) satisfying the following inequalities*

$$\alpha_1 |x|^2 \leq v(t, x) \alpha_2 |x|^2 \tag{A.3}$$

$$\frac{dv(t, x)}{dt}|_{Eqn.(A.2)} \leq -\alpha_3 |x|^2 \tag{A.4}$$

$$|\frac{\partial v(t,x)}{\partial x}| \leq \alpha_4|x| \tag{A.5}$$

*Then there exists similar situation for the perturbed system stated in Eqn.(A.1):*

$$\begin{aligned} \frac{dv(t,x)}{dt}|_{Eqn.(A.1)} &= \frac{dv(x,t)}{dt}|_{Eqn.(A.2)} + \sum_{i=1}^{n} \frac{\partial v(t,x)}{\partial x_i} \cdot [f_i(t,x,u) - f_i(t,x,0)] && \text{(A.6)} \\ &\leq -\alpha_3|x|^2 + \alpha_4|x|l_u \parallel u \parallel_\infty && \text{(A.7)} \end{aligned}$$

*where*

$$\parallel u \parallel_\infty = \sup_{t \geq 0} \tag{A.8}$$

*and define*

$$\gamma_\infty \overset{\text{def}}{=} \frac{\alpha_4}{\alpha_3} l_u \left[\frac{\alpha_2}{\alpha_1}\right]^{\frac{1}{2}} \tag{A.9}$$

$$\delta \overset{\text{def}}{=} \gamma_\infty \parallel u \parallel_\infty \tag{A.10}$$

$$m \overset{\text{def}}{=} \left[\frac{\alpha_2}{\alpha_1}\right]^{\frac{1}{2}} \tag{A.11}$$

*If* $x = 0$ *is an exponentially stable equilibrium point of the unperturbed system*

*Then*

1. *The perturbed system is small-signal $L_\infty$-stable, that is there exists $\gamma_\infty, c_\infty > 0$, such that $\| u \|_\infty < c_\infty$ implies that*

$$\| x \|_\infty \leq \gamma_\infty \| u \|_\infty < h \tag{A.12}$$

*where*

$$\| x \|_\infty = \sup_{t \geq 0} |x(t)|, \tag{A.13}$$

*and $x$ is the solution to Eqn.(A.1) starting at $x_0 = 0$;*

2. *There exists $m \geq 1$ s.t., for all $|x_0| < h/m$, $0 < \| u \|_\infty < c_\infty$ implies that $x(t)$ converges to a $B_\delta$ ball of radius $\delta = \gamma_\infty \| u \|_\infty < h$, that is: for all $\epsilon > 0$, there exists $T \geq 0$ s.t.,*

$$|x(t)| \leq (1 + \epsilon)\delta \tag{A.14}$$

*for all $t \geq T$, along the solution of Eqn.(A.1) starting at $x_0$ and also for $t \geq 0$, $|x(t)| < h$.*

The detail proof is done in [Sastry and Bodson, 1989]. The implication of the Theorem 3 is that it relates internal exponential stability to external input/output stability. The term $-\alpha_3|x|^2$, where originates from the exponential stability of the unperturbed system, acts like a restoring force bringing the state vector back to the origin. The term $\alpha_4|x|l_u \| u \|_\infty$ acts like a disturbing force, pulling the state away from the origin. This is caused by the input $u$ where in this case is the disturbance acting on the system. While the first term is square in norm of $x$ and the second term is only proportional to the norm. the restoring term will equal the disturbing term when $|x| = \delta/m = \gamma_\infty/m \| u \|_\infty$. Though the size of the ball $B_h$ and ball $B_c$ vary depending on each of their choices and that the system satisfies a constant Lipschitz condition with a constant $l_u$, with varying $\gamma_\infty$, the $L_\infty$ gain, and the $c_\infty$, the stability margin, will vary but remain bounded. Also pointed out by [Sastry and Bodson, 1989] that when the rate of the convergence rate increase, the $L_\infty$ decreases, and the stability gain increases.

## A.2 Robustness to Ouput Disturbance

**Theorem 4** *Consider the direct adaptive control system stated earlier with measured output $y_p$ of the plant, where $n \in L_\infty$. Let $h > 0$.*

*If* $\quad w_m$ *is PE*

*Then* $\quad$ *there exists $\gamma_n, c_n > 0$, and $m \geq 1$, s.t. $\parallel n \parallel_\infty < c_n$ and $|x(0)| < h/m$ implies that $x(t)$ converges to a $B_\delta$ ball of radius $\delta = \gamma_n \parallel n \parallel_\infty$, with $|x(t)| \leq m|x_0| < h$ for all $t \geq 0$.*

Through the use of the Theorem 3 as a component of $u$ and then translates into $n$, we can proof the theorem. The detail proof can be found in [Sastry and Bodson, 1989].

## A.3 Robustness to Unmodeled Dynamics

**Theorem 5** *Considered the system described earlier that includes the unmodeled dynamics*

*If* $\quad w_m$ *is PE.*

*Then* $\quad$ *for $x_0, \gamma_a, \beta_a$ sufficiently small, the state trajectory of the adaptive system remain bounded.*

*where*

$$\gamma_a = \int_0^\infty |h_a(\tau)| d\tau, \tag{A.15}$$

*and $h_a(\tau)$ is the impulse response of $H_a$ and $\beta_a$ depends on the initial conditions in the unmodeled dynamics.*

To prove Theorem 5, Let $T > 0$ s.t. $x(t) \leq h$ for all $t \in [0, T]$. Define $n = H_a(u)$, so that, by assumption

$$\parallel n_t \parallel_\infty \leq \gamma_a \parallel u_t \parallel_\infty + \beta_a \tag{A.16}$$

for all $t \in [0, T]$. The input $u$ is shown below:

$$\begin{aligned} u &= \theta^T w = \theta^{*T} w + \phi^T w & \text{(A.17)} \\ &= \theta^{*T} w_m + \theta^{*T} Qe + \theta^{*T} q_n n + \phi^T w_m + \phi^T Qe + \phi^T q_n n & \text{(A.18)} \end{aligned}$$

Since $x \in B_h$, then there exist $\gamma_u$, $\beta_u \geq 0$ sufficiently smalll that

$$\gamma_a \gamma_u < 1 \tag{A.19}$$

$$\frac{\beta_a + \gamma_a \beta_u}{1 - \gamma_a \gamma_u} < c_n \tag{A.20}$$

where $c_n$ is defined in Theorem 4 and $\| \ n_t \ \|_\infty < c_n$. Thus, Theorem 4 implies that $|x(t)| < h$ for all $t \in [0, T]$. Since none of the constrains $\gamma_a$, $\beta_a$, $\gamma_n$, and $\beta_n$ is depending on $T$, $|x(t)| < h$ for all $t \geq 0$. If it is not ture, then $|x(t)| \leq h$ for all $t \in [0, T]$ and $x(T) = h$. Then the theorem would then apply and result in a contradiction since $|x(t)| < h$. $\diamond$.

## Appendix B

# Notation Used in the Thesis

In this chapter, we summarize all the variables used in the thesis from Table B-1 to Table B-6.

Table B.1: Constant

| Notation | Description |
|---|---|
| $n$ | Order of the system |
| $n_q$ | Order of the system at stage q |
| $n_n$ | Order of the system at stage n (overall) |
| $R_i$ | amplitude of the $i$th ref. exc. spectrum |
| $v_i$ | amplitude of the $i$th noise. spectrum |

Table B.2: Transfer Function

| Notation | Description |
|---|---|
| $W_p$ | True open-loop plant transfer function |
| $W_m$ | Desired system transfer function |
| $G_{m_q}$ | Estimated desired transfer function at stage q |
| $G_{p_q}$ | Estimated open loop transfer function of the system at stage q |
| $G_{r_q}$ | Residual open-loop transfer function at stage q |
| $G_{t_q}$ | Estimated tuned closed loop transfer function of the system at stage q |
| $F_{G_q}$ | Estimated closed loop transfer function of the system at stage q |
| $F_{p_q}$ | closed loop transfer function of the system at stage q |
| $F_{r_q}$ | Residual closed-loop transfer function at stage q |
| $F_{r_{t_q}}$ | Residual tuned closed-loop transfer function at stage q |
| $F_{t_q}$ | closed loop transfer function of the tuned system at stage q |
| $\Phi$ | characteristics polynomial |
| $k$ | gain of the transfer function |
| $Z$ | zero polynomial of the open loop trans. func. |
| $R$ | characteristics polynomial of the open-loop trans. func. |
| $*$, H | complex conjugate transpose |

Table B.3: MRAC

| Notation | Description |
|---|---|
| $\Delta_q$ | observer characteristic polynomial at stage q |
| $r$ | ref. |
| $u$ | system input |
| $y_p$ | measured system output |
| $y_m$ | desired system output |
| $w_1$ | input observer |
| $w_2$ | output observer |
| $e_1$ | output error |
| $w$ | regressor |
| $C_q$ | input feedback controller at stage q |
| $D_q$ | output feedback controller at stage q |
| $C_{t_q}$ | input feedback tuned controller at stage q |
| $D_{t_q}$ | output feedback tuned controller at stage q |

Table B.4: Control Parameters

| Notation | Description |
|---|---|
| $\theta$ | full-order control parameter $\in \mathbb{R}^{2n_n}$ |
| $\theta_q$ | control parameter $\in \mathbb{R}^{2n_q}$ at stage q |
| $\phi$ | parameter error |
| $\theta_{t_q}$ | tuned control parameter $\in \mathbb{R}^{2n_q}$ at stage q |
| $\theta_q^o$ | projected optimal control parameter $\in \mathbb{R}^{2n_q}$ at stage q |

Table B.5: Frequency

| Notation | Description |
|---|---|
| $\omega$ | frequency |
| $\omega_s$ | switching frequency |
| $\omega_H$ | maximum allowable stable exc. frequency |
| $\omega_i$ | excitation frequency at $i$th spectrum |
| $\varepsilon$ | increment in frequency of the matched phase region |

Table B.6: Phase Shift

| Notation | Description |
|---|---|
| $\angle\{\}.$ | phase of the argument |
| $\varphi.$ | phase shift angle |
| $\Gamma_{max}$ | maximum allowable phase |

# Appendix C

# Derivation of the Tuned Characteristics Polynomial

In this chapter, we show the derivation of the phase angle relation obtained from the stability condition to the measured phase angle output relation.

$$\begin{aligned} F_{t_q} &\overset{\text{def}}{=} \frac{k_{m_q} k_{r_q} Z_{m_q} Z_{r_q}}{R_{m_q} R_{r_q}} \tag{C.1} \\ &= \frac{k_p k_{t_q} Z_p \Delta_{0_q} Z_{m_q}}{(\Delta_q - C_t) R_p - k_p Z_p D_t} \tag{C.2} \end{aligned}$$

By equating Eqn. C.1 and Eqn. C.2, we can obtain the following relationship:

$$\begin{aligned} \Phi_{m_t} &\overset{\text{def}}{=} \frac{Z_p \Delta_{0_q} R_{m_q} R_{r_q}}{Z_{r_q}} \tag{C.3} \\ \Phi_{t_q} &\overset{\text{def}}{=} (\Delta_q - C_t) R_p - k_p Z_p D_t \tag{C.4} \\ k_{t_q} &= k_p k_c = k_{m_q} k_{r_q} \tag{C.5} \\ \Phi_{m_t} &= \Phi_{t_q} \tag{C.6} \end{aligned}$$

Then by simple algebraic manipulations, we obtained the following relationships:

$$\begin{aligned} \Phi_{t_q} &= \frac{Z_p \Delta_{0_q} R_{m_q} R_{r_{tq}}}{Z_{r_{tq}}} \tag{C.7} \\ &= \frac{k_p k_c Z_p \Delta_{0_q} Z_{m_q} R_{m_q} R_{r_{tq}}}{k_{m_q} k_{r_{tq}} Z_{m_q} Z_{r_{tq}}} \tag{C.8} \\ &= \left[ k_p k_c Z_p \Delta_{0_q} Z_{m_q} \right] \frac{R_{m_q}}{k_{m_q} Z_{m_q}} \frac{R_{r_{tq}}}{k_{r_{tq}} Z_{r_{tq}}} \tag{C.9} \\ &= \left[ k_p k_c Z_p \Delta_{0_q} Z_{m_q} \right] \frac{1}{G_{m_q} F_{r_{tq}}} \tag{C.10} \end{aligned}$$

Finally using result presented in Eqn. C.10, we obtain

$$\frac{\Phi_{t_q}}{\Phi_{p_q}} = \frac{k_p k_c Z_p \Delta_{0_q} Z_{m_q}}{\Phi_{p_q}} \frac{1}{G_{m_q} F_{r_{t_q}}} \tag{C.11}$$

$$= \frac{F_{p_q}}{G_{m_q} F_{r_{t_q}}} \tag{C.12}$$

The angle of $\frac{\Phi_{t_q}}{\Phi_{p_q}}$ can then be derived as

$$\angle\{\frac{\Phi_{t_q}}{\Phi_{p_q}}\} = \angle\{F_{p_q}\} - \angle\{G_{m_q}\} - \angle\{F_{r_{t_q}}\} \tag{C.13}$$

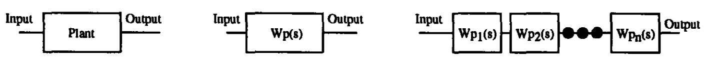
# Appendix D

# Model Reduction

In this section, we state underlining assumptions that we made when we model any real physical system. Due to the nature of the real physical system's behavior, unless we made some justifications, assumptions, and operating conditions, we can not obtain a "model" for the system. As the operating conditions expand, we usually change the assumed model of the system to a more complex model in practice. In practice, the relationship between the model complexity in terms of its order and operating conditions have been always stated in the form of the "rule of thumb" or widely practiced without any formal statements. Since major research here is to "progressively" learn the model structure, we need to formally establish these "rule of thumb" rules in the form of Proposition 1 and Proposition 2.

As shown in Figure D.1(a) and Figure D.1(b), we can always use lump parameter model to represent any physical single-input-single-output (SISO) linear time invarient (LTI) system. We are going to state the two propositions in order for the model augmentation of the progressive learning to be valid. Each proposition is given an valid example to illustrate the proposed proposition. Proposition 1 is used to validate the argumentation in an open-loop sense and Proposition 2 us used to validate the closed-loop model argumentation.

**Proposition 1** *For a physical system witha transfer function $W_p$ to represent the overall system and a simplified transfer function $W_{ps}$ to represent the reduced-order system, and a given set of $\phi_{max}$, the maximum allowable phase shift and $\rho_{max}$, the maximum allowable DC-gain, there exist an $\omega_s$*



(a) Plant

(b) Model

(c) Cascade lump-parameter model

Figure D.1: Various representation

*s.t.* $\exists \quad \omega_s$ *s.t.* $|arg\{Wp(j\omega)\} - arg\{W_{ps}(j\omega)\}| < \phi_{max} for \forall \quad \omega \in \{\omega | 0 \leq \omega \leq \omega_{w_s}\}$

*Illustration* For an given transfer function $W_p(s)$ which is a forth-order system, that has a form of two cascade second order transfer function, $W_{p_1}$ and $W_{p_2}$, can be correctly reduced into just a single second order system $W_{p_1}(s)$ in the frequency range $\omega_s$ as long as the ratio of the two nature frequencies are greater than $\left(\zeta_2\beta + \sqrt{\zeta_2^2\beta^2 + \phi_{max}^2\beta^2}\right)/\phi_{max}$ where $\beta$ is the ratio of the excitation frequency to the nature frequency of $W_{p_1}(s)$, $\zeta_2$ is the damping ratio of the $W_{p_2}(s)$.

*Proof*

For and given $W_p$ such that

$$W_p = \left(\frac{\omega_{n_1}^2}{s^2 + 2\zeta_1\omega_{n_1}s + \omega_{n_1}^2}\right)\left(\frac{\omega_{n_2}^2}{s^2 + 2\zeta_2\omega_{n_2}s + \omega_{n_2}^2}\right), \tag{D.1}$$

can be represented using two lumped parameter models $W_{p_1}$ and $W_{p_2}$ where

$$W_{p_1} = \left(\frac{\omega_{n_1}^2}{s^2 + 2\zeta_1\omega_{n_1}s + \omega_{n_1}^2}\right) \tag{D.2}$$

and

$$W_{p_2} = \left(\frac{\omega_{n_2}^2}{s^2 + 2\zeta_2\omega_{n_2}s + \omega_{n_2}^2}\right). \tag{D.3}$$

The phase angle of $W_p$, $\phi(W_P)$ at a given excitation frequency $\omega$ is

$$\phi\{W_p\} = -\arctan\{\frac{2\zeta_1\omega_{n_1}\omega}{\omega_{n_1}^2 - \omega^2}\} - \arctan\{\frac{2\zeta_2\omega_{n_2}\omega}{\omega_{n_2}^2 - \omega^2}\}. \tag{D.4}$$

By using the small angle approximation to approximate arctan in Eqn.(D.4) and the given maximum allowable phase shift, $\phi_{max}$, of the higher order transfer function at the frequency $\omega$, we can obtain the following relation:

$$\text{If } \phi\{W_{p_2}\} \leq \phi_{max}, \quad \text{then } \phi\{W_p\} \approx \phi\{W_{p_1}\} \text{ at the } \omega. \tag{D.5}$$

or

$$\frac{2\zeta_1\omega_{n_1}\omega}{\omega_{n_1}^2 - \omega^2} \leq \phi_{max} \tag{D.6}$$

Now, we further assume that $\omega_{n_2}$, the nature frequency of $W_{p_2}$ is

$$\omega_{n_2} = \alpha\omega_{n_1}, \tag{D.7}$$

and the excitation frequency $\omega$ is

$$\omega = \beta\omega_{n_1}, \tag{D.8}$$

then we can simplify Eqn.(D.6) into

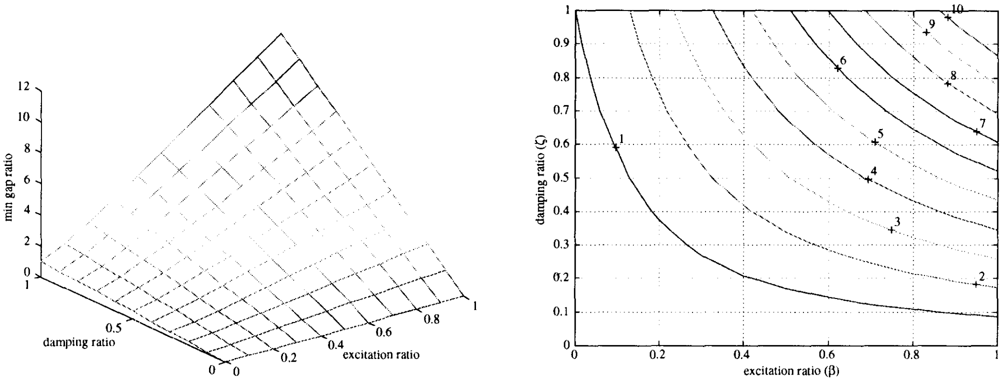
$$\frac{2\zeta_2\alpha\beta}{\alpha^2 - \beta^2} \leq \phi_{max}, \tag{D.9}$$

then Eqn.(D.9) can lead to Eqn.(D.10) as below:

$$\alpha \geq \frac{\zeta_2\beta + \sqrt{\zeta_2^2\beta^2 + \phi_{max}^2\beta^2}}{\phi_{max}} \tag{D.10}$$

Eqn.(D.10) establishes the lowest separation requirement between the nature frequencies of the two second order systems in order for the simplification to hold true in the open loop sense. Figure D.2(a) shows how this separation $\alpha$, in $z - axis$ varies with different damping ratio $\zeta_2$ and excitation frequency $\omega$ that represents in term of the ratio, $\beta$, of the first nature frequency and for a given maximum allowable phase lag $\phi_{max}$. Figure D.2(b) show the contour representation of Figure D.2(a). More importantly, what these two figures show besides just how far apart the two nature frequencies have to be in order for the simplification to work, but how high the excitation frequency we can excite the system for the simplified system representation to be valid even the separation of the two falls below the requirement. As demonstrated in Figure D.4(a) and Figure D.4(b), we see how the phase lag $\phi$ varies as we start to excite the system with different damping ratio for a given $\alpha$ that is clearly in violation of the separation requirement presented in Proposition 1. By re-deriving Eqn.(D.9) we can derive the maximum allowable excitation frequency ratio, $\beta_{max}$ that the higher order dynamics can be sufficiently ignored despite the low separation ratio:

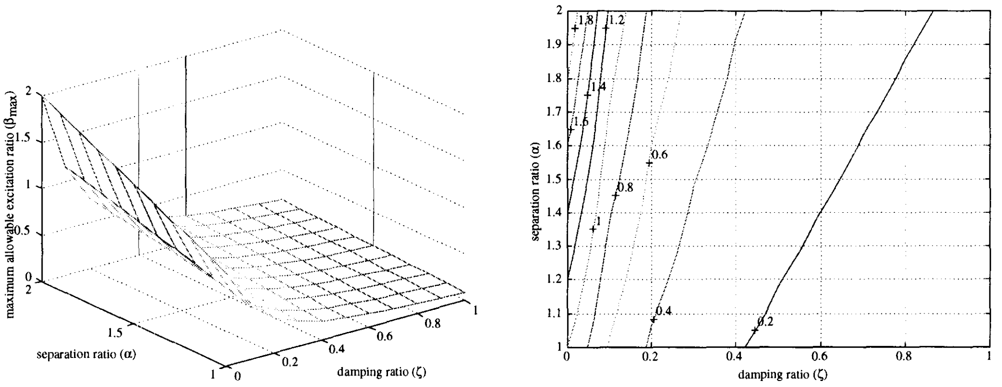
$$\beta \leq \frac{\sqrt{\zeta_2^2\alpha^2 + \phi_{max}^2\alpha^2} - \zeta_2\alpha}{\phi_{max}} \tag{D.11}$$



(a) Mesh

(b) Contour

Figure D.2: Minimal separation



(a) Mesh

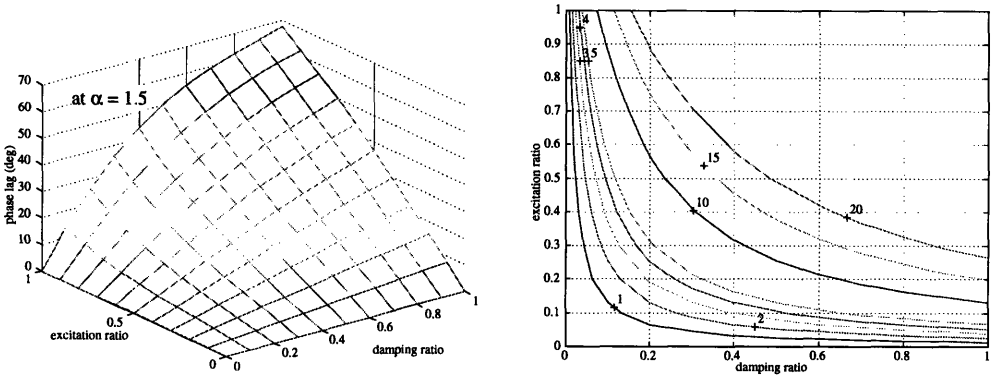
(b) Contour

Figure D.3: Maximum allowable excitation

This is also been validated using an example as shown in Figure D.3(a), the mesh representation, and Figure D.3(b), the contour map representation, for a given $\phi_{max} = 10^o$ there are region where Eqn.(D.11) holds though the separation ratio $\alpha$ is low as well.

In another example, we fixed the separation ratio, $\alpha$ to be 1.5, we again see that for a given damping ratio and excitation frequency there exists some region that the simplification still holds as shown in Figure D.4(a) and Figure D.4(b) the mesh and contour map representation respectively.

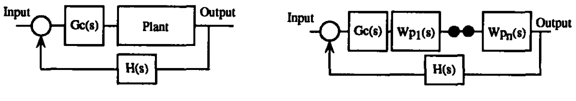
Next, we are going to use results from Proposition 1 to extend the finding of minimal bound for the open-loop transfer function to closed-loop transfer function. In general, based on the open-



(a) Mesh

(b) Contour

Figure D.4: Phase lag with varies excitation ratio and damping ratio



(a) Plant

(b) Lump-parameter model

Figure D.5: Various representation for the closed-loop system

loop system, $G_p(s)$ as represented in Figure D.1, Figure D.5(a) and Figure D.5(b) show the block diagram representations of the closed-loop system.

**Proposition 2** *For a physical system with transfer function $W_p$ to represent the full system and simplified transfer function $W_{ps}$ to represent the reduced-order system, together with a feedforward and/or feedback controller $G_c(s)$ and $H(s)$ respectively, for a given maximum allowable phase shift, $\phi_{max}$ and $\rho_{max}$, the maximum allowable amplitude ratio, and a region of frequencies, $\omega\forall \quad \omega \in \{\omega | 0 \leq \omega \leq \omega_{w_s}\}$, such that,*

$$\frac{G_c(j\omega)W_p(j\omega)}{1 + G_c(j\omega)W_p(j\omega)H_(j\omega)} \approx \frac{G_c(j\omega)W_p(j\omega)}{1 + G_c(j\omega)W_{ps}(j\omega)H_(j\omega)} \tag{D.12}$$

*Proof* For an given transfer function $W_p(s)$, with a feedforward controller, $G_c(s)$, and a feedback controller, $H(s)$, we can write its closed loop transfer function,$W_{p_{cl}}$ between its input and output as shown in Figure D.5(a) for the block diagram representation or as follow:

$$W_{p_{cl}} = \frac{G_c(j\omega)W_p(j\omega)}{1 + G_c(j\omega)W_p(j\omega)H(j\omega)} \tag{D.13}$$

If we can further assume the transfer function $W_p(s)$ can be represented in the cascade form based on Proposition 1, or $W_p(s) = \Pi_{i=1}^{n} W_{p_i}(s)$ where $n$ is the number of the cascaded transfer function then we can rewrite Eqn. D.13 as follow:

$$W_{p_{cl}}(s) = \frac{G_c(s)(\Pi_{i=1}^{n} W_{p_i}(s))}{1 + G_c(s)(\Pi_{i=1}^{n} W_{p_i}(s))H(s)} \tag{D.14}$$

Then first by separating $G_p(s)$ into two $j$th and $n - j$th transfer functions such that $W_p(s) = (\Pi_{i=1}^{j} W_{p_i}(s))(\Pi_{j+1}^{n} W_{p_i}(s))$, then we can always find an equivalent transfer function representation by performing block-diagram algebra to Eqn. D.14 or in Figure D.5(b) to a new equivalent form in Figure D.6 or below:

$$W_{p_{cl}(s)} = \frac{G_c(s)(\Pi_{i=1}^{j} W_{p_i}(s))}{1 + G_c(s)(\Pi_{i=1}^{j} W_{p_i}(s))H(s)} \times W'_{pr}(s), \tag{D.15}$$

where

$$\begin{aligned}
W'_{pr}(s) &= \frac{1 + G_c(s)(\Pi_{i=1}^{j} W_{p_i}(s))H(s)}{1 + G_c(s)(\Pi_{i=1}^{n} W_{p_i}(s))H(s)}(\Pi_{i=j+1}^{n} W_{p_i}(s)) && \text{(D.16)}\\
&= \frac{G_c(s)(\Pi_{i=1}^{j} W_{p_i}(s))}{1 + G_c(s)(\Pi_{i=1}^{j} W_{p_i}(s))H(s)} \times W_{pr}(s), && \text{(D.17)}\\
&= |W_{pr}(j\omega)| \exp^{-ang\{W_{pr}(j\omega)\}} && \text{(D.18)}
\end{aligned}$$

Then for any $\omega \in \{\omega | 0 \leq \omega \leq \omega_s\}$ s.t.

$$|W_{pr}(j\omega)| \exp^{-ang\{W_{pr}(j\omega)\}} \leq |\rho_{max}| \exp^{-ang\{\phi_{max}\}} \tag{D.19}$$

Then

$$W_{p_{cl}(s)} = \frac{G_c(s)(\Pi_{i=1}^{j} W_{p_i}(s))}{1 + G_c(s)(\Pi_{i=1}^{j} W_{p_i}(s))H(s)} \times W'_{pr}(s) \tag{D.20}$$

$$= \frac{G_c(s)W_{ps}}{1+G_c(s)W_{ps}(s)H(s)} \times W'_{pr}(s) \tag{D.21}$$

$$\begin{aligned} W_{p_{cl}(j\omega)} &= \frac{G_c(j\omega)(\Pi_{i=1}^{j} W_{p_i}(j\omega))}{1+G_c(j\omega)(\Pi_{i=1}^{j} W_{p_i}(j\omega))H(j\omega)} \times W'_{pr}(j\omega) && \text{(D.22)} \\ &= \frac{G_c(j\omega)W_{ps}}{1+G_c(j\omega)W_{ps}(j\omega)H(j\omega)} \times W'_{pr}(j\omega) && \text{(D.23)} \\ &\approx \frac{G_c(j\omega)(\Pi_{i=1}^{j} W_{p_i}(j\omega))}{1+G_c(j\omega)(\Pi_{i=1}^{j} W_{p_i}(j\omega))H(j\omega)} && \text{(D.24)} \end{aligned}$$

where $W_{ps} = \Pi_{i=1}^{j} W_{p_i}$, or the reduced-form transfer function, $W_{pr} = \Pi_{i=j+1}^{n} W_{p_i}$, which is the residual transfer function or the higher-order transfer function that is been ignored. Thus, based on the condition stated in Proposition 1, Proposition 2 is proved. ◇

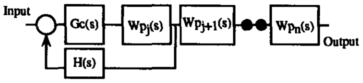
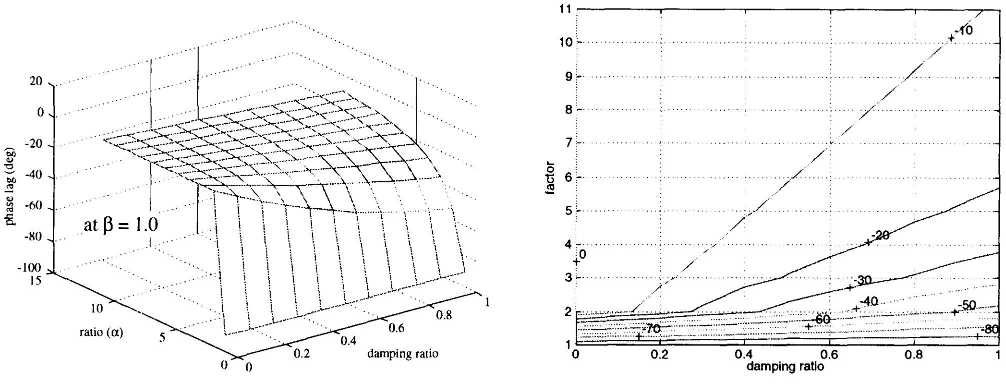


Figure D.6: Reduced-form closed loop transfer function representation

<u>Illustration</u> For an given transfer function $W_p(s)$ which is a forth-order system, and has a form of two cascade second order transfer functions, $W_{p_1}$ and $W_{p_2}$, it can be correctly reduced into just a single second order system $W_{p_1}(s)$ in the frequency range between $0 \leq \omega \leq \omega_s$ following criteria of Proposition 1. It has a feedforward controller that has a form of $G_c(s) = \frac{\theta_1}{\lambda}$ and a feedback controller $H(s) = \frac{\theta_0\lambda+\theta_2}{\lambda}$ where $\lambda$ is a first order transfer function. The value of $\theta_0, \theta_1, \theta_0$ are set to the values such that

$$\frac{G_c(s)W_{p_1}(s)}{1+G_c(s)W_{p_1}(s)H_(s)} = \frac{k_m\lambda}{s^2+2\zeta_m\omega_{n_m}s+\omega_{n_m}^2} \tag{D.25}$$

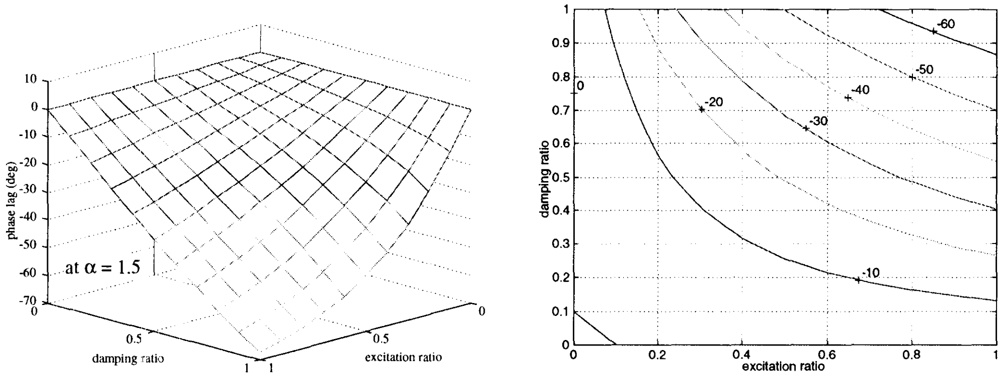
where $\zeta_m = 0.707, \omega_{n_m} = 1.5 \times \omega_{n_1}(1-\zeta_1^2)^{0.5}/(1-\zeta_m^2)^{0.5}$, $\zeta_1$ and $\omega_{n_1}$ are the damping ratio and nature frequency of $W_{p_1}(s)$. Figure D.7(a) and Figure D.7(b) show the mesh and contour map of how the phase lag varies by varying the damping ratio $\zeta_2$ and separation nature frequency ratio, $\alpha$ as defined before for the next second-order transfer function $W_{p_2}(s)$ when we excite at the nature frequency of the reduced transfer function. These two figures illustrate the existence of Proposition 2 that for a given larger separation ratio as obtained by Proposition 1 since the phase lag introduced



(a) Mesh

(b) Contour

Figure D.7: Extra phase lag by making the reduce-order assumption



(a) Mesh

(b) Contour

Figure D.8: Phase lag with varies excitation ratio and damping ratio

by the "residual transfer function", or the ignored high-order transfer function is less than the maximum allowable phase lag. Similar to the first illustration as done for the Proposition 1, we can still use the result from Proposition 2 even when the separation ratio is less than specified bound stated in Proposition 1 when the excitation frequency is within the range specified in Eqn.(D.11) as shown in Figure D.8(a) and Figure D.8(a).

These illustrations have established the model reduction propositions for both the open and closed loop proposition.

# References

[Acton, 1970] F. S. Acton, Numerical Methods That Work, Harper and Row, 1970

[Anderson, 1986] B. D. O. Anderson, et al., Stability of Adaptive Systems: Passivity and Averaging Analysis," MIT Press, Cambridge, MA 1986

[Anderson, et. al, 1986] B. D. Anderson, R. R. Bitmead, C. R. Johnsons, Jr., P. V. Kokotovic, R. L. Kosut, I. M. Mareels, L. Praly, and B. D. Riedle, Stability of Adaptive Systems: Passitivity 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] Karl Johan Åström and Björn Wittenmark, *Adaptive Control,* Addison-Wesley Publishing Company.

[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

[Bakker and Annaswammy, 1995] , R. Bakker and A. M. Annaswamy, *Stability and Robustness Properties of a Simple Adaptive Controller,* Accepted for publication, IEEE Trans. on Automatic Control, 1995

[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] 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

[Coppel, 1968] W. A. Coppel, "Dichotomies and Reducibility (II)", Journal of Differential Equations, vol. 4, No. 3, pp. 386-398, July 1968

[Desoer and Vidyasagar, 1975] C.A. Desoer and M. Vidyasagar, "Feedback Systems: Input-output Properties," Academic Press, 1975

[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

[Gawthrop, 1980] P.J. Gawthrop, "Hybrid self-tuning control." IEE Proceeding, 127:229-336, Part D, 1980

[Gullapalli, 1990] V. Gullapalli, "A Stochastic Reinforcement Learning Algorithm for Learning Real-Valued Functions", Neural Networks, Vol. 3, pp. 671-692, 1990

[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

[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

[Ichikawa, 1995] K. Ichikawa, "PAdptive Control of High Order Plants with Low Order Nominal Plants", Pg. 1102-1106, Proceeding of American Control Conference, Seattle, Washington, June 1996

[Ioannou and Kokotovic, 1983] P. A. Ioannou, P. V. Kokotovic, *Adaptive Systems with Reduced Models*, Springer Verlag, N.Y., 1983.

[Kreisselmeier and Narendra, 1982] G. Kreisselmeier and K. S. Narendra, "Stable Model Reference Adaptive Control in the Presence of Bounded disturbances," IEEE Trans. on Automatic Control, 27:1169-1175, Dec. 1982

[Kokotovic, et al., 1985] P. Kokotovic, et al., *On a Stability Criterion for Continuous Slow Adaptation,* System & Control Letters, vol. 6,pp.7-14, June 1985

[Jordan and Rumelhart, 1992] M. I. Jordan and D. E. Rumelhart, "Forward Models: Supervised Learning with a Distal Teacher", Cognitive Science, 16, 1992

[Kosut, et al., 1985] R.L. Kosut, et al., "Stability Theory for Adaptive Systems: Methods of Averaging and Persistency of Excitation", Proc. of 24th Conference on Decision and Control, Dec. 1985

[Kokotovic, 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

[Li, et al., 1994] Shih-Hung Li, Noriyuki Fujiwara, and Haruhiko Asada, *An Ultrahigh Speed Assembly Robot System: Part I. Design* Proc. IEEE 1994 Japan-USA Symposium on Flexible Automation, Japan, 1994

[Li and Asada, 1995] Shih-Hung Li and Haruhiko Asada, *Progressive Learning Part II: Trajectory Synthesis and Implementation,* Proc. ASME WAM 1995, CA.,1995

[Narendra and Annaswamy, 1986] K. S. Narendra and A. M. Annaswamy, "Robust Adaptive Control in the Presence of Bounded Disturbance," IEEE Trans. on Automatic Control, Vol. AC-31, no. 4, pp. 306-315, 1986

[Narendra and Annaswamy, 1987] K. S. Narendra and A. M. Annaswamy, "A new adaptive law for robust adaptive control without persistent excitation," IEEE Trans. on Automatic Control, Vol. AC-32,pp. 134:145, 1987..

[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

[Ogata, 1990] K. Ogata, Modern Control Engineering, 2nd. Ed. Prentice-Hall, Inc. N.J., 1990

[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

[Peterson and Narendra, 1982] B. B. Peterson and K.S. Narendra, "Bounded Error Adaptive Control," IEEE Trans. on Automatic Control, Vol. AC-27, no. 6, pp. 1161-1168, 1982

[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, L. Valvavani, M. Athan, and G. Stein, " Robustness of Adaptive Algorithms in the Presence of Unmodeled Dynamics," Proc. of 21st IEEE CDC, Florida, 1982

[Rohrs, et al., 1985] C. E. Rohrs, L. Valvavani, M. Athan, and G. Stein, " Robustness of Continuous-Time Adaptive Control Algorithms in the Presence of Unmodeled Dynamics," IEEE Trans. on Automatic Control, Vol. AC-30, no. 9, pp 881-889, 1985

[Sastry, 1984] S. Sastry, "Model-Reference Adaptive Control-Stability, Parameter Convergence, and Robustness," IMA Journal of Mathematical Control and Information, Vol. 1, pp. 27-66, 1984.

[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 Stearms, 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

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

[Wittenmark and Astrom, 1984] B. Wittenmark, and K. J. Astrom, "Practicle Issues in the Implementation of Self-Tuning Control," Automatica, Vol. 20, pp. 595-605, 1984

[Yang and Asada, 1993-(a)] B.-H. Yang and H. Asada, "Adaptive Reinforcement Learning and Its Application to Robot Control", DSC-Vol. 49, Advances in Robotics, 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

[Yang and Asada, 1994] B.-H. Yang and H. Asada, "Progressive Learning for Robot Impedance Control," Proc. of ACC, Maryland, June, 1994

[Yang and Asada, 1995a] B.-H. Yang and H. Asada, "Progressive Learning for Robotic Assembly: Learning Impedance with an Excitation Scheduling Method," Proc. of IEEE Conf. on R & A, Nagoya, Japan, May, 1995

[Yang and Asada, 1995b] B.-H. Yang and H. Asada, "Progressive Learning Approach to Stable Adaptive Control," Proc. of the 34th IEEE CDC, New Orleans, Dec., 1995

[Yang, 1995] B.-H. Yang "Progressive Learning and Its Application to Robotic Assembly," Ph.D. Thesis, MIT, 1995

[Yang and Asada, 1995] B.-H. Yang and H. Asada, "Progressive Learning: An Input Design Method for Fast, Stable Learning," Proc. of American Control Conf., Seattle, June, 1995