CONCURRENT LEARNING FOR CONVERGENCE IN ADAPTIVE CONTROL WITHOUT PERSISTENCY OF EXCITATION

A Thesis
Presented to
The Academic Faculty

by

Girish V. Chowdhary

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Daniel Guggenheim School of Aerospace Engineering

Georgia Institute of Technology
December 2010

CONCURRENT LEARNING FOR CONVERGENCE IN ADAPTIVE CONTROL WITHOUT PERSISTENCY OF EXCITATION

Approved by:

Eric N. Johnson, Committee Chair
Daniel Guggenheim School of
Aerospace Engineering
Georgia Institute of Technology

Professor Wassim M. Haddad
Daniel Guggenheim School of
Aerospace Engineering
Georgia Institute of Technology

Assoc. Professor Eric N. Johnson,
Advisor
Daniel Guggenheim School of
Aerospace Engineering
Georgia Institute of Technology

Professor Magnus Egerstedt
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor Anthony Calise
Daniel Guggenheim School of
Aerospace Engineering
Georgia Institute of Technology

Asst. Professor Patricio Antonio Vela
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor Panagiotis Tsiotras
Daniel Guggenheim School of
Aerospace Engineering
Georgia Institute of Technology

Date Approved: November 2010

ACKNOWLEDGEMENTS

It is my pleasure to take this opportunity to thank some of the people who directly or indirectly supported me through this effort. I owe my deepest gratitude to my advisor and mentor Dr. Eric Johnson for his unfailing support and guidance through my time at Georgia Tech. His leadership skills and his ability to find innovative solutions above and beyond the convention will always inspire me. I would also like to thank Dr. Anthony Calise for the many interesting discussions we have had. He has inspired several insights about adaptive control and research in general. I want to thank Dr. Magnus Egerstedt for taking time out to advise me on my research in networked control. I am indebted to Dr. Wassim Haddad and Dr. Panagiotis Tsiotras for teaching me to appreciate the elegance of mathematical theory in control research. Dr. Haddad's exhaustive book on nonlinear control and my lecture notes from his class account for much of my understanding of this subject. Dr. Tsiotras' treatment of optimal, nonlinear, and robust control theory have inspired rigor and critical thought in my research. It is a pleasure having Dr. Patricio Vela on my committee, and I am thankful for the efforts he puts in his adaptive control class. I am also indebted to Dr. Eric Feron for his continued support and encouragement. He has taught me to appreciate the value of intuition and insight in controls theory research. I want to thank Dr. Ravindra Jategaonkar for teaching me to appreciate the subtleties and the art of system identification. Finally, I would like to thank all my teachers, including those at Tech, R.M.I.T., and J.P.P., I have learned a lot from them.

My time here at Tech has been made pleasurable by all my friends and colleagues.

I am specially grateful to my current and past lab-mates, peers, and friends, including Suresh Kannan, Allen Wu, Nimrod Rooz, Claus Christmann, Jeong Hur, M. Scott Kimbrell, Ep Pravitra, Chester Ong, Seung-Min Oh, Yoko Watanabe, Jincheol Ha, Hiroyuki Hashimoto, Tansel Yucelen, Rajeev Chandramohan, Kilsoo Kim, Raghavendra Cowlagi, Maxime Gariel, Ramachandra Sattegiri, Suraj Unnikrisnan, Ramachandra Rallabandi, Efstathios Bakolas, Timothy Wang, So-Young Kim, Erwan Salaün, Maximillian Mühlegg and many others. I also want to thank my colleagues and friends from Germany, who helped me prepare for this endeavor, Preeti Sankhe, Joerg and Kirsten Dittrich, Andreas and Jaga Koch, Florian, Lucas, Jzolt, Olaf, Dr. Frank Thielecke, and others.

I want to specially thank my very close friends Abhijit, Amol, Mughdha, and Mrunal, for encouraging me right from the beginning. I am grateful to my mother and father for teaching me to be strong in presence of adversities. I have probably gotten my love for the Sciences from my grandfather, Appa, who is a retired Professor of physics. I am proud to follow in his footsteps. My grandmother, Aai, has been an immense source of comfort, without which I would be lost. I am grateful to all my family, friends, and extended family for their support through my studies here. My wife, Rakshita, has stood by me through this entire time. She has helped me cope with the stress and always welcomed me with a smile on her face no matter how tough the times. For that, I am forever indebted to her; with her, I am always home.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

Model Reference Adaptive Control (MRAC) is a widely studied adaptive control methodology that aims to ensure that a nonlinear plant with significant modeling uncertainty behaves like a chosen reference model. MRAC methods attempt to achieve this by representing the modeling uncertainty as a weighted combination of known nonlinear functions, and using a weight update law that ensures weights take on values such that the effect of the uncertainty is mitigated. If the adaptive weights do arrive at an ideal value that best represent the uncertainty, significant performance and robustness gains can be realized. However, most MRAC adaptive laws use only instantaneous data for adaptation and can only guarantee that the weights arrive at these ideal values if and only if the plant states are Persistently Exciting (PE). The condition on PE reference input is restrictive and often infeasible to implement or monitor online. Consequently, parameter convergence cannot be guaranteed in practice for many adaptive control applications. Hence it is often observed that traditional adaptive controllers do not exhibit long-term-learning and global uncertainty parametrization. That is, they exhibit little performance gain even when the system tracks a repeated command.

This thesis presents a novel approach to adaptive control that relies on using current and recorded data concurrently for adaptation. The thesis shows that for a concurrent learning adaptive controller, a verifiable condition on the linear independence of the recorded data is sufficient to guarantee that weights arrive at their ideal values even when the system states are not PE. The thesis also shows that the same condition can guarantee exponential tracking error and weight error convergence to zero, thereby allowing the adaptive controller to recover the desired transient response

and robustness properties of the chosen reference models and to exhibit long-term-learning. This condition is found to be less restrictive and easier to verify online than the condition on persistently exciting exogenous input required by traditional adaptive laws that use only instantaneous data for adaptation. The concept is explored for several adaptive control architectures, including neuro-adaptive flight control, where a neural network is used as the adaptive element. The performance gains are justified theoretically using Lyapunov based arguments, and demonstrated experimentally through flight-testing on Unmanned Aerial Systems.

CHAPTER I

INTRODUCTION

Control technologies are enabling a multitude of capabilities in modern systems. In fact, for modern systems such as unmanned aircraft and space vehicles, control systems are often critical to the system's safety and functionality. Hence, the design of efficient and robust control systems has been heavily researched. Most well-understood methods of control design rely on developing a mathematical models of systems and their physical interconnections. However, it is not realistic to expect that a perfect mathematical model of a physical system will always be available. Therefore, "real-world" controllers must account for modeling uncertainties to ensure safe operation in uncertain environments. Adaptive control is framework that allow the design of control systems for plants with significant modeling uncertainties without having to first obtain a detailed dynamical model. Most adaptive controllers achieve this by adjusting online a set of controller parameters using available information.

In flight control applications, having an accurate model for aircraft for example, means significant effort must be spent on modeling from first principles, system identification using flight test data and wind tunnel data, and model verification. Furthermore, a single model that describes aircraft dynamics accurately over its entire operating envelop often ends up being nonlinear and coupled. Hence, a single linear controller often cannot be used over the entire flight envelop. Robust control is one approach that has been extensively studied for systems with uncertainties. In these methods, an estimate of the structure and the magnitude of the uncertainty is used to design static linear controllers that function effectively in presence of the uncertainties (see for example [100], [24], [6] and the references therein). One benefit of robust

control methods is that the linear models used for design need not be extremely accurate. By their nature however, robust controllers are conservative, and can result in poor performance. Nonlinear model based methods have also been studied for aircraft control. These include backstepping, sliding mode control, state dependent Riccati equations, and Lyapunov design. These methods rely on a nonlinear model of the aircraft, and their performance can be affected the model's fidelity. Furthermore, well understood linear stability metrics such as gain margin and phase margin do not translate easily to nonlinear designs, thus providing the control designer with no metrics to characterize stability and performance. Hence, there are not many industrial implementations of these methods.

One prevailing trend in aerospace applications has been to identify several linear models around different trim points, design linear controllers for each of these linear models, and devise a switching or scheduling scheme to switch between the different controllers. Some authors consider such switching controllers as some of the first adaptive controllers devised [3]. Subsequent adaptive control designs followed heuristic rules that varied controller parameters to achieve desired performance. These designs suffered from lack of rigorous stability proofs, and important lessons about the effects of deviating from the rigor of control theory were learned at great expense. The most well known example is that of the NASA X-15 flight tests, in which it is believed that a heuristic adaptive controller resulted in loss of aircraft when operating in off-nominal condition [8]. More modern methods of adaptive control however, use Lyapunov based techniques to form a framework for adaptive control theory in which the stability of different adaptive laws can be ascertained rigorously. In fact, Dydek, Annaswamy, and Lavretsky have argued that modern Lyapunov based methods could have prevented the X-15 crash [26]. The two main differences between the modern methods of adaptive control and the older scheduling methods are that the modern methods employ a single control law that varies the controller parameters to

accommodate modeling uncertainty over the plant operating domain, and that modern adaptive controllers are motivated through nonlinear stability analysis, and have associated stability proofs.

Modern adaptive controllers can be roughly classified as "direct adaptive controllers" and "indirect adaptive controllers". Direct adaptive controllers traditionally use the instantaneous tracking error to directly modify the parameters of the controller. Direct adaptive controllers are characterized by fast control response and efficient tracking error mitigation. However, direct adaptive controllers are not focused on estimating the uncertainty itself, and hence often suffer from "Short Term Learning", that is, their tracking performance does not necessarily improve over time, even when the same command is repeatedly tracked. On the other hand, indirect adaptive controllers use the available information to form an estimate of the plan dynamics and use this information to control the plant. Therefore, as the estimate of the plant dynamics becomes increasingly accurate, the tracking performance of indirect adaptive controllers improves. However, the reliance on estimating plant dynamics can often lead to poor transient performance in indirect adaptive control if the initial estimates are not accurate. This fact makes it hard to provide guarantees of performance and stability for indirect adaptive control methods.

The most widely studied class of direct adaptive control methods is known as Model Reference Adaptive Control (MRAC) (see for example [70, 3, 43, 93, 40] and the references therein). In MRAC the aim is to design a control law that ensures that the states of the plant track the states of an appropriately chosen reference model which characterizes the desired transient response and stability properties of the plant. Other notable recent adaptive control methods include adaptive backstepping and tuning function based methods (see for example [56]). Adaptive backstepping is a powerful approach with many emerging applications. However, it relies

on the knowledge of higher order plant state derivatives which are not easy to estimate. Furthermore, complex instructions must be implemented in software for this approach, which makes it susceptible to numerical issues. Perhaps due to these reasons, limited success has been obtained with this method in real-world applications, for example the results of Ishihara et al. suggest that adaptive autopilots developed with adaptive backstepping could be highly sensitive to time-delays [44]. In this thesis we will not pursue adaptive backstepping further, rather we will be concerned with extending MRAC with a novel Concurrent Learning adaptive control framework that combines the benefits of direct and indirect adaptive control.

1.1 Model Reference Adaptive Control

MRAC has been widely studied for a class of nonlinear systems with modeling uncertainties and full state feedback (see [70],[3],[43],[93] and the references therein). Many physical systems can be controlled using MRAC approaches, and wide ranging applications can be found, including control of robotics arms (see for example [55], [77]), flight vehicle control, (see for example [48], [50], [90]), and control of medical processes (see for example [33], [95], [96]). MRAC architectures are designed to guarantee that the controlled plant states x track the states x_{rm} of an appropriately chosen reference model which characterizes the desired transient response and robustness properties. Most MRAC methods achieve this by using a parameterized model of the uncertainty, often referred to as the adaptive element and its parameters referred to as adaptive weights. Adaptive elements in MRAC can be classified as those that are designed to cancel structured uncertainties, and those designed to cancel unstructured uncertainties. In problems where the structure of the modeling uncertainty is known, that is, where it is known that the uncertainty can be linearly parameterized using a set of known nonlinear basis functions, the adaptive element is formed by using a weighted combination of the known basis (see for example [69, 70, 40]). In

this thesis we refer to this case as the case of structured uncertainty. For this case it is known that if the adaptive weights arrive at the ideal (true) weights then the uncertainty can be uniform canceled. In problems where the structure of the uncertainty is not known but it is known that the uncertainty is continuous and defined over a compact domain, Neural Networks have been used by many authors as adaptive elements [61, 48, 59, 54, 53, 47]. In this case the universal approximation property of neural networks guarantees that a set of ideal weights exists that guarantees optimal approximation of the uncertainty with a bounded error that is valid over a compact domain. In this thesis we refer to this case as the case of unstructured uncertainty.

The key point to note about the MRAC architecture is that it is designed to augment a baseline linear control architecture with an adaptive element, whose parameters are updated to cancel the uncertainties in the plant. Even when these uncertainties are linear, the adaptive law itself becomes nonlinear. This is a result of multiplications between the real system states and the adaptive weights, which can be thought of as augmented system states. However, the tracking error dynamics in MRAC are formed through a combination of an exponentially stable linear term in the error e with a nonlinear disturbance term equal to the difference between the adaptive element's estimate of the uncertainty and the true uncertainty. Hence, if the adaptive weights arrive at the ideal weights, the linear tracking error dynamics of MRAC can be made to dominate.

Traditionally in MRAC, the adaptive law is designed to update the parameters in the direction of maximum reduction of the instantaneous tracking error cost (e.g. $V(t) = e^T(t)e(t)$). Such minimization results in a weight update law that is at most rank-1 [20, 22]. This approach aids in ensuring that the parameters take on values such that the tracking error is instantaneously suppressed, it does not however guarantee the convergence of the parameters to their ideal values unless the system states are Persistently Exciting (PE) [70, 43, 93, 3] (one exception that is not pursued

further here is the special case of uncertainties with periodic regressor functions [4]). A mathematical definition of what constitutes a persistently exciting signal is given in definition 3.2. In essence, the PE condition requires that over all predefined time intervals, the plant states span the complete spectrum of the state space. Boyd and Sastry have shown that the condition on PE system states can be related to a PE exogenous reference input by noting the following: If the exogenous reference input $r(t)$ contains as many spectral lines as the number of unknown parameters, then the plant states are PE, and the parameter error converges exponentially to zero [9]. However, the condition on PE reference input is restrictive and often infeasible to implement or monitor online. For example, in adaptive flight control applications, PE reference inputs may be operationally unacceptable, waste fuel, and may cause undue stress on the aircraft. Furthermore, since the exogenous reference inputs for many online applications are event-based and not known *a-priori*, it is often impossible to verify online whether a signal is PE. Consequently, parameter convergence cannot be guaranteed in practice for many adaptive control applications.

Various methods have been developed to guarantee robustness and efficient uncertainty suppression without PE reference inputs. These include the classic σ modification of Ionnou [43] and the e modification of Narendra [69] which guarantee that the adaptive weights do not diverge even when the system states are not PE. Further modifications include projection based modifications in which the weights are constrained to a compact set through the use of a weight projection operator [93, 40]. These modifications however, are aimed at ensuring boundedness of weight rather than uncertainty cancelation. The motivation being that if the weights stay bounded then an application of the Barbalat's lemma results in asymptotic tracking error convergence. However, this approach suffers from the issue that transient response of the tracking error cannot be guaranteed. Furthermore, most implementations of σ and e modification as well as projection operator based modifications bound the weights

around a neighborhood of a preselected value, usually set to zero. This can slowdown or even prevent the adaptive element from estimating constants that are far away from zero, such as trims or input biases.

Recently Volyanksyy et al. have introduced the Q modification approach [94, 96, 95]. In Q modification, an integral of the tracking error is used to drive the weights to a hypersurface that contains the ideal weights. The rationale in Q modification is that weight convergence is not necessary as long as the uncertainty is instantaneously canceled. Weight convergence does occur however, if states are PE. In the recent L_1 control approaches Cao, Hovakimyan, and others have used a low pass filter on the output of the adaptive element to ensures that high adaptive gains can be used to instantaneously dominate the uncertainty [15, 13]. Nguyen has developed an "optimal control modification" to adaptive control which also allows high adaptation gains to be used to efficiently suppress the uncertainty [71]. The main focus in many such methods however has been on instantaneously dominating the uncertainty rather than guaranteeing weight convergence. In fact, many authors have argued that guaranteed weight convergence is not required in MRAC schemes if the only concern is to guarantee $e(t) \to 0$ as $t \to \infty$. However, asymptotic convergence of tracking error does not guarantee transient performance, and further modifications, such as those introduced in L_1 adaptive control must be used. On the other hand, if the adaptive weights do converge to their ideal values, then the uncertainty is uniformly canceled over an operating domain of the plant. This allows the linear (in e), exponentially stable, tracking error dynamics of MRAC to dominate, guaranteeing that the tracking error vanishes exponentially, thus recovering the desired transient performance and robustness properties of the chosen reference model.

Furthermore, we also agree with the authors in [9], and [1] that exponential weight convergence is needed to meaningfully discuss robustness of adaptive controllers using linear metrics, with the authors in [86] that exponential weight convergence leads to

exponential tracking error convergence, and with the authors in [14] that weight convergence is needed to handle a class of adaptive control problems where the reference input is dependent on the unknown parameters.

In summary, weight convergence results in the following benifits:

- Exponential error convergence
- Guaranteed exponentially bounded transient performance
- Uniform approximation of plant uncertainty, effectively making the tracking error dynamics linear
- If plant uncertainty is uniformly canceled, the plant tracks the reference model exponentially. For an appropriately chosen reference model the plant states will become exponentially indistinguishable from the reference model states. This allows us to meaningfully speak about recovering the phase and gain margin and the transient performance characteristics of the reference model, and thus meaningfully evaluate the performance of the controller through well understood linear stability metrics.

We note that the requirement on PE system states is common for guaranteeing parameter convergence in adaptive control methods other than MRAC, including adaptive backstepping [56]. Therefore the methods presented in this thesis should be of interest beyond MRAC.

To realize the benefits of weight convergence, other authors have sought to combine direct and indirect adaptive control to guarantee efficient tracking error reduction and uniform uncertainty cancelation through weight convergence. Duarte and Narendra introduced the concept of combined direct-indirect adaptive control [25]. Among others, Yu et al. explored combined direct and indirect adaptive control for control of constrained robots [98], Dimtris et al. combined direct and indirect adaptive control for control using artificial neural networks [79]. Slotine and Li introduced the

Composite MRAC method for combining direct and indirect adaptive control [88], which has been further studied by Lavretsky [58]. Nguyen studied the use of recursive least squares to augment a direct adaptive law [72]. In these efforts, the aim is to develop an adaptive law that trains on a signal other than the instantaneous tracking error to arrive at an accurate estimate of the plant uncertainty. That is, to ensure that the parameter error converges to zero, thereby ensuring that the weights converge to their ideal values. However, these methods require that the plant states be persistently exciting for the weights to converge.

1.2 Contributions of This Work

The main contribution of this thesis is to show that if recorded data are used concurrently with current data for adaptation, a simple condition on the richness of the recorded data is sufficient to guarantee exponential tracking error and parameter error convergence in MRAC; without requiring PE exogenous reference input. Adaptive control laws making such concurrent use of recorded and current data are defined here as "Concurrent Learning" adaptive laws. The concurrent use of recorded and current data is motivated by the intuitive argument that if the recorded data is made sufficiently rich and used concurrently for adaptation, then weight convergence can occur without the system states being persistently exciting. In this thesis, this intuitive argument is formalized and it is shown that if the following condition on the recorded data is met, then exponential tracking error and parameter convergence can be achieved:

> The recorded data have as many linearly independent elements as the dimension of the basis of the uncertainty.

This condition relates weight convergence to the spectral properties of the recorded data, and in this way differs from the classical PE condition, which relates the convergence of weights to the spectral properties of future system signals (see for example

Boyd and Sastry 1986 [9]). Furthermore, the condition stated in this thesis is less restrictive than a condition on PE reference input, and is conducive to online verification. The following is a summary of the main contributions of this work.

A method that guarantees exponential convergence in adaptive control:

Currently in order to guarantee exponential tracking error convergence in adaptive control, the states need to be PE. This thesis presents a method that concurrently uses current and recorded data to guarantee exponential tracking error convergence in adaptive control subject to an easily verifiable condition on linear independence of the recorded data.

Guaranteed transient performance:

The concurrent learning adaptive laws presented in this thesis guarantee that the tracking performance of the adaptive controller is exponentially bounded once the stated condition on the recorded data is met. Furthermore, since a-priori recorded data can be used, the method provides a way to guarantee exponential transient performance bounds even before it has been turned on.

Guaranteed uncertainty Characterization:

The concurrent learning adaptive laws presented in this thesis guarantee that the adaptive weights will converge to their ideal values if the stated verifiable condition on the recorded data is met. This allows for a mechanism that can be used to monitor whether the uncertainty has been approximated. Furthermore, the approximated uncertainty can be used to improve control and guidance performance.

Pathway to Stability Metrics for Adaptive Controllers:

If plant uncertainty is uniformly canceled, the plant tracks the reference model exponentially. Hence, for an appropriately chosen reference model the plant states will become exponentially indistinguishable from the reference model

states. For aerospace applications particularly, guaranteed weight convergence is of utmost importance. Because if the weights converge, the performance and robustness measures associated with the baseline linear control design will be recovered, and hence handling specifications such as those in reference [89] can be used [82],[50], enabling a pathway to flight certification of adaptive controllers.

A concurrent gradient descent law that converges without PE signals:

Gradient descent bases methods have been widely used to solve parameter identification problems which are linearly parameterized. In these methods, the parameters are updated in the direction of maximum reduction of a quadratic cost on the estimation error. Such gradient based parameter update laws have been used for NN training [36], in system identification, and in decentralized control of networked robots [27]. It is well known that gradient based adaptive laws are subject to being stuck at local minima and do not have guaranteed rate of convergence. Many different methods have been tried to remedy this situation. Among others, Jankt has tried adaptive learning rate schemes to improve performance of gradient based NN training algorithms [45], Ochai has tried to use kickout algorithms for reducing the possibility of weights being stuck at local minima [73]. However, the fact remains that the only way to guarantee the convergence of gradient based adaptive laws that only use instantaneous data is to require that the system signals are PE [3, 93]. In this thesis we show that if recorded data is used concurrently with current data for gradient based training, then a verifiable condition on linear independence of the recorded data is sufficient to guarantee global exponential weight convergence for these problems. This result has wide ranging implications beyond adaptive control.

In a broader sense, this thesis represents one of the first rigorous attempts to evaluate the impact of memory on adaptive control and parameter identification algorithms. Many previously studied methods that use memory in order to improve performance of adaptive algorithms have been heuristic. For example, one commonly used approach is to add a "momentum term" to standard gradient based weight update laws [92, 36, 78]. The momentum term scales the most recent weight update in the direction of the last weight update. This speeds up the convergence of weights when in the vicinity of local minima, and slows the divergence. This heuristic modification cannot guarantee the convergence of weights, and results only in a modest improvement. Another common approach is the use of a forgetting factor which can be tuned to indicate the degree of reliance on past data [42]. This approach is also heuristic, and suffers from the drawbacks that the forgetting factor is difficult to tune, and an improper value can adversely affect the performance of the adaptive controller. Particularly, a smaller value of the forgetting factor indicates higher reliance on recent data, which could lead to local parameterizations, while a larger value of the forgetting factor indicates higher reliance on past data, which could lead to sluggish adaptation performance. Patiño et al. suggested the use of a bank of NNs trained around different operating conditions as a basis for the space of all operating conditions [77]. The required model error was then calculated by using a linear combination of the outputs of these different NNs. In order to overcome the shortcomings of online training algorithms, Patiño et al. also suggested that the bank of NNs be adapted off-line using recorded data. The reliance on off-line training makes this approach inappropriate for adaptive flight applications. All of these methods represent important heuristic "tweaks" that can improve controller performance, however, they lack rigorous justification and are not guaranteed to work on all problems. In this thesis however, we introduce a method that uses memory along with the associated theory that characterizes the impact and benefit of including memory. In that sense, another

contribution of this thesis is to rigorously show that recorded data can indeed be used to significantly improve the performance of control algorithms. These findings are in excellent agreement with those of Bernstein et al., who have used recorded data to design retrospective cost optimizing adaptive controllers (see for example [84], [85], [37]). The fact that memory can be used to improve adaptive control performance has interesting implications, especially when one considers that modern embedded computers can easily handle control algorithms that go beyond simple instantaneous calculations.

1.3 Outline of the Thesis

We begin by discussing MRAC in Chapter 2. In that chapter, the classical parameter adaptation laws, and MRAC adaptive laws for both cases of structured and unstructured uncertainties are presented. In Chapter 3 concurrent learning adaptive laws that use instantaneous and recorded data concurrently for adaptation are presented. Theorem 3.1 shows that a concurrent learning gradient based parameter update law guarantees exponential parameter convergence in parameter identification problems without PE states subjects to a verifiable condition on linear independence of the recorded data (Condition 3.1), referred to here as the rank-condition. In Theorem 3.2 it is shown that a concurrent learning adaptive law guarantees exponential parameter error and tracking error convergence in adaptive control problems with structured uncertainty subject to the rank-condition, without requiring PE exogenous inputs. In Theorem 3.3 it is shown that a concurrent learning adaptive law that prioritizes learning on current data over that of learning on recorded data guarantees asymptotic tracking error and parameter error convergence subject to the rank-condition.

Concurrent learning adaptive control is extended to neuro-adaptive control in Chapter 4 for a class of nonlinear systems with unstructured uncertainties. For this class of systems Theorem 4.1 shows that the rank-condition is sufficient to guarantee

that the adaptive weights of a radial basis function NN stay bounded within a compact neighborhood of the ideal weights when using concurrent learning adaptive laws. In Chapter 5 the results are extend to approximate model inversion based MRAC for adaptive control of a class of multi- input-multi-state nonlinear systems. and show that the rank-condition is once again sufficient to guarantee exponential parameter and tracking error convergence.

In Section 6 we discuss methods for selecting data points in order to maximize convergence rate. In Chapter 7 we show that least squares based methods can also be used for concurrent learning adaptive control. We show that a modified adaptive law that drives the weights to an online least squares estimate of the ideal weights can guarantee exponential convergence subject again to the rank-condition.

In Chapters 8 and 9 the developed methods are implemented on real flight hardware, and flight test results that characterize the improvement in performance are presented. In Chapter 10 the problem of network discovery for a decentralized network of mobile robots is discussed, and it is shown that under two key assumptions the problem can be posed as that of parameter estimation. Simulation results using the concurrent gradient descent law for solving the network discovery problem are presented.

The thesis is concluded in Chapter 11 and future research directions are suggested in Section 11.1.

1.3.1 Some Comments on Notation

In this thesis, $f(t)$ represents a function of time t. Often we will drop the argument t consistently over an entire equation for ease of exposition. Indices are denoted only by subscripts. The operator $\|.\|$ denotes the Euclidian norm unless otherwise stated. For a vector ξ and a positive real constant a we define the compact ball B_a as $B_a = \{\xi : \|\xi\| \leq a\}$. We let ∂D denote the boundary of the set D. If a vector

function $\xi(t)$ is equivalently equal to zero for all time $t \geq T,\ T \in \Re^{+}$ then we say that $\xi \equiv 0$.

CHAPTER II

MODEL REFERENCE ADAPTIVE CONTROL

2.1 Adaptive Laws for Online Parameter Estimation

Parameter estimation is concerned with using available information to form an online estimate of unknown system parameters and has been widely studied (see for example [3], [69], [86], [93], [46] and the references therein). In parameter estimation for flight system identification for example, the parameters to be estimated are directly related to meaningful physical quantities such as aerodynamic derivatives. Hence, the convergence of the unknown parameters to their true values is highly desirable. We shall assume that the problem is posed such that the unknown system dynamics are linearly parameterized. Hence letting $y(t) : \Re^m \to \Re$ denote the measured output of an unknown linearly parameterized model whose unknown parameters are contained in the constant ideal weight vector $W^* \in \Re^m$, whose basis function $\Phi(x)$ is continuously differentiable, and the measurements $\Phi(x(t)) \in D$ where $D \subset \Re^m$ is a compact set, we have

$$y(t) = W^{*T}\Phi(x(t)). \tag{2.1}$$

Note that the regressor vector $\Phi(x)$ can be a nonlinear function that represents a meaningful system signal, however the model 2.1 itself is linearly parameterized as it represents an unknown linear combination of a known basis.

Let $W(t) \in \Re^m$ denote an online estimate of the ideal weights W^*; then an online estimate of y can be given by the mapping $\nu : \Re^m \to \Re$ in the following form:

$$\nu(t) = W^T(t)\Phi(x(t)). \tag{2.2}$$

This results in an approximation error $\epsilon(t) = \nu(t) - y(t)$:

$$\epsilon(t) = (W(t) - W^*)^T \Phi(x(t)). \tag{2.3}$$

Letting $\tilde{W}(t) = W(t) - W^*$ we have,

$$\epsilon(t) = \tilde{W}^T(t)\Phi(x(t)). \tag{2.4}$$

In the above form it is clear that $\epsilon(t) \to 0$ uniformly as $t \to \infty$ if the parameter error $\tilde{W}(t) \to 0$ as $t \to \infty$. Therefore, we wish to design a parameter adaptation law $\dot{W}(t)$, which uses the measurements of $x(t)$, $y(t)$, and the knowledge of the mapping $\Phi(.)$, to ensure $W(t) \to W^*$ as $t \to \infty$. A well known choice for $\dot{W}(t)$ is the following gradient based adaptive law which updates the adaptive weight in the direction of maximum reduction of the instantaneous quadratic cost $V(t) = \epsilon^T(t)\epsilon(t)$ [3], [69], [43], [93],

$$\dot{W}(t) = -\Gamma\Phi(x(t))\epsilon(t), \tag{2.5}$$

where $\Gamma > 0$ contains the learning rate.

It is well known that when using the gradient descent based parameter adaptation law of equation 2.5, $W(t) \to W^*$ as $t \to \infty$ if and only if the vector signal $\Phi(x(t)) \in \Re^m$ is Persistently Exciting (PE) [93], [3], [43], [1], [70].

2.2 Model Reference Adaptive Control

In this section, an introduction to Model Reference Adaptive Control (MRAC) is presented. Let $x(t) \in \Re^n$ be the known state vector, let $u(t) \in \Re$ denote the control input, and consider the following system:

$$\dot{x}(t) = Ax(t) + B(u(t) + \Delta(x(t))), \tag{2.6}$$

where $A \in \Re^{n\times n}$, $B = [0, 0, ..., 1]^T \in \Re^n$, and $\Delta(x)$ is a continuous function representing the scalar uncertainty. The assumption on scalar input and the form of B matrix

is made for ease of exposition in this section, these assumptions are lifted in chapte 5. We assume that the pair (A, B) in equation 2.6 is controllable.

A reference model can be designed that characterizes the desired response of the system

$$\dot{x}_{rm}(t) = A_{rm}x_{rm}(t) + B_{rm}r(t), \tag{2.7}$$

where $A_{rm} \in \Re^{n\times n}$ is a Hurwitz matrix and $r(t)$ denotes a bounded reference signal. A tracking control law consisting of a linear feedback part $u_{pd}(t) = K(x_{rm}(t) - x(t))$, a linear feedforward part $u_{crm}(t) = K_r[x_{rm}^T(t), r(t)]^T$, and an adaptive part $u_{ad}(x)(t)$ is chosen to have the following form

$$u = u_{crm} + u_{pd} - u_{ad}. \tag{2.8}$$

Note that in the above equation, we assumed that the baseline linear design is attempting to make the plant behave like the reference model, hence the linear feedback controller operates on the tracking error e.

2.2.1 Tracking Error Dynamics

The tracking error e is the difference between the plant state and the state of the reference model and is defined as:

$$e(t) = x_{rm}(t) - x(t). \tag{2.9}$$

Differentiating equation 2.9 we have

$$\dot{e}(t) = A_{rm}x_{rm}(t) + B_{rm}r(t) - (Ax(t) + B(u(t) + \Delta(x(t)))), \tag{2.10}$$

letting $\Delta A = A_{rm} - A$ and using the control law in 2.8 the above equation can be further simplified to

$$\dot{e}(t) = A_m e(t) + \Delta A x_{rm} + B_{rm}r(t) - Bu_{crm}(t) + B(u_{ad}(t) - \Delta(t)), \tag{2.11}$$

Assuming that an appropriate choice of u_{crm} exists such that the matching condition $Bu_{crm} = (A_{rm} - A)x_{rm} + B_{rm}r$ is satisfied, the tracking error dynamics can be written as

$$\dot{e} = A_m e + B(u_{ad}(x) - \Delta(x)), \tag{2.12}$$

where the baseline full state feedback controller $u_{pd} = Ke$ is chosen such that $A_m = A - BK$ is a Hurwitz matrix. Hence for any positive definite matrix $Q \in \Re^{n\times n}$, a positive definite solution $P \in \Re^{n\times n}$ exists to the Lyapunov equation

$$A_m^T P + P A_m + Q = 0. \tag{2.13}$$

2.2.2 Case I: Structured Uncertainty

Consider the case where the structure of the uncertainty $\Delta(x)$ is known, that is, it is known that the uncertainty can be represented as a linear combination of a known continuously differentiable basis function. This case is captured by the following assumption.

Assumption 2.1 The uncertainty $\Delta(x)$ can be linearly parameterized, that is, there exist a unique constant vector $W^* \in \Re^m$ and a vector of known continuously differentiable regressor functions $\Phi(x(t)) = [\phi_1(x(t)), \phi_2(x(t)), ..., \phi_m(x(t))]$, such that there exists an interval $[t, t + \Delta t]$, $\Delta t \in \Re^+$ over which the integral $\int_t^{t+\Delta t} \Phi(x(t))\Phi^T(x(t))dt$ can be made positive definite for bounded $\Phi(x(t))$, and $\Delta(x)$ can be uniquely represented as

$$\Delta(x(t)) = W^{*T}\Phi(x(t)). \tag{2.14}$$

A large class of nonlinear uncertainties can be written in the above form (see for example the nonlinear wing-rock dynamics model [87], [66]). Note that the requirement on unique W^* for a given basis of the uncertainty $\Phi(x(t))$ ensures that the representation of equation 2.14 is minimal, that is functions such as $\Delta(x) = w_1^* \sin(x(t)) + w_2^* \cos(x) + w_3^* \sin(x)$ are represented as $\Delta(x) = [w_1^* + w_3^*, w_2^*]^T[\sin(x), \cos(x)]$. Since

the mapping $\Phi(x)$ is known, letting $W(t) \in \Re^{m \times n}$ denote the estimate W^* the adaptive law can be written as

$$u_{ad}(x(t)) = W^T(t)\Phi(x(t)). \tag{2.15}$$

For this case it is well known that the adaptive law

$$\dot{W} = -\Gamma_W \Phi(x) e^T P B \tag{2.16}$$

where Γ_W is a positive definite learning rate matrix results in $e(t) \to 0$ as $t \to \infty$; however 2.16 does not guarantee the convergence (or even the boundedness) of W. [93]. Equation 2.16 will be referred to as the baseline adaptive law. For the baseline adaptive law, it is also well known that a necessary and sufficient condition for guaranteeing $\lim_{t\to\infty} W(t) = W^*$ is that $\Phi(t)$ be PE [70], [43], [93]. Furthermore, Boyd and Sastry have shown that $\Phi(t)$ can be made PE if the exogenous reference input has as many spectral lines as the unknown parameters [9].

2.2.3 Case II: Unstructured Uncertainty

In the more general case where it is only known that the uncertainty $\Delta(x)$ is continuously differentiable and defined over a compact domain $D \subset \Re^n$, the adaptive part of the control law can be formed using Neural Networks (NNs). In the following we will present two different types of NN for capturing unstructured uncertainty.

2.2.3.1 Radial Basis Function Neural Network

The output of a Radial Basis Function (RBF) NN [36] can be given as

$$u_{ad}(x) = W^T \sigma(x). \tag{2.17}$$

where $W \in \Re^{l \times n}$ and $\sigma(x) = [1, \sigma_2(x), \sigma_3(x), ..., \sigma_l(x)] \in \Re^l$ is a vector of known radial basis functions. In this case, l denotes the number of radial basis function nodes in the NN. For $i = 2, 3..., l$ let c_i denote the RBF centroid and μ_i denote the

RBF width then for each i The radial basis functions are given as

$$\sigma_i(x) = e^{-\|x-c_i\|^2/\mu_i} \tag{2.18}$$

Appealing to the universal approximation property of Radial Basis Function Neural Networks (see [76], [36], or [92]) we have that given a fixed number of radial basis functions l there exists ideal weights $W^* \in \Re^l$ and $\tilde{\epsilon}(x) \in \Re$ such that the following approximation holds for all $x \in D \subset \Re^m$ where D is compact

$$\Delta(x) = W^{*T}\sigma(x) + \tilde{\epsilon}(x), \tag{2.19}$$

and $\bar{\epsilon} = \sup_{x\in D} \|\tilde{\epsilon}(x)\|$ can be made arbitrarily small given sufficient number of radial basis functions. For this case it is well known that the baseline adaptive law of equation 2.16 (with $\Phi(x(t))$ replaced by $\sigma(x(t))$) guarantees uniform ultimate boundedness of the tracking error, and guarantees that the adaptive weights stay bounded within a neighborhood of the ideal weights if the system states are PE (see for example [61], [55] and the references therein).

2.2.3.2 Single Hidden Layer Neural Network

A Single Hidden Layer (SHL) NN is a nonlinearly parameterized map that has also been often used for capturing unstructured uncertainties that are known to be continuous. The output of a SHL NN can be given as

$$u_{ad}(x) = W^T\sigma(V^T\bar{x}). \tag{2.20}$$

The terms $W, V, \bar{x}$ are defined in the following. Let n_3 denote the number of output layer neurons, n_2 denote the number of hidden layer neurons, and n_1 denote the number of input layer neurons. Note that for the uncertainty in equation 2.6, $n_3 = 1$. For SHL NN representation in equation 2.20 $W \in \Re^{(n_2+1)\times n_3}$ is the NN synaptic weight matrix connecting the hidden layer with the output layer. Letting

Θ_{w_i} denote the hidden layer bias for the i^{th} hidden layer neuron, we have the following form for W

$$W = \begin{pmatrix} \Theta_{w,1} & \cdots & \Theta_{w,n_3} \\ w_{1,1} & \cdots & w_{1,n_3} \\ \vdots & \ddots & \vdots \\ w_{n_2,1} & \cdots & w_{n_2,n_3} \end{pmatrix} \in \Re^{(n_2+1)\times n_3}, \tag{2.21}$$

The NN synaptic weight matrix connecting the input layer with the hidden layer is given by $V \in \Re^{(n_1+1)\times n_2}$. Letting Θ_{v_i} denote the hidden layer bias for the i^{th} input layer neuron, we have the following form for V

$$V = \begin{pmatrix} \Theta_{v,1} & \cdots & \Theta_{v,n_2} \\ v_{1,1} & \cdots & v_{1,n_2} \\ \vdots & \ddots & \vdots \\ v_{n_1,1} & \cdots & w_{n_1,n_2} \end{pmatrix} \in \Re^{(n_1+1)\times n_2}, \tag{2.22}$$

The input to the NN is given by $\bar{x} \in D \subset \Re^{n_1+1}$, where D is a compact set, and $\bar{x}$ contains the states over which the uncertainty is to be parameterized x_{in} and the constant bias term b_v usually set to 1

$$\bar{x} = \begin{pmatrix} b_v \\ x_{in} \end{pmatrix} = \begin{pmatrix} b_v \\ x_{in_1} \\ x_{in_2} \\ \vdots \\ x_{in_{n_1}} \end{pmatrix} \in \Re^{n_1+1}. \tag{2.23}$$

For ease in notation, let $z = V^T\bar{x} \in \Re_2^n$, and b_w denote the constant bias term usually set to 1 for the hidden layer neuron. Then the vector function $\sigma(z) \in \Re^{n_2+1}$

is given by

$$\sigma(z) = \begin{pmatrix} b_w \\ \sigma_1(z_1) \\ \vdots \\ \sigma_{n_2}(z_{n_2}) \end{pmatrix} \in \Re^{n_2+1}. \tag{2.24}$$

The elements of σ consist of sigmoidal activation functions, which are given by

$$\sigma_j(z_j) = \frac{1}{1+e^{-a_j z_j}}. \tag{2.25}$$

Single Hidden Layer (SHL) perceptron NN are known to be universal approximators (see [38] or [92]). That is, given an $\bar{\epsilon} > 0$, for all $\bar{x} \in D$, where D is a compact set, there exists a number of hidden layer neurons n_2, and an ideal set of weights (W^*, V^*) that brings the NN output to within an ϵ neighborhood of the function approximation error. The largest such ϵ is given by

$$\bar{\epsilon} = \sup_{\bar{x} \in D} \left\| W^{*^T} \sigma(V^{*^T} \bar{x}) - \Delta(\bar{x}) \right\|. \tag{2.26}$$

Hence in a similar fashion to RBF NN we have that the following approximation holds for all $x \in D \subset \Re^n$ where D is compact

$$\Delta(x) = W^{*^T} \sigma(V^{*^T} \bar{x}) + \tilde{\epsilon}(x), \tag{2.27}$$

and $\bar{\epsilon} = \sup_{\bar{x} \in D} \|\tilde{\epsilon}(x)\|$ can be made arbitrarily small given sufficient number of hidden layer neurons.

For this case it has been shown that the following adaptive laws which contain an e-modification term with $\kappa > 0$ (see [69]) guarantee uniform ultimate boundedness of the tracking error, and guarantees that the adaptive weights stay bounded (see for example [61], [55] and the references therein)

$$\dot{W} = -(\sigma(V^T \bar{x}) - \sigma'(V^T \bar{x}) V^T \bar{x}) r^T \Gamma_w - k\|e\| W \tag{2.28}$$

$$\dot{V} = -\Gamma_V \bar{x} r^T W^T \sigma'(V^T \bar{x}) - k\|e\| V. \tag{2.29}$$

CHAPTER III

CONCURRENT LEARNING ADAPTIVE CONTROL

3.1 Persistency of Excitation

It is well known that when using instantaneous gradient descent (see equation 2.5) to solve the online parameter estimation problem described in Section 2.1, the online weight estimates will arrive at their ideal values if and only if the vector signal $\Phi(x(t)) \in \Re^m$ is Persistently Exciting (PE) [93], [3], [43], [1], [70]. For the case of adaptive control, Boyd and Sastry have shown that the condition on persistency of excitation in the system states ($\Phi(x)$) can be related to persistency of excitation in the exogenous reference input $r(t)$ by noting the following: If the exogenous reference input $r(t)$ contains as many spectral lines as the number of unknown parameters, then the plant states are PE, and the parameter error converges exponentially to zero [9]. Hence exponential parameter and tracking error convergence in Model Reference Adaptive Control (MRAC) that uses only instantaneous data for adaptation (equation 2.16) is dependent on persistency of excitation in system states.

Various equivalent definitions of excitation and the persistence of excitation of a bounded vector signal exist in the literature (see for example [3], [70]), we will use the definitions proposed by Tao in [93]:

Definition 3.1 A bounded vector signal $\Phi(t)$ is *exciting* over an interval $[t, t+T]$, $T > 0$ and $t \geq t_0$ if there exists $\gamma > 0$ such that

$$\int_t^{t+T} \Phi(\tau)\Phi^T(\tau)d\tau \geq \gamma I. \tag{3.1}$$

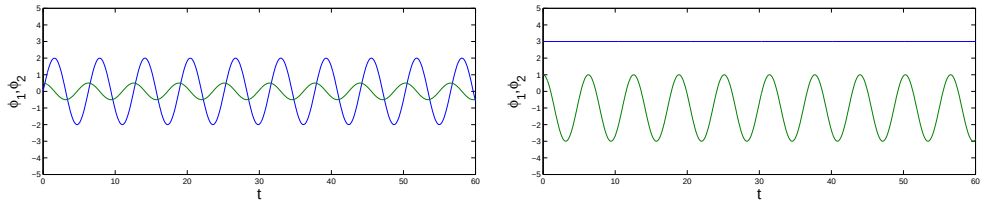
Definition 3.2 A bounded vector signal $\Phi(t)$ is *persistently exciting* if for all

$t > t_0$ there exists $T > 0$ and $\gamma > 0$ such that

$$\int_t^{t+T} \Phi(\tau)\Phi^T(\tau)d\tau \geq \gamma I. \tag{3.2}$$

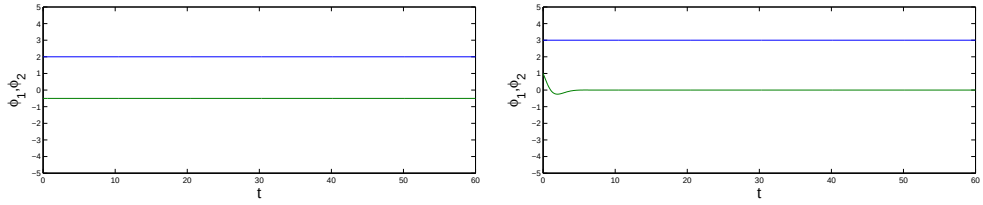
Note that the above definition requires that the matrix $\int_t^{t+T} \Phi(\tau)\Phi^T(\tau)d\tau \in \Re^{m\times m}$ be positive definite over any finite interval. This is equivalent to requiring that over any finite interval the signal $\phi(t)$ contain at least m spectral lines.

Let us consider the two dimensional case as an example. The vector signals $\Phi_1(t) = [2\sin(t)\ 0.5\cos(t)]$ (figure 2(a)) and $\Phi_2(t) = [3\ 2(-0.5+\cos(t))]$ (figure 2(b)) are PE. The vector signal $\Phi_3(t) = [2\ \ -0.5]$ (figure 2(a)) is not exciting over any finite interval, whereas the vector signal $\Phi_4(t) = [3\ 2e^{-t}(-0.5+\cos(t))]$ (figure 2(b)) is exciting over a finite interval, but not PE.



(a) PE signal $\Phi_1(t)$ (b) PE signal $\Phi_2(t)$

Figure 3.1: Two dimensional persistently exciting signals plotted as function of time



(a) Non-PE signal $\Phi_3(t)$ (b) Non-PE signal $\Phi_4(t)$

Figure 3.2: Two dimensional signals that are exciting over an interval, but not persistently exciting

However, the condition on PE reference input (or PE $\Phi(x)$) is restrictive and often infeasible to implement or monitor online. For example, in flight control applications,

PE reference inputs may be operationally unacceptable, waste fuel, and may cause undue stress on the aircraft. Furthermore, since the exogenous reference inputs for many online applications are event based and not known a-priori, it is often impossible to monitor online whether a signal is PE. Consequently, parameter convergence often can not be guaranteed in practice for many adaptive control applications.

3.2 Concurrent Learning for Convergence without Persistence of Excitation

In this thesis we show that if carefully selected and recorded data is used concurrently with current data for adaptation, then the stored information could be used to guarantee convergence without requiring persistency of excitation. Adaptive control laws making such concurrent use of recorded and current data are termed as "Concurrent Learning" adaptive laws. The concurrent use of recorded and current data is motivated by the intuitive argument that if the recorded data is made sufficiently rich, perhaps by recording when the system states *were* exciting for a short period, and used concurrently for adaptation, then weight convergence can occur without the system states being *persistently* exciting. In the following we will present a rank-condition for characterizing the sufficient richness of recorded data and show that this condition is sufficient to guarantee global exponential convergence in adaptive control and parameter estimation problems with structured uncertainties.

3.2.1 A Condition on Recorded Data for Guaranteed Parameter Convergence

The recorded data used in concurrent learning contains carefully selected and stored systems states $\Phi(x_k)$ which are stored in a matrix referred to as the history-stack, and the associated measured output y_k of the system whose parameters are to be estimated (see equation 2.1). The following condition characterizes the richness of recorded data:

Condition 3.1 The history-stack in the recorded data contains as many linearly independent elements $\Phi(x_k) \in \Re^m$ as the dimension of the basis of the uncertainty. That is, if $Z = [\Phi(x_1), ..., \Phi(x_p)]$ denotes the history-stack, then $rank(Z) = m$.

This condition requires that the recorded data contain sufficiently different elements to form a basis for the linearly parameterized uncertainty. This condition differs from the condition on PE $\Phi(t)$ in the following ways:

1. This condition applies to recorded data, whereas persistency of excitation applies to how $\Phi(t)$ should behave in the future.

2. In contrast with persistence of excitation, this condition applies only to a subset of the set of all recorded data, particularly it applies only to data that has been specifically selected and recorded.

3. Since it is fairly straight forward to determine the rank of a matrix online, this condition is conducive to online monitoring.

4. It is straight forward to see that it is always possible to record data such that Condition 3.1 is met when the system states are exciting over a finite time interval.

5. It is also possible to meet this condition by selecting and recording data during a normal course of operation over a longer period without requiring special excitation.

In essence, this condition relates parameter convergence to the spectral properties of the recorded data, and thus, is similar in spirit to Boyd and Sastry's condition which relates the convergence of weights to the spectral properties of future system signals. However, this condition is less restrictive, and conducive to online monitoring.

In the next three sections we will use Lyapunov stability theory to show that Condition 3.1 is sufficient to guarantee parameter convergence in adaptive control

problems without requiring persistence of excitation.

3.3 Guaranteed Convergence in Online Parameter Estimation without Persistency of Excitation

We now present a concurrent learning algorithm for adaptive parameter identification that builds on this intuitive concept, and show that exponential parameter convergence can be guaranteed subject to an easily monitored condition on linear independence of the recorded data. Let $j \in \{1, 2, ...p\}$ denote the index of a recorded data point x_j, let $\Phi(x_j)$ denote the regressor vector evaluated at point x_j, let $\epsilon_j = \nu(\Phi(x_j)) - y_j$, let $\Gamma > 0$ denote a positive definite learning rate matrix, then the concurrent learning gradient descent algorithm is given as

$$\dot{W}(t) = -\Gamma\Phi(x(t))\epsilon(t) - \sum_{j=1}^{p} \Gamma\Phi(x_j)\epsilon_j(t). \tag{3.3}$$

The parameter error dynamics for the concurrent learning gradient descent algorithm can be found by differentiating $\tilde{W}$ and using equation 3.3

$$\begin{aligned}
\dot{\tilde{W}}(t) &= -\Gamma\Phi(x(t))\epsilon(t) - \Gamma\sum_{j=1}^{p} \Phi(x_j)\epsilon_j(t) \\
&= -\Gamma\Phi(x(t))\Phi^T(x(t))\tilde{W}(t) - \Gamma\sum_{j=1}^{p} \Phi(x_j)\Phi^T(x_j)\tilde{W}(t) \\
&= -\Gamma[\Phi(x(t))\Phi^T(x(t)) + \sum_{j=1}^{p} \Phi(x_j)\Phi^T(x_j)]\tilde{W}(t).
\end{aligned} \tag{3.4}$$

This is a linear time varying differential equation in $\tilde{W}$. Furthermore, note that if Condition 3.1 is satisfied, then $\tilde{W} \equiv 0$ is the only equilibrium point for this system. The following theorem shows that once Condition 3.1 on the recorded data is met then the concurrent learning gradient descent law of equation 3.3 guarantees exponential parameter convergence.

Theorem 3.1 Consider the system model given by equation 2.1, the online estimation model given by equation 2.2, the concurrent learning gradient descent weight

update law of equation 3.3, and assume that the regressor function $\Phi(x)$ is continuously differentiable and that the measurements $\Phi(x(t)) \in D$ where $D \subset \Re^m$ is a compact set. If the recorded data points satisfy Condition 3.1, then the zero solution of the weight error dynamics of equation 3.4 $\tilde{W} \equiv 0$ is globally uniformly exponentially stable.

Proof Consider the quadratic function given by $V(\tilde{W}) = \frac{1}{2}\tilde{W}(t)^T\Gamma^{-1}\tilde{W}(t)$, and note that $V(0) = 0$ and $V(\tilde{W}) > 0 \ \forall \ \tilde{W} \neq 0$, hence $V(\tilde{W})$ is a Lyapunov function candidate. Since $V(\tilde{W})$ is quadratic, letting $\lambda_{\min}(.)$ and $\lambda_{\max}(.)$ denote the operators that return the minimum and maximum eigenvalue of a matrix, we have: $\lambda_{\min}(\Gamma^{-1})\|\tilde{W}\|^2 \leq V(\tilde{W}) \leq \lambda_{\max}(\Gamma^{-1})\|\tilde{W}\|^2$. Differentiating with respect to time along the trajectories of 3.4 we have

$$\begin{aligned}\dot{V}(\tilde{W}(t)) = -\tilde{W}(t)^T[\Phi(x(t))\Phi^T(x(t)) \\ + \sum_{j=1}^{p} \Phi(x_j)\Phi^T(x_j)]\tilde{W}(t).\end{aligned} \tag{3.5}$$

Since $\Phi(x(t))\Phi^T(x(t)) \geq 0 \ \forall\Phi(x(t))$, this results in

$$\dot{V}(\tilde{W}(t)) \leq -\tilde{W}(t)^T[\sum_{j=1}^{p} \Phi(x_j)\Phi^T(x_j)]\tilde{W}(t) \tag{3.6}$$

Let $\Omega = \sum_{j=1}^{p} \Phi(x_j)\Phi^T(x_j)$, and note that $\sum_{j=1}^{p} \Phi(x_j)\Phi^T(x_j) > 0$ due to Condition 3.1, therefore $\Omega > 0$. Hence

$$\dot{V}(\tilde{W}) \leq -\lambda_{\min}(\Omega)\|\tilde{W}\|^2. \tag{3.7}$$

Hence, using Lyapunov stability theory (see Theorem 4.6 from [34]) uniform exponential stability of the zero solution $\tilde{W} \equiv 0$ of the parameter error dynamics of equation 3.4 is established. Furthermore, since the Lyapunov candidate is radially unbounded, the result is global. ∎

Remark 3.1 The above proof shows exponential convergence of parameter estimation error to zero without requiring persistency of excitation in the signal $\Phi(x(t))$. The proof requires that $\sum_{j=1}^{p} \Phi(x_j)\Phi^T(x_j)$ be positive definite, which is guaranteed if Condition 3.1 is satisfied.

Remark 3.2 The rate of convergence is determined by the spectral properties of $\sum_{j=1}^{p} \Phi(x_j)\Phi^T(x_j)$, which is dependent on the choice of the recorded states; particularly on $\lambda_{\min}(\sum_{j=1}^{p} \Phi(x_j)\Phi^T(x_j))$

3.3.1 Numerical Simulation: Adaptive Parameter Estimation

In this section we present a simple two dimensional example to illustrate the effect of Condition 3.1. Let t denote the time, dt denote a discrete time interval, and for each $t + dt$ let $\theta(t)$ take on incrementally increasing values from $-\pi$ continuing on to 2π with an increment step equal to dt. Let $y = W^{*T}\Phi(\theta)$ be the model of the structured uncertainty that is to be estimated online with $W^* = [0.1, 0.6]$ and $\Phi(\theta) = [1, e^{-|\theta-\pi/2\|^2}]$. We note that y is the output of a RBF Neural Network with a single hidden node, and is assumed to be measured. Figure 3.3 compares the model output y with the estimate ν for the concurrent learning parameter estimation algorithm of Theorem 3.1 and the baseline gradient descent algorithm of equation 2.5. The output of the concurrent learning algorithm is shown by dashed and dotted lines, whereas the output of the baseline algorithm is shown by dotted lines. The concurrent learning gradient descent algorithm outperforms the baseline gradient descent. Figure 3.4 compares the trajectories of the online estimate of the ideal weights in the weight space. The dashed arrows show the scaled magnitude and direction of weight update based only on current data at regular intervals, whereas the solid arrows show the scaled magnitude and direction of weight updates based only on recorded data. It can be seen that at the end of the simulation the concurrent learning gradient descent

algorithm of Theorem 3.1 arrives at the ideal weights (denoted by $*$) while the baseline gradient algorithm does not. On observing the arrows, we see that the weight updates based on both recorded and current data combine two linearly independent directions to improve weight convergence. This illustrates the effect of using recorded data when Condition 3.1 is met. For this simulation the learning rate was set to $\Gamma = 5$ for both concurrent learning and baseline gradient descent case. The regressor vector $\Phi(x(t))$ and the model output $y(t)$ for data points satisfying $W^T(t)\Phi(x(t)) - y(t) > 0.05$ were selected for storage and were used by the concurrent learning algorithm.

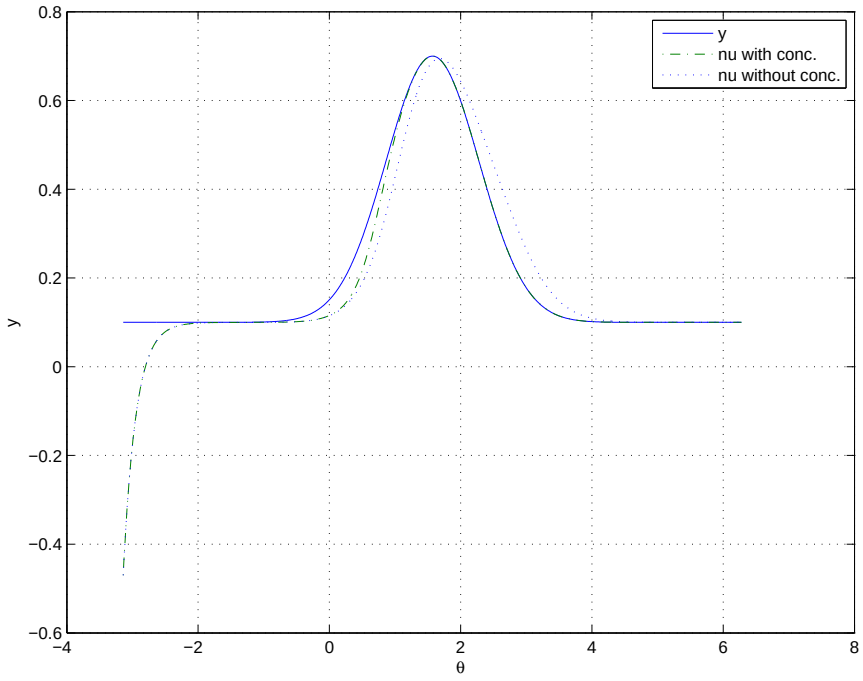


Figure 3.3: Comparison of performance of online estimators with and without concurrent learning, note that the concurrent learning algorithm exhibits a better match than the baseline gradient descent. The improved performance is due to weight convergence.

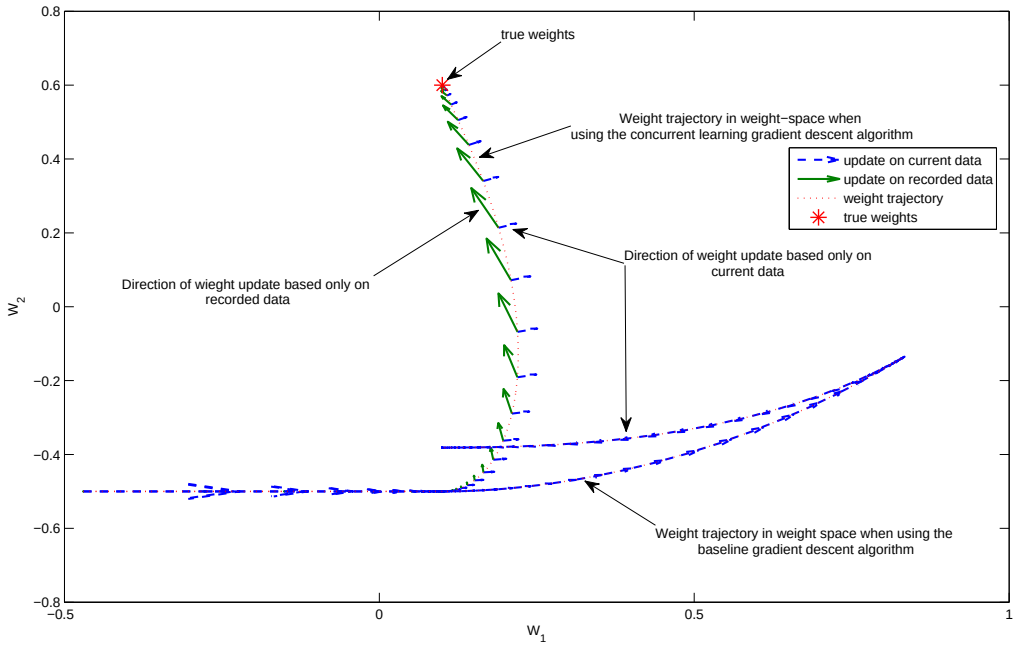


Figure 3.4: Comparison of weight trajectories with and without concurrent learning, note that the concurrent learning algorithm combines two linearly independent directions to arrive at the true weights, while the weights updated by the baseline algorithm do not converge.

3.4 Guaranteed Convergence in Adaptive Control without Persistency of Excitation

In this section, we consider the problem of tracking error and parameter error convergence in the framework of Model Reference Adaptive Control (MRAC). We show that Condition 3.1 is sufficient to guarantee exponential parameter error and tracking error convergence when using a concurrent learning adaptive algorithm without requiring PE reference input. In this section we assume that the uncertainty is linearly parameterized and that its structure is known (Case I in Section 2.2.2, with the uncertainty characterized by equation 2.14). The more general case of unstructured uncertainty (Case II in Section 2.2.3) is handled in the next chapter. Two key theorems that guarantee global tracking error and parameter error convergence to

0 when using the concurrent learning adaptive control method in the framework of MRAC are presented. The first theorem shows that global exponential stability of the tracking error dynamics (equation 2.12) and exponential convergence of the adaptive weights W to their ideal values W^* is guaranteed if Condition 3.1 is satisfied. The second theorem considers the case when adaptation on recorded data is restricted to the nullspace of the adaptation on current data and shows that global asymptotic stability of the tracking error dynamics and asymptotic convergence of the adaptive weights W to their ideal values W^* is guaranteed subject to Condition 3.1. The restriction of adaptation based on recorded data into the nullspace of the adaptation based on current data allows one to prioritize the weight updates based on current data.

Letting for each recorded data point j, $\epsilon_j(t) = W^T(t)\Phi(x_j) - \Delta(x_j)$, a concurrent learning adaptive law that uses both recorded and current data concurrently for adaptation is chosen to have the following form:

$$\dot{W}(t) = -\Gamma_W \Phi(x(t)) e^T(t) PB - \sum_{j=1}^{p} \Gamma_W \Phi(x_j)\epsilon_j(t). \tag{3.8}$$

Remark 3.3 For evaluating the adaptive law of equation 3.8 the term $\epsilon_j = \nu_{ad}(x_j) - \Delta(x_j)$ is required for the j^{th} data point where $j \in [1, 2, ..p]$. The model error $\Delta(x_j)$ can be observed by noting that:

$$\Delta(x_j) = B^T[\dot{x}_j - Ax_j - Bu_j]. \tag{3.9}$$

Since A, B, x_j, u_j are known, the problem of estimating system uncertainty can be reduced to that of estimation of $\dot{x}$ by using 3.9. In cases where an explicit measurement for $\dot{x}$ is not available, $\dot{x_j}$ can be estimated using an implementation of a fixed point smoother [31], we have discussed the details of this process and its implications in [20] and in Appendix A. Note that using fixed point smoothing for estimating $\dot{x}_j$ will entail a finite time delay before ϵ_j can be calculated for that data point. However, since ϵ_j does not directly affect the tracking error at time t, this delay does

not adversely affect the instantaneous tracking performance of the controller. Other methods, such as that suggested in [60] and [97] can also be used to estimate $\dot{x}_j$.

Remark 3.4 In equation 2.6 we assumed that $B = [0, ..., 1]$ for ease of exposition, alternatively, we can require that $B^T B$ is invertible, i.e. B has full column rank. With this requirement, $\Delta(x_j) = (B^T B)^{-1} B^T [\dot{x}_j - Ax_j - Bu_j]$. Note that $B = [0, ..., 1]$ satisfies this requirement trivially. This formulation allow extension to multi-input systems. Extension to multi input systems is performed in Chapter 5.

The weight error dynamics can be found by differentiating $\tilde{W}(t) = W(t) - W^*$:

$$\dot{\tilde{W}}(t) = -\sum_{j=1}^{p} \Phi(x_j)\Phi^T(x_j)\tilde{W}(t) - \Gamma_W \Phi(x(t))e^T(t)PB. \tag{3.10}$$

The following theorem shows that Condition 3.1 is sufficient to guarantee exponential parameter and tracking error convergence when using the concurrent learning adaptive law of equation 3.8.

3.4.1 Guaranteed Exponential Tracking Error and Parameter Error Convergence without Persistency of Excitation

Theorem 3.2 Consider the system in equation 2.6, the reference model in equation 2.7, the control law given by equation 2.8, the case of structured uncertainty with the uncertainty given by $\Delta(x) = W^{*T}\Phi(x)$, the weight update law of equation 3.8, and assume that the recorded data points $\Phi(x_j)$ satisfy Condition 3.1, then the solution $(e(t), W(t)) \equiv (0, W^*)$ of the closed loop system given by equations 2.12 and 3.8 is globally exponentially stable.

Proof Consider the following positive definite and radially unbounded function

$$V(e, \tilde{W}) = \frac{1}{2} e^T P e + \frac{1}{2} \tilde{W}^T {\Gamma_W}^{-1} \tilde{W}, \tag{3.11}$$

since $V(0, 0) = 0$ and $V(e, \tilde{W}) > 0 \ \forall \ (e, \tilde{W}) \neq 0$, $V(e, \tilde{W})$ is a Lyapunov candidate. Let $\xi = [e, \tilde{W}]$, and let $\lambda_{\min}(.)$ and $\lambda_{\max}(.)$ denote operators that return the smallest

and the largest eigenvalue of a matrix, then we have

$$\begin{aligned}\frac{1}{2}\min(\lambda_{\min}(P), \lambda_{\min}({\Gamma_W}^{-1}))\|\xi\|^2 \leq V(e,\tilde{W}) \\ \leq \frac{1}{2}\max(\lambda_{\max}(P), \lambda_{\max}({\Gamma_W}^{-1}))\|\xi\|^2.\end{aligned} \tag{3.12}$$

Differentiating 3.11 along the trajectory of 2.12, and equation 3.10, and using the Lyapunov equation (equation 2.13) we have

$$\begin{aligned}\dot{V}(e,\tilde{W}) = -\frac{1}{2}e^TQe + e^TPB(u_{ad} - \Delta) \\ + \tilde{W}^T(-\sum_{j=1}^{p}\Phi(x_j)\Phi^T(x_j)\tilde{W} - \Phi(x)e^TPB).\end{aligned} \tag{3.13}$$

Using equations 2.14 and 2.15 to note that $u_{ad}(x) - \Delta(x) = \tilde{W}^T\Phi(x)$, canceling like terms, and simplifying we have

$$\dot{V}(e,\tilde{W}) = -\frac{1}{2}e^TQe - \tilde{W}^T(\sum_{j=1}^{p}\Phi(x_j)\Phi^T(x_j))\tilde{W}. \tag{3.14}$$

Let $\Omega = \sum_{j=1}^{p}\Phi(x_j)\Phi^T(x_j)$, then due to Condition 3.1 $\Omega > 0$. Then, we have

$$\dot{V}(e,\tilde{W}) \leq -\frac{1}{2}\lambda_{\min}(Q)e^Te - \lambda_{\min}(\Omega)\tilde{W}^T\tilde{W}. \tag{3.15}$$

Hence,

$$\dot{V}(e,\tilde{W}) \leq -\frac{\min(\lambda_{\min}(Q), 2\lambda_{\min}(\Omega))}{\max(\lambda_{\max}(P), \lambda_{\max}({\Gamma_W}^{-1}))}V(e,\tilde{W}), \tag{3.16}$$

establishing the exponential stability of the zero solution $(e(t), W(t)) \equiv (0, W^*)$ of the closed loop system given by equations 2.12 and equation 3.8 (using Lyapunov stability theory, see Theorem 3.1 in [34]). Since $V(e,\tilde{W})$ is radially unbounded, the result is global, hence x tracks x_{rm} exponentially and $W(t) \to W^*$ exponentially as $t \to \infty$. ∎

Remark 3.5 The above proof shows exponential convergence of tracking error $e(t)$ and parameter estimation error $\tilde{W}(t)$ to 0 without requiring persistency of excitation in the signal $\Phi(x(t))$. The only condition required is Condition 3.1, which

guarantees that the matrix $\sum_{j=1}^{p} \Phi(x_j)\Phi^T(x_j)$ is positive definite. This condition is easily verified online and is found to be less restrictive than a condition on PE reference input.

Remark 3.6 The inclusion or removal of new data points in equation 3.8 does not affect the Lyapunov candidate. Hence, the Lyapunov candidate serves as a common Lyapunov function, therefore, using Theorem 1 in [62], global uniform exponential stability of the zero solution of the tracking error dynamics $e \equiv 0$ and the weight error dynamics $\tilde{W} \equiv 0$ is guaranteed even when data points are removed or added from the history-stack, as long as Condition 3.1 remains satisfied.

Remark 3.7 The rate of convergence is determined by the spectral properties of Q, P, Γ_W, and Ω, the first three are dependent on the choice of the linear gains K_p and the learning rates, and the last one is dependent on the choice of the recorded data.

3.4.2 Concurrent Learning with Training Prioritization

In Theorem 3.2 the adaptive law did not prioritize weight updates based on the instantaneous tracking error over the weight updates based on recorded data. Such prioritization can be achieved by enforcing separation in the training law by restricting the weight updates based on recorded data to the nullspace of the weight updates based on current data. Such prioritization may prove useful if some elements of the recorded data have become corrupt or irrelevant. To achieve this, we let $\dot{W}_t(t) = \Phi(x(t))e^T(t)PB$, let $I \in \Re^{m\times m}$ denote the identity matrix, and use the following projection operator

$$W_c(t) = \begin{cases} I - \dot{W}_t(t)(\dot{W}_t(t)^T\dot{W}_t(t))^{-1}\dot{W}_t(t)^T & \text{if } \dot{W}_t(t) \neq 0 \\ I & \text{if } \dot{W}_t(t) = 0 \end{cases} \tag{3.17}$$

For this case, the following theorem ascertains that global asymptotic stability

of the zero solution of the tracking error dynamics and the weight error dynamics subject to Condition 3.1.

Theorem 3.3 Consider the system in equation 2.6, the reference model in equation 2.7, the control law given by equation 2.8, the case of structured uncertainty with the uncertainty given by $\Delta(x(t)) = W^{*T}\Phi(x(t))$, the definition of $W_c(t)$ in equation 3.17, and let for each recorded data point j, $\epsilon_j(t) = W^T(t)\Phi(x_j) - \Delta(x_j)$ with $\Delta(x_j) = B^T[\dot{x_j} - Ax_j - Bu_j]$. Furthermore, let for each time t, $N_\Phi(t)$ be the set containing all $\Phi(x_j) \perp \dot{W}_t(t)$, that is $N_\Phi(t) = \{\Phi(x_j) : W_c(t)\Phi(x_j) = \Phi(x_j)\}$, and consider the following weight update law

$$\dot{W}(t) = -\Gamma_W \Phi(x(t))e^T(t)PB - \Gamma_W W_c(t) \sum_{j \in N_\Phi} \Phi(x_j)\epsilon_j(t). \tag{3.18}$$

If the recorded data points $\Phi(x_j)$ satisfy Condition 3.1, then the zero solution $(e(t), W(t)) \equiv (0, W^*)$ of the closed loop system given by equations 2.12 and 3.18 are globally asymptotically stable.

Proof Consider the following positive definite and radially unbounded Lyapunov candidate

$$V(e, \tilde{W}) = \frac{1}{2}e^T Pe + \frac{1}{2}\tilde{W}^T {\Gamma_W}^{-1}\tilde{W}. \tag{3.19}$$

Differentiating 3.19 along the trajectory of 2.12, noting that $\dot{\tilde{W}}(t) = -\Gamma_W W_c(t) \sum_{j \in N_\Phi} \Phi(x_j)\Phi^T(x_j)\tilde{W}(t) - \Gamma_W \Phi(x(t))e^T(t)PB$, and using the Lyapunov equation (equation 2.13), we have

$$\begin{aligned} \dot{V}(e, \tilde{W}) &= -\frac{1}{2}e^T Qe + e^T PB(u_{ad} - \Delta) \\ &+ \tilde{W}^T(-W_c \sum_{j \in N_\Phi} \Phi(x_j)\Phi^T(x_j)\tilde{W} - \Gamma_W \Phi(x)e^T PB). \end{aligned} \tag{3.20}$$

Using equations 2.14 and 2.15 to note that $u_{ad}(x) - \Delta(x) = \tilde{W}^T\Phi(x)$, canceling like

terms, and simplifying we have

$$\begin{aligned}\dot{V}(e,\tilde{W}) &= -\frac{1}{2}e^T Q e \\ &- \tilde{W}^T(W_c \sum_{j\in N_\Phi} \Phi(x_j)\Phi^T(x_j))\tilde{W}.\end{aligned} \tag{3.21}$$

Note that $\tilde{W} \in \Re^m$ can be written as $\tilde{W}(t) = (I - W_c(t))\tilde{W}(t) + W_c(t)\tilde{W}(t)$, where W_c is the orthogonal projection operator given in equation 3.17, furthermore note that $W_c^2(t) = W_c(t)$ and $(I - W_c(t))W_c(t) = 0$. Hence we have

$$\begin{aligned}\dot{V}(e,\tilde{W}) &= -\frac{1}{2}e^T Q e \\ &- \tilde{W}^T W_c \sum_{j\in N_\Phi} \Phi(x_j)\Phi^T(x_j) W_c \tilde{W} \\ &- \tilde{W}^T W_c \sum_{j\in N_\Phi} \Phi(x_j)\Phi^T(x_j)(I - W_c)\tilde{W}.\end{aligned} \tag{3.22}$$

However, since the sum in the last term of $\dot{V}(e,\tilde{W})$ is only performed on the elements in N_Φ we have that for all j $\Phi(x_j) = W_c(t)\Phi(x_j)$, therefore it follows that $\tilde{W}^T(t)W_c(t) \sum_{j\in N_\Phi(t)}^{p} W_c(t)\Phi(x_j)\Phi^T(x_j)W_c(t)(I - W_c(t))\tilde{W}(t) = 0$, hence

$$\begin{aligned}\dot{V}(e,\tilde{W}) &= -\frac{1}{2}e^T Q e \\ &- \tilde{W}^T W_c \sum_{j\in N_\Phi} \Phi(x_j)\Phi^T(x_j) W_c \tilde{W} \leq 0.\end{aligned} \tag{3.23}$$

This establishes Lyapunov stability of the zero solution $e \equiv 0$, $\tilde{W} \equiv 0$ of the closed loop system given by equation 2.12 and 3.18. To show asymptotic stability, we must show that $\dot{V}(e,\tilde{W}) = 0$ only when $e = 0$ and $\tilde{W} = 0$. Consider the case when $\dot{V}(e,\tilde{W}) = 0$, since Q is positive definite, this means that $e = 0$. Let $e = 0$ and suppose *ad absurdum* there exists a $\tilde{W} \neq 0$ such that $\dot{V}(e,\tilde{W}) = 0$. Since $e = 0$ we have that $\dot{W}_t = 0$, hence from the definition of W_c (equation 3.17) $W_c = I$. Therefore it follows that the set N_Φ contains all the recorded data points, therefore we have that $\tilde{W}^T \sum_{j=0}^{p} \Phi(x_j)\Phi^T(x_j)\tilde{W} = 0$. However, since the recorded data points satisfy Condition 3.1, $\tilde{W}^T \sum_{j=1}^{p} \Phi(x_j)\Phi^T(x_j)\tilde{W} > 0$ for all $\tilde{W} \neq 0$, contradicting the

claim. Therefore, we have shown that $\dot{V}(e, \tilde{W}) = 0$ only when $e = 0$ and $\tilde{W} = 0$. Thus establishing asymptotic stability of the zero solution $(e(t), W(t)) = (0, W^*)$ of the closed loop system given by equations 2.12 and 3.18. Guaranteeing x tracks x_{rm} asymptotically and $W \to W^*$ as $t \to \infty$. Since the Lyapunov candidate is radially unbounded, the result is global.

∎

Remark 3.8 The above proof shows asymptotic convergence of tracking error $e(t)$ and parameter estimation error $\tilde{W}(t)$ without requiring persistency of excitation in the signal $\Phi(x(t))$. The only condition required is Condition 3.1, which guarantees that the matrix $\sum_{j=1}^{p} \Phi(x_j)\Phi^T(x_j)$ is positive definite.

Remark 3.9 The inclusion or removal of new data points in equation 3.18 or the fact that the summation is performed only over the set $N_\Phi(t)$ does not affect the Lyapunov candidate. Hence, the Lyapunov candidate serves as a common Lyapunov function for the switching adaptive law of equation 3.18, therefore, using Theorem 1 in [62], global asymptotic stability of the zero solution of the tracking error dynamics $e \equiv 0$ and the weight error dynamics $\tilde{W} \equiv 0$ is guaranteed even when data points are removed or added from the history-stack, as long as Condition 3.1 remains satisfied.

Remark 3.10 $\dot{V}(e, \tilde{W})$ will remain negative even when N_Φ is empty at time t if $e \neq 0$, in this case an application of Barbalat's lemma yields $e(t) \to 0$ as $t \to \infty$. If $e = 0$, and Condition 3.1 is satisfied, N_Φ cannot remain empty due to the definition of W_c.

Remark 3.11 If $e(t) = 0$ or $\Phi(x(t)) = 0$ and $\tilde{W}(t) \neq 0$, we have that $\dot{V}(e, \tilde{W}) = \tilde{W}^T \sum_{j=0}^{p} \Phi(x_j)\Phi^T(x_j)\tilde{W} < 0$ due to Condition 3.1 and the definition of $W_c(t)$ (equation 3.17). This indicates that parameters will converge to their true values even when the tracking error or system states are not PE.

Remark 3.12 For practical applications the following approximations are useful:

- $N_\Phi = \{\Phi(x_j) : \|W_c(t)\Phi(x_j) - \Phi(x_j)\| < \beta\}$, where β is a small positive constant,
- $W_c(t) = I$ if $|e(t)| < \alpha$ where α is a small positive constant.

These approximations will reduce the asymptotic stability result to that of uniform ultimate boundedness.

3.4.3 Numerical Simulations: Adaptive Control

In this section we present numerical simulation results of adaptive control of an inverted pendulum model. Let θ denote the angular position of the pendulum and δ denote the control input, then the unstable pendulum dynamics under consideration are given by:

$$\ddot{\theta} = \delta + sin(\theta) - |\dot{\theta}|\dot{\theta} + 0.5e^{\theta\dot{\theta}}. \tag{3.24}$$

A second order reference model with natural frequency and damping ration of 1 is used, the linear control is given by $K = [-1.5, -1.3]$, and the learning rate is set to $\Gamma_W = 3.5$. The initial conditions are set to $x(0) = [\theta(0), \dot{\theta}(0)] = [1, 1]$ and $W = 0$. The model uncertainty is given by $y = W^{*T}\Phi(x)$ with $W^* = [-1, 1, 0.5]$ and $\Phi(x) = [sin(\theta), |\dot{\theta}|\dot{\theta}, e^{\theta\dot{\theta}}]$. A step in position ($\theta_c = 1$) is commanded at $t =$ 20 seconds. Figure 3.5 compares the reference model tracking performance of the baseline adaptive control law of equation 2.16, the concurrent learning adaptive law of Theorem 3.2 ($W_c(t) = I$), and the concurrent learning adaptive law Theorem 3.3 ($W_c(t)$ as in 3.17). It can be seen that in both cases the concurrent learning adaptive laws outperform the baseline adaptive law, especially when tracking the step commanded at $t = 20$ seconds. The reason for this becomes clear when we examine the evolution of weights, for both concurrent learning laws, the weights are very close to their ideal values by this time, whereas for the baseline adaptive law, this is not true.

This difference in performance is indicative of the benefit of parameter convergence. We note that in order to make a fair comparison the same learning rate (Γ_W) was used, with this caveat, we note that the concurrent learning adaptive law of Theorem 3.2 outperforms the other two laws. It should be noted that increasing Γ_W for the baseline case will result in an oscillatory response. Furthermore, note that approximately up to 3 seconds the tracking performance of the concurrent learning adaptive law of Theorem 3.3 is similar to that of the baseline adaptive law, indicating that until this time the set N_Φ is empty. As sufficient recorded data points become available such that the set N_Φ starts to become nonempty the performance of the concurrent learning adaptive law of Theorem 3.3 approaches that of the concurrent learning adaptive law of Theorem 3.2. In this simulation, the data points for concurrent adaptation were selected for recording if at time t, $x(t)$ satisfied $\|x_p - x(t)\|/\|x(t)\| > 0.1$, where x_p denotes the last stored data point. This method is a computationally efficient way of ensuring that sufficiently different points are recorded and 3.1 was found to be met within the first 0.06 seconds of the simulation. We note in passing that this MRAC implementation is equivalent to Approximate Model Inversion-MRAC implementation (see Chapter 5) with the approximate inversion model $\nu = \delta$.

3.5 Notes on Implementation

An implementation of concurrent learning adaptive controllers will have the following components:

1. A history-stack or memory bank which holds the recorded data. The recorded data contains carefully selected and stored systems states $\Phi(x_j)$ which are stored in a matrix referred to as the history-stack (the criteria for selecting which $\Phi(x_j)$ to record is discussed in Chapter 6), and the associated measured or estimated $\dot{x}_k$ (see Appendix A for one method to estimate $\dot{x}_j$, other methods have been suggested in [60] and [97]).

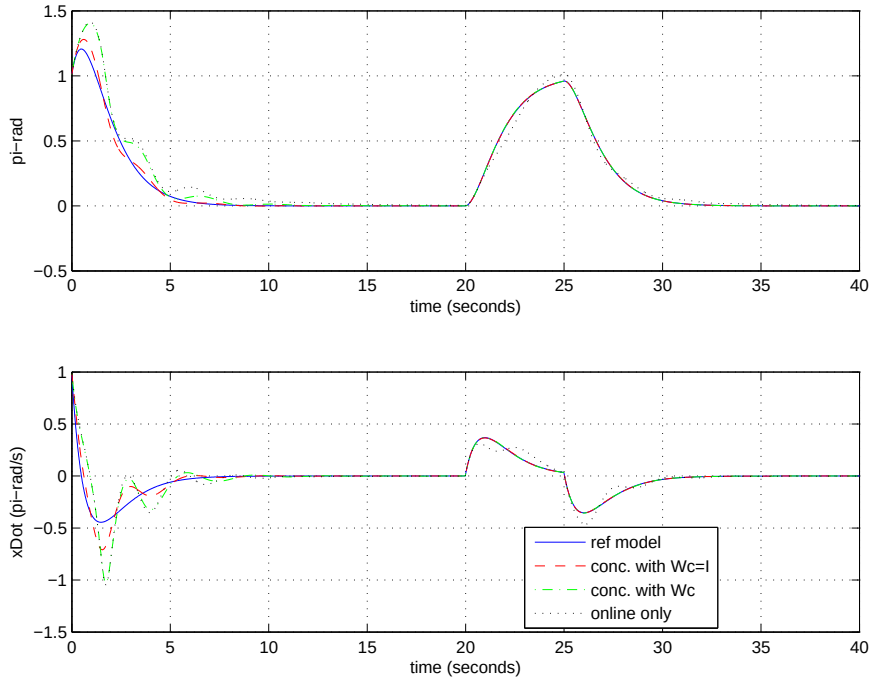


Figure 3.5: Comparison of tracking performance of concurrent learning and baseline adaptive controllers, note that the concurrent learning adaptive controllers outperform the baseline adaptive controller which uses only instantaneous data.

2. An algorithm to select data for recording and an estimate the model error $\Delta(x_j)$ for selected data points (see remark 3.3 for further details),

3. A numeric implementation of the concurrent learning update law (for example equation 3.8).

As an example, an algorithmic implementation of a concurrent learning adaptive controller of Theorem 3.2 is given below. The implementation shown is similar to one used to produce the results in Section 3.4.3. The algorithm begins with assuming that a measurements of $x(t)$ is available.

In the above algorithm, if a measurement of $\dot{x}(t)$ is not available, an estimate can be formed using an appropriate filter, including fixed point smoothers. Fixed point smoothing uses a forward and backward Kalman filter to arrive at an accurate estimate [31]. This means that the algorithm must wait for a small number of time

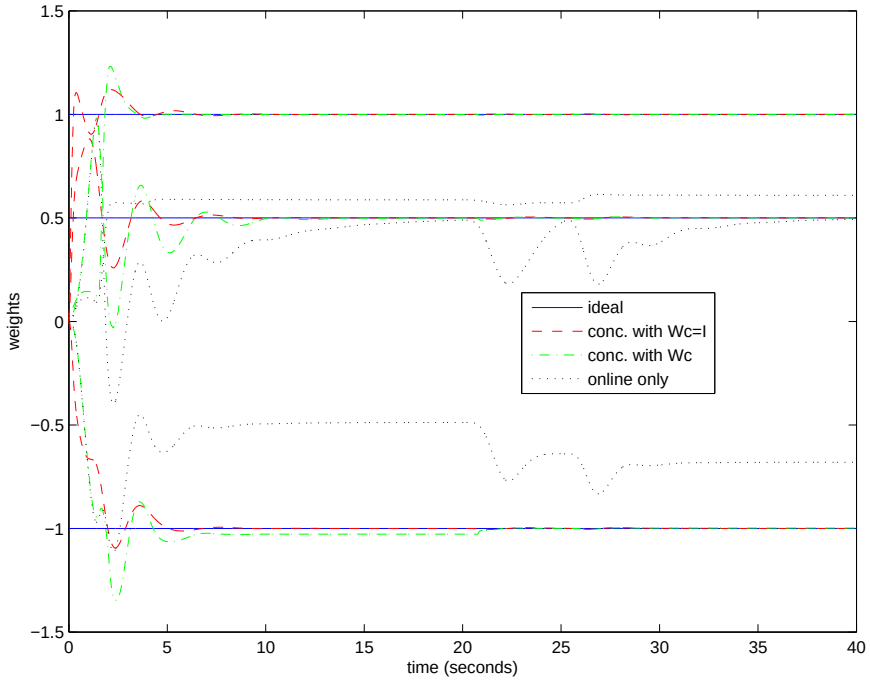


Figure 3.6: Comparison of evolution of adaptive weights when using concurrent learning and baseline adaptive controllers. Note that the weight estimates updated by the concurrent learning algorithms converge to the true weights without requiring persistently exciting exogenous input.

steps until sufficient information is available to use a fixed point smoothing approach. Hence, the incorporation of a selected data point into the history-stack will be slightly delayed. However, this delay does not adversely affect the tracking performance, as the weights continue to be updated so as to minimize the instantaneous tracking error cost ($e^T e$). Figure 3.7 shows a schematic of an implementation of the concurrent learning adaptive controller of Theorem 3.2. The figure serves to depict pictorially algorithm 3.1.

Algorithm 3.1 An algorithmic implementation of concurrent learning adaptive controller of Theorem 3.2

propagate $\dot{x}_{rm}(t)$
$e(t) = x(t) - x_{rm}(t)$
propagate $W(t)$ $\{\dot{W}$ as in equation 3.8$\}$
$u_{ad}(t) = W^T(t)\Phi(x(t))$ {output of the adaptive element}
$u(t) = u_{pd}(t) + u_{rm}(t) - u_{ad}(t)$ {MRAC control law}
if $\frac{\|\Phi(x(t))-\Phi_p\|^2}{\|\Phi(x(t))\|} \geq \epsilon$ **then**
  use a selection criterion (e.g. equation 6.1 or algorithm 6.1) to determine whether to record $\Phi(x(t))$ in the history-stack
  if data point is selected for recording **then**

    if $\dot{x}(t)$ is available **then**
      $\Delta(x(t)) = B^T[\dot{x}_j - Ax_j - Bu_j]$
      $\bar{\Delta}(:, j) = \Delta(x(t))$ {store model error in history-stack}
    else
      initiate fixed point smoother to estimate $\dot{x}(t)$ {use delayed estimate of $\dot{x}(t)$ to estimate $\Delta(x(t))$, see Appendix A}
    end if
  end if
end if

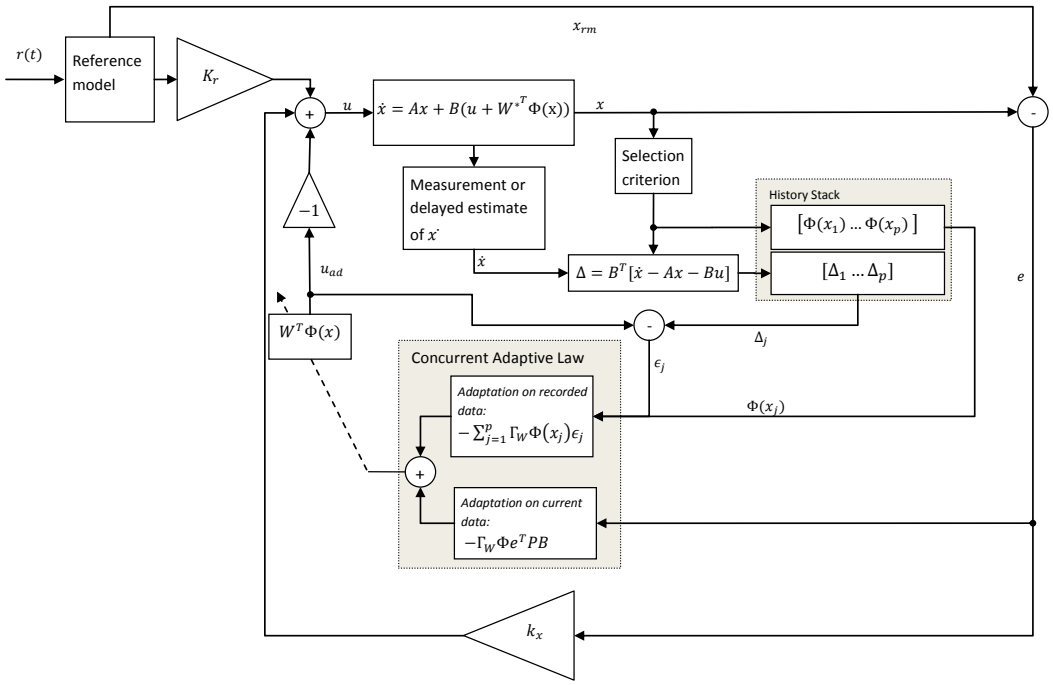


Figure 3.7: Schematic of implementation of the concurrent learning adaptive controller of Theorem 3.2. Note that the history-stack contains $\Phi(x_j)$, which are the data points selected for recording as well as the associated model error formed as described in remark 3.3. The adaptation error ϵ_j for a stored data point is found by subtracting the instantaneous output of the adaptive element from the estimate of the uncertainty. The adaptive law concurrently trains on recorded as well as current data.

CHAPTER IV

CONCURRENT LEARNING NEURO-ADAPTIVE CONTROL

Neural Networks (NN) have been widely used in MRAC to capture the uncertainty in equation 2.12 when the exact structure of the uncertainty $\Delta(x)$ is unknown (Case II in Section 2.2.3, see for example [92], [55], [61], [78], [77], [50], [57], [96], and the references therein). NNs are parameterized function approximators, and they enjoy the desirable universal approximation property which guarantees that any continuous function over a compact domain can be modeled to arbitrary accuracy using a NN if sufficient number of NN nodes are available (see [76] for Radial Basis Function (RBF) NN, and [38] for Single Hidden Layer (SHL) NN). The universal approximation property guarantees a set of unknown ideal weights for a given number of neurons that achieves the aforementioned parametrization. Adaptive laws that drive the adaptive weights towards the ideal weights benefit from the universal approximation property.

However, traditional NN weight adaptation laws do not guarantee that the adaptive weights will approach and stay bounded within a compact neighborhood of the ideal weights if the system signals are not Persistently Exciting (PE). In fact, if the system signals are not PE, then the traditional adaptive laws do not even guarantee boundedness of the adaptive weights. Hence an extra term (such as σ-modification or e-modification) is needed to guarantee boundedness of the adaptive weights. However, both σ-modification or e-modification cause the weights to be restricted within a neighborhood of a preselected value (usually set to 0) which may not necessarily reflect the ideal weights. In this chapter we show that a rank-condition similar to 3.1 is sufficient to guarantee that the adaptive weights stay bounded within a

compact neighborhood of the ideal weights when using concurrent learning adaptive controllers.

Condition 4.1 The recorded data $\sigma(x_j)$ has l linearly independent elements, where l is the dimension of the the RBF basis (equation 2.19). That is, if $Z = [\sigma(x_1), ..., \sigma(x_p)]$, then $rank(Z) = l$.

4.1 Concurrent Learning Neuro-Adaptive Control with RBF NN

Let P be the positive definite solution to the Lyapunov equation 2.13 for a given positive definite Q. Let Γ_W be a positive definite matrix containing the learning rates, let $\zeta(t) = (e(t), \tilde{W}(t))$ be a solution to the closed loop system of equations 2.12 and 4.1 for $t \geq 0$. Let $\beta = \sqrt{\frac{e^T(0)Pe(0)+\tilde{W}^T(0)\Gamma_W^{-1}\tilde{W}(0)}{\min(\lambda_{\min}(P),\lambda_{\min}(\Gamma_W^{-1}))}}$. The following theorem shows that $\zeta(t)$ is uniformly ultimately bounded.

Theorem 4.1 Consider the system in equation 2.6 with the structure of the plant uncertainty unknown and the uncertainty approximated over a compact domain D using a Radial Basis Function NN as in equation 2.19 with $\bar{\epsilon} = \sup_{x\in D} \|\tilde{\epsilon}(x)\|$, the control law of equation 2.8, with u_{ad} given by the output of a RBF NN as in equation 2.17. Let for each recorded data point j, $\epsilon_j(t) = W^T(t)\Phi(x_j) - \Delta(x_j)$, with $\Delta(x_j) = B^T[\dot{x_j} - Ax_j - Bu_j]$, and consider the following weight update law

$$\dot{W}(t) = -\Gamma_W \sigma(x(t))e^T(t)PB - \sum_{j=1}^{p} \Gamma_W \sigma(x_j)\epsilon_j^T(t), \tag{4.1}$$

and assume that the recorded data points $\sigma(x_j)$ satisfy Condition 4.1. Let B_α be the largest compact ball in D, and assume $\zeta(0) \in B_\alpha$, define $\delta = \max(\beta, \frac{2\|PB\|\bar{\epsilon}}{\lambda_{\min}(Q)} + \frac{p\bar{\epsilon}\sqrt{l}}{\lambda_{\min}(\Omega)})$, and assume that D is sufficiently large such that $m = \alpha - \delta$ is a positive scalar. If the exogenous input $r(t)$ is such that the state $x_{rm}(t)$ of the bounded input bounded output reference model of equation 2.7 remains bounded in the compact ball $B_m =$

$\{x_{rm} : \|x_{rm}\| \leq m\}$ for all $t \geq 0$ then the solution of the closed loop system of equations 2.12 and 4.1 $\zeta(t)$ is uniformly ultimately bounded.

Proof Consider the following positive definite and radially unbounded function

$$V(e, \tilde{W}) = \frac{1}{2} e^T P e + \frac{1}{2} \tilde{W}^T {\Gamma_W}^{-1} \tilde{W}. \tag{4.2}$$

Note that $V(0,0) = 0$ and $V(e, \tilde{W}) \geq 0 \ \forall (e, \tilde{W}) \neq 0$ hence 4.2 is a Lyapunov like candidate [34].

Note that since $\nu_{ad}(x_j) - \Delta(x_j) = \tilde{W}^T \sigma(x_j) + \tilde{\epsilon}(x_j)$

$$\dot{\tilde{W}}(t) = -\sum_{j=1}^{p} \sigma(x_j)\sigma^T(x_j)\tilde{W}(t) - \sum_{j=1}^{p} \sigma(x_j)\tilde{\epsilon}(x_j) - \Gamma_W \sigma(x(t)) e^T(t) P B \tag{4.3}$$

Differentiating 4.2 along the trajectory of 2.12, 4.3, and using the Lyapunov equation (equation 2.13), we have

$$\begin{aligned} \dot{V}(e, \tilde{W}) &= -\frac{1}{2} e^T Q e + e^T P B (u_{ad} - \Delta) \\ &+ \tilde{W}^T (-\sum_{j=1}^{p} \sigma(x_j)\sigma^T(x_j))\tilde{W} + \tilde{W}^T (-\sum_{j=1}^{p} \sigma(x_j)\tilde{\epsilon}^T(x_j) - \sigma(x) e^T P B) \end{aligned} \tag{4.4}$$

Canceling like terms, noting that $\nu_{ad}(x) - \Delta(x) = \tilde{W}^T \sigma(x) + \tilde{\epsilon}(x)$, and simplifying we have

$$\dot{V}(e, \tilde{W}) = -\frac{1}{2} e^T Q e - \tilde{W}^T (\sum_{j=1}^{p} \sigma(x_j)\sigma^T(x_j)\tilde{W} + e^T P B \tilde{\epsilon}(x) - \sum_{j=1}^{p} \sigma(x_j)\tilde{\epsilon}(x_j)). \tag{4.5}$$

Let $\Omega = \sum_{j=1}^{p} \sigma(x_j)\sigma^T(x_j)$, then due to Condition 4.1 $\Omega > 0$, using equation 2.19, we have

$$\dot{V}(e, \tilde{W}) \leq -\frac{1}{2} \lambda_{\min}(Q) e^T e - \lambda_{\min}(\Omega) \tilde{W}^T \tilde{W}(t) + e^T P B \bar{\epsilon} - \tilde{W}^T \sum_{j=1}^{p} \sigma(x_j)\tilde{\epsilon}(x_j), \tag{4.6}$$

where $\bar{\epsilon}$ denotes the supremum over all $\tilde{\epsilon}(x)$ for all $x \in D$. Simplifying further and noting that for all $x(t)$ $\|\sigma(x(t))\| \leq \sqrt{l}$ due to the definition of RBF (equation 2.18) we have

$$\dot{V}(e, \tilde{W}) \leq -\frac{1}{2} \lambda_{\min}(Q) \|e\|^2 - \lambda_{\min}(\Omega) \|\tilde{W}\|^2 + \|e^T P B\| \bar{\epsilon} + p \|\tilde{W}\| \bar{\epsilon} \sqrt{l}. \tag{4.7}$$

Let $c_1 = \|PB\|\bar{\epsilon}$, $c_2 = p\bar{\epsilon}\sqrt{l}$ then simplifying further we have

$$\dot{V}(e,\tilde{W}) \leq \|e\|(-\frac{1}{2}\lambda_{\min}(Q)\|e\| + c_1) + \|\tilde{W}\|(-\lambda_{\min}(\Omega)\|\tilde{W}\| + c_2). \tag{4.8}$$

Hence, if $\|e\| > \frac{2c_1}{\lambda_{\min}(Q)}$ and $\|\tilde{W}\| > \frac{c_2}{\lambda_{\min}(\Omega)}$ we have that $\dot{V}(e,\tilde{W}) < 0$. Therefore the set $\Omega_\delta = \{\zeta : \|e\| + \|\tilde{W}\| \leq \frac{2c_1}{\lambda_{\min}(Q)} + \frac{c_2}{\lambda_{\min}(\Omega)}\}$ is positively invariant, hence e and $\tilde{W}$ are ultimately bounded. Let $\delta = \max(\beta, \frac{2c_1}{\lambda_{\min}(Q)} + \frac{c_2}{\lambda_{\min}(\Omega)})$, and $m = \alpha - \delta$. Hence, if the exogenous input $r(t)$ is such that the state $x_{rm}(t)$ of the bounded input bounded output reference model of equation 2.7 remains bounded in the compact ball $B_m = \{x_{rm} : \|x_{rm}\| \leq m\}$ for all $t \geq 0$, then $x(t) \in D\ \forall t$ hence the NN approximation holds and the solution of the closed loop system of equations 2.12 and 4.1 $\zeta(t)$ is uniformly ultimately bounded.

∎

Corollary 4.2 If Theorem 4.1 holds, then the adaptive weights $W(t)$ will approach and remain bounded in a compact neighborhood of the ideal weights W^*.

Proof Since Theorem 4.1 holds the proof follows by noting that $\dot{V}(e,\tilde{W}) \leq 0$ when

$$\|\tilde{W}(t)\| \geq \frac{c_2 + \sqrt{(}p^2\bar{\epsilon}^2 + 4\lambda_{\min}(\Omega)(-\frac{1}{2}\lambda_{\min}(Q)\|e\|^2) + \|e\|c_1)}{2\lambda_{\min}(\Omega)}. \tag{4.9}$$

∎

Remark 4.1 Theorem 4.1 shows ultimate uniform boundedness of weights and tracking error without requiring persistency of excitation or any other robustifying term (such as e-mod, σ-mod or weight projection), subject only to Condition 4.1. The tracking errors and weights are bounded outside of a compact neighborhood of the origin, whose size is dependent on $\bar{\epsilon}$ which in turn is dependent on the number of hidden layer nodes of the RBF NN used. Remarks 3.3 and 3.6 also apply to this theorem.

Remark 4.2 In the proof of Theorem 4.1, we needed to ensure that the exogenous reference input $r(t)$ is such that the reference model remain bounded to ensure that the largest level set remains in the compact domain D over which the NN approximation of equation 2.19 holds. Another approach to arrive at a similar result is presented in [99].

Remark 4.3 The uniform ultimate boundedness properties are dependent on the choice of the linear gains (which determines $\lambda_{\min}(Q)$) and the quality of the recorded data (which determines $\lambda_{\min}(\Omega)$). Appealing to Micchelli's theorem the satisfaction of Condition 4.1 for RBF NN is reduced to selecting distinct points for storage [65], [36]. However, it should be noted that a larger $\lambda_{\min}(\Omega)$ will result in restricting $W(t)$ to a smaller neighborhood of W^* due to Corollary 4.2. Hence recorded data points should be selected to maximize $\lambda_{\min}(\Omega)$.

Remark 4.4 We note that in special cases by making certain assumptions about the uncertainty (such as sector bounded uncertainty in [35]), asymptotic convergence of tracking errors may be shown.

CHAPTER V

EXTENSION TO APPROXIMATE MODEL INVERSION BASED MODEL REFERENCE ADAPTIVE CONTROL OF MULTI-INPUT SYSTEMS

In this chapter we extend concurrent learning adaptive control to Approximate Model Inversion based Adaptive Control (AMI-MRAC) with full state feedback and multiple inputs. AMI-MRAC is an MRAC method that allows the design of adaptive controllers for a general class of nonlinear plants for which an approximate inversion model exists. The main benefits of AMI-MRAC are: 1) Wider class of nonlinear systems (than equation 2.6) for which an approximate inversion model exists can be handled, 2) Matching conditions are implicitly handled through the selection of approximate inversion model, 3) Desired states can be directly commanded through the use of pseudo-control, 4) The estimation of model error (Δ) for recorded data points is simplified, 5)Extension to multi-input multi-output case is relatively simpler, and is performed in this section.

5.1 Approximate Model Inversion based Model Reference Adaptive Control for Multi Input Multi State Systems

Let $x(t) \in \Re^n$ be the known state vector, let $\delta(t) \in \Re^l$ denote the control input, and consider the following feedback stabilizable multiple-input system

$$\dot{x} = f(x(t), \delta(t)), \tag{5.1}$$

where the function f is assumed to be continuously differentiable in x, and control input δ is assumed to be bounded and piecewise continuous. The conditions for the existence and the uniqueness of the solution to 5.1 are assumed to be met.

In AMI-MRAC we are concerned with finding a pseudo-control input $\nu \in \Re^n$ which can be used to find the control input δ such that the plant states track the output of a reference model. If the exact plant model (equation 5.1) is available and invertible, for a given $\nu(t)$, $\delta(t)$ can be found by inverting the plant dynamics. However, since the exact plant model is usually not available or not invertible, we let ν be the output of an approximate inversion model $\hat{f}$ which satisfies the following assumption:

Assumption 5.1 The approximate inversion model $\nu = \hat{f}(x, \delta) : \Re^{n+l} \rightarrow \Re^n$ is continuous and the operator $\hat{f}^{-1} : \Re^{2n} \rightarrow \Re^l$ exists and assigns for every unique element of $\Re^{2n}$ a unique element of $\Re^l$.

Assumption 5.1 is required to guarantee that given a desired pseudo-control input $\nu \in \Re^n$ a control command δ can be found by

$$\delta = \hat{f}^{-1}(x, \nu). \tag{5.2}$$

This approximation results in a model error of the form

$$\dot{x} = \nu + \Delta(x, \delta) \tag{5.3}$$

where the model error Δ is given by:

$$\Delta(x, \delta) = f(x, \delta) - \hat{f}(x, \delta). \tag{5.4}$$

A reference model can be designed that characterizes the desired response of the system

$$\dot{x}_{rm}(t) = f_{rm}(x_{rm}(t), r(t)), \tag{5.5}$$

Where $f_{rm}(x_{rm}(t), r(t))$ denote the reference model dynamics which are assumed to be continuously differentiable in x for all $x \in D_x \subset \Re^n$. The exogenous command $r(t)$ is assumed to be bounded and piecewise continuous, furthermore, it is assumed that

all requirements for guaranteeing the existence of a unique and bounded solution to 2.7 are satisfied for bounded $r(t)$.

The pseudo-control input ν consisting of a linear feedback part $\nu_{pd} = Ke$ with $K \in \Re^{n\times n}$, a linear feedforward part $\nu_{crm} = \dot{x}_{rm}$, and an adaptive part $\nu_{ad}(x, \delta)$ is chosen to have the following form

$$\nu = \nu_{crm} + \nu_{pd} - \nu_{ad}. \tag{5.6}$$

5.1.1 Tracking Error Dynamics

Defining the tracking error e as $e(t) = x_{rm}(t) - x(t)$, and using equation 5.3 the tracking error dynamics can be written as

$$\dot{e} = \dot{x}_{rm} - [\nu + \Delta(x, \delta)]. \tag{5.7}$$

Letting $A = -K$ and using equation 5.6 we have the following tracking error dynamics that are linear in e

$$\dot{e} = Ae + [\nu_{ad}(x, \delta) - \Delta(x, \delta)]. \tag{5.8}$$

Note that the above tracking error dynamics have the same form as the tracking error dynamics of MRAC (equation 2.12). This point of commonality between traditional MRAC and AMI-MRAC allows same weight adaptation laws to be used.

The baseline full state feedback controller ν_{pd} is chosen such that A is a Hurwitz matrix. Hence for any positive definite matrix $Q \in \Re^{n\times n}$, a positive definite solution $P \in \Re^{n\times n}$ exists to the Lyapunov equation

$$A^T P + PA + Q = 0. \tag{5.9}$$

As in the section on MRAC (Section 2.2) the following two cases for characterizing the uncertainty $\Delta(x)$ are considered:

5.1.2 Case I: Structured Uncertainty

Consider the case where it is known that the uncertainty is linearly parameterized and the mapping $\Phi(x)$ is known. This case is captured through the following assumption

Assumption 5.2 The uncertainty $\Delta(x, \delta)$ can be linearly parameterized, that is letting $z = [x^T, \delta^T]^T \in \Re^{n+l}$, there exist a unique matrix of constants $W^* \in \Re^{m \times n}$ and an m dimensional vector of continuously differentiable regressor functions $\Phi(z) = [\phi_1(z), \phi_2(z), ..., \phi_m(z)]^T$ such that there exists an interval $[t, t + \Delta t],\ \Delta t \in \Re^+$ over which the integral $\int_t^{t+\Delta t} \Phi(x(t))\Phi^T(x(t))dt$ can be made positive definite for bounded $\Phi(x(t))$, and $\Delta(z)$ can be uniquely represented as

$$\Delta(z) = W^{*T}\Phi(z). \tag{5.10}$$

In this case letting $W \in \Re^{m \times n}$ denote the estimate of W^*, the adaptive law can be written as

$$\nu_{ad}(z) = W^T\Phi(z). \tag{5.11}$$

For this case it is well known that for a positive definite learning rate Γ_W, the following baseline adaptive law guarantees exponential tracking error and weight convergence if $\Phi(z)$ is PE.

$$\dot{W} = -\Gamma_W\Phi(z)e^T PB \tag{5.12}$$

This case is similar to Case I in Section 2.2.

5.1.3 Case II: Unstructured Uncertainty

In the more general case where it is only known that the uncertainty $\Delta(z)$ is continuous and defined over a compact domain $D \subset \Re^{n+l}$, the adaptive part of the control law (5.6) can be represented using a Radial Basis Function (RBF) or a Single Hidden Layer (SHL) Neural Network(NN). This case is similar to Case II in Section 2.2.

5.1.3.1 Radial Basis Function Neural Network

The output of a RBF NN is given by

$$\nu_{ad}(z) = W^T \sigma(z), \tag{5.13}$$

where $W \in \Re^{q \times n}$ and $\sigma(z) = [1, \sigma_2(z), \sigma_3(z), ..., \sigma_q(z)]^T$ is a q dimensional vector of known radial basis functions (equation 2.18). Appealing to the universal approximation property of RBF NN [76] we have that given a fixed number of radial basis functions q there exists ideal weights $W^* \in \Re^{q \times n}$ and a vector $\tilde{\epsilon} \in \Re^n$ such that the following approximation holds for all $z \in D \subset \Re^{n+l}$ where D is compact

$$\Delta(z) = W^{*T} \sigma(z) + \tilde{\epsilon}(z), \tag{5.14}$$

and $\bar{\epsilon} = \sup_{z \in D} \|\tilde{\epsilon}(z)\|$ can be made arbitrarily small given sufficient number of radial basis functions.

5.1.3.2 Single Hidden Layer Neural Networks

A Single Hidden Layer (SHL) NN is a nonlinearly parameterized map that has also been often used for capturing unstructured uncertainties that are known to be piecewise continuous and defined over a compact domain. Let $\bar{x} = [b_v, \ z^T]^T$ denote the input to the NN with $z = [x^T, \delta^T]^T \in \Re^{n+l}$ and b_v is a constant bias term, then the output of a SHL NN can be given as

$$\nu_{ad}(z) = W^T \sigma(V^T \bar{x}) \in \Re^{n_3}. \tag{5.15}$$

Letting n_2 denote the number of hidden layer nodes and $n_1 = n + l$ denote the number of input layer nodes, $W \in \Re^{(n_2+1) \times n_3}$, and $V \in \Re^{(n_1+1) \times n_2}$ are the NN synaptic weight matrix connecting the hidden layer with the output layer. Note that $\bar{x} \in D \subset \Re^{n_1+1}$, where D is a compact set. The function $\sigma(.)$ denotes the sigmoidal activation function and was described in detail in Section 2.2.3.2.

SHL NN are universal function approximators [38], hence the following approximation holds for all $\bar{x} \in D$

$$\Delta(z) = W^{*^T}\sigma(V^{*^T}\bar{x}) + \tilde{\epsilon}(\bar{x}), \tag{5.16}$$

and $\bar{\epsilon} = \sup_{\bar{x}\in D} \|\tilde{\epsilon}(\bar{x})\|$ can be made arbitrarily small given sufficient number of hidden layer neurons.

For this case it has been shown that the following adaptive laws guarantee guarantees uniform ultimate boundedness of the tracking error, and guarantees that the adaptive weights stay bounded (see for example [61], [55] and the references therein)

Define $r = e^T PB$, where P is the positive definite solution to the Lyapunov equation as defined in 2.13

$$\dot{W} = -(\sigma(\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})r^T\Gamma_W, \tag{5.17}$$

$$\dot{V} = -\Gamma_V\bar{x}r^T W^T\sigma'(V^T\bar{x}), \tag{5.18}$$

where Γ_W, Γ_V are positive definite matrices that define the learning rate of the NN. This update law closely resembles the backpropagation method of tuning NN weights [81, 92, 36, 55]. However, it is important to note that the training signal r is different from that of the backpropagation based learning laws [55].

5.2 Guaranteed Convergence in AMI-MRAC without Persistency of Excitation

The recorded data used in concurrent learning AMI-MRAC includes carefully selected and stored systems states $\Phi(x_k)$ which are stored in a matrix referred to as the history-stack. This section shows that the following condition on linear independence of the recorded data is sufficient to guarantee weight and tracking error convergence in AMI-MRAC adaptive control problems.

Condition 5.1 The history-stack in the recorded data contains as many linearly independent elements as the dimension of the basis of the uncertainty. That is, if $Z = [\Phi(z_1), ..., \Phi(z_p)]$ denotes the history-stack, then $rank(Z) = n + l$.

Note that this condition is equivalent to Condition 3.1 with x_k replaced by z_k.

Letting for each recorded data point j, $\epsilon_j(t) = W^T(t)\Phi(z_j) - \Delta(z_j)$, a concurrent learning adaptive law that uses both recorded and current data concurrently for adaptation is chosen to have the following form

$$\dot{W}(t) = -\Gamma_W \Phi(z(t)) e^T(t) PB - \sum_{j=1}^{p} \Gamma_W \Phi(z_j) \epsilon_j^T(t). \tag{5.19}$$

Remark 5.1 For evaluating the adaptive law of equation 5.19 the term $\epsilon_j = W^T(t)\Phi(z_j) - \Delta(z_j)$ is required for the j^{th} data point where $j \in [1, 2, ..p]$. The model error $\Delta(z_j)$ needs to be recorded along with $\Phi(z_k)$ in the history-stack, and can be observed by using equation 5.4 noting that

$$\Delta(z_j) = \dot{x}_j - \nu(z_j). \tag{5.20}$$

Since $\nu(z_j)$ is known, the problem of estimating system uncertainty can be reduced to that of estimation of $\dot{x}$. In cases where an explicit measurement for $\dot{x}$ is not available, $\dot{x_j}$ can be estimated using an implementation of a fixed point smoother [31]. The details of this process are presented in Appendix A. Note that using fixed point smoothing for estimating $\dot{x}_j$ will entail a finite time delay before ϵ_j can be calculated for that data point. However, since ϵ_j does not directly affect the tracking error at time t, this delay does not adversely affect the instantaneous tracking performance of the controller. Other methods, such as that suggested in [60] and [97] can also be used to estimate $\dot{x}_j$.

Define the weight error as $\tilde{W} = W - W^*$, then the weight error dynamics for the case of can be written as

$$\dot{\tilde{W}}(t) = -\Gamma_W \sum_{j=1}^{p} \Phi(z_j) \Phi^T(z_j) \tilde{W}(t) - \Gamma_W \Phi(z(t)) e^T(t) PB. \tag{5.21}$$

In the following, we will establish the stability of closed loop concurrent learning AMI-MRAC. Due to the commonality between the error dynamics equation for AMI-MRAC (5.8) and MRAC (2.12), the proofs are analogous to the proofs of theorems in Section 2.2; with the key difference being the consideration of multiple inputs. We begin with the following theorem that establishes the global exponential stability of the closed loop concurrent learning AMI-MRAC for the case of structured uncertainty (Case I).

Theorem 5.1 Consider the system in equation 5.1, the reference model in equation 5.5, the inverting controller of equation 5.2, assumption 5.1, the control law of equation 5.6, the case of structured uncertainty with the uncertainty given by equation 5.10, the weight update law of equation 5.19, and assume that the recorded data points $\Phi(z_j)$ satisfy Condition 5.1, then the zero solution $(e(t), W) \equiv (0, W^*)$ of the closed loop system given by equations 5.8 and 5.19 is globally exponentially stable.

Proof Let $tr(.)$ denote the trace operator and consider the following quadratic functional

$$V(e, \tilde{W}) = \frac{1}{2} e^T P e + tr(\frac{1}{2} \tilde{W}^T {\Gamma_W}^{-1} \tilde{W}). \tag{5.22}$$

Note that $V(0, 0) = 0$ and $V(e, \tilde{W}) > 0 \, \forall (e, \tilde{W}) \neq 0$, therefore, $V(e, \tilde{W})$ is a Lyapunov candidate. Let $\xi = [e, vec(\tilde{W})]$ where $vec(.)$ is the operator that stacks the columns of a matrix into a vector, and let $\lambda_{\min}(.)$ and $\lambda_{\max}(.)$ denote operators that return the smallest and the largest eigenvalue of a matrix, then we have

$$\begin{aligned} \frac{1}{2} \min(\lambda_{\min}(P), \lambda_{\min}({\Gamma_W}^{-1})) \|\xi\|^2 &\leq V(e, \tilde{W}) \\ &\leq \frac{1}{2} \max(\lambda_{\max}(P), \lambda_{\max}({\Gamma_W}^{-1})) \|\xi\|^2. \end{aligned} \tag{5.23}$$

Differentiating 5.22 along the trajectory of 5.8 and the weight error dynamics of equation 5.21, and using the Lyapunov equation (equation 5.9), we have

$$\begin{aligned}\dot{V}(e,\tilde{W}) &= -\frac{1}{2}e^TQe + e^TPB(u_{ad}-\Delta) \\ &+ tr(\tilde{W}^T(-\sum_{j=1}^{p}\Phi(z_j)\Phi^T(z_j)\tilde{W} - \Phi(z)e^TPB)).\end{aligned} \tag{5.24}$$

Using equations 5.10 and 5.11 to note that $\nu_{ad}(z(t)) - \Delta(z(t)) = \tilde{W}^T(t)\Phi(z(t))$, canceling like terms and simplifying we have

$$\dot{V}(e,\tilde{W}) = -\frac{1}{2}e^TQe - tr(\tilde{W}^T(\sum_{j=1}^{p}\Phi(z_j)\Phi^T(z_j))\tilde{W}). \tag{5.25}$$

Let $\Omega = \sum_{j=1}^{p}\Phi(z_j)\Phi^T(z_j)$, then due to Condition 5.1 $\Omega > 0$. Hence we have

$$\dot{V}(e,\tilde{W}) \leq -\frac{1}{2}\lambda_{\min}(Q)e^Te - \lambda_{\min}(\Omega)tr(\tilde{W}^T\tilde{W}). \tag{5.26}$$

It follows that

$$\dot{V}(e,\tilde{W}) \leq -\frac{\min(\lambda_{\min}(Q), 2\lambda_{\min}(\Omega))}{\max(\lambda_{\max}(P), \lambda_{\max}({\Gamma_W}^{-1}))}V(e,\tilde{W}), \tag{5.27}$$

establishing the exponential stability of the solution $(e(t), W) \equiv (0, W^*)$ of the closed loop system given by equations 5.8 and 5.19 (using Lyapunov stability theory, see Theorem 3.1 in [34]). Since $V(e,\tilde{W})$ is radially unbounded, the result is global. ∎

Remark 5.2 The above proof shows exponential convergence of tracking error $e(t)$ and parameter estimation error $\tilde{W}(t)$ to 0 without requiring persistency of excitation in the signal $\Phi(z(t))$. The only condition required is Condition 5.1, which guarantees that the matrix $\sum_{j=1}^{p}\Phi(x_j)\Phi^T(x_j)$ is positive definite. This condition is easily verified online and is found to be less restrictive than a condition on PE reference input.

Remark 5.3 The inclusion or removal of new data points in equation 3.8 does not affect the Lyapunov candidate. Hence, the Lyapunov candidate serves as a common

Lyapunov function, therefore, using Theorem 1 in [62], global uniform exponential stability of the zero solution of the tracking error dynamics $e \equiv 0$ and the weight error dynamics $\tilde{W} \equiv 0$ is guaranteed even when data points are removed or added from the history-stack, as long as Condition 5.1 remains satisfied.

Remark 5.4 The rate of convergence is determined by the spectral properties of Q, P, Γ_W, and Ω, the first three are dependent on the choice of the linear gains K and the learning rate, and the last one is dependent on the choice of the recorded data.

The next theorem considers the case when the updates based on current data are given higher priority by restricting the updates based on recorded data to the nullspace of the updates based on current data.

Theorem 5.2 Consider the system in equation 5.1, the reference model in equation 5.5, the inverting controller of equation 5.2, assumption 5.1, the control law of equation 5.6, the case of structured uncertainty with the uncertainty given by equation 5.10. Let for each recorded data point j, $\epsilon_j(t) = W^T(t)\Phi(z_j) - \Delta(z_j)$, with $\Delta(z_j) = \dot{x}_j - \nu(z_j)$, and let $W_c(t) = \dot{W}_t(t)(\dot{W}_t^T(t)\dot{W}_t(t))^+\dot{W}_t(t)^T$ where $+$ denotes the Moore-Penrose pseudo inverse and $\dot{W}_t$ denotes the baseline adaptive law of equation 5.12. Furthermore, Let for each time t, $N_\Phi(t)$ be the set containing all $\Phi(z_j) \perp range(\dot{W}_t(t))$, that is $N_\Phi = \{\Phi(z_j) : W_c(t)\Phi(z_j) = \Phi(z_j)\}$ and consider the following weight update law

$$\dot{W}(t) = -\Gamma_W \Phi(z(t)) e^T(t) PB - \Gamma_W W_c(t) \sum_{j \in N_\Phi} \Phi(z_j)\epsilon_j^T(t), \tag{5.28}$$

If the recorded data points $\Phi(z_j)$ satisfy Condition 5.1, then the zero solution $(e(t), W(t)) \equiv (0, W^*)$ of the closed loop system given by equations 5.8 and 5.28 are globally asymptotically stable.

Proof Noting that the error dynamics in equation 5.8 have a similar form to that of equation 2.12, the proof can be constructed in an analogous manner to the proof of Theorem 3.3 using the Lyapunov candidate of equation 5.22. ∎

5.3 Guaranteed Boudedness Around Optimal Weights in Neuro-Adaptive AMI-MRAC Control with RBF-NN

In this section we show that a verifiable condition on the linear independence of the recorded data is sufficient to guarantee that the adaptive weights stay bounded within a compact neighborhood of the ideal weights when using concurrent learning AMI-MRAC. As in the previous section, the commonality between the error dynamics equation for AMI-MRAC (5.8) and MRAC (2.12) is used to relate the proofs to those previously presented.

Let P be the positive definite solution to the Lyapunov equation 2.13 for a given positive definite Q. Let Γ_W be a positive definite matrix containing the learning rates. Let $\zeta = [e, vec(\tilde{W})]$ and define $\beta = \sqrt{\frac{e^T(0)Pe(0)+tr(\tilde{W}^T(0)\Gamma_W^{-1}\tilde{W}(0))}{\min(\lambda_{\min}(P),\lambda_{\min}(\Gamma_W^{-1}))}}$.

Theorem 5.3 Consider the system in equation 5.1, the inverting controller of equation 5.2, assumption 5.1, with the structure of the plant uncertainty unknown and the uncertainty approximated over a compact domain D using a Radial Basis Function NN as in equation 5.14 with $\bar{\epsilon} = \sup_{z\in D} \|\tilde{\epsilon}(z)\|$, the control law of equation 5.6, and nu_{ad} given by the output of a RBF NN as in equation 5.13. Let for each recorded data point j, $\epsilon_j(t) = W^T(t)\Phi(z_j) - \Delta(z_j)$, with $\Delta(z_j) = \dot{x}_j - \nu(z_j)$ and consider the following update law for the weights of the RBF NN

$$\dot{W} = -\Gamma_W \Phi(z) e^T PB - \sum_{j=1}^{p} \Gamma_W \Phi(z_j) \epsilon_j^T, \tag{5.29}$$

and assume that if $Z = [\sigma(z_1), ..., \sigma(z_p)]$ then $rank(Z) = l$. Let B_α be the largest compact ball in D, and assume $\zeta(0) \in B_\alpha$, define $\delta = \max(\beta, \frac{2\|PB\|\bar{\epsilon}}{\lambda_{\min}(Q)} + \frac{p\bar{\epsilon}\sqrt{l}}{\lambda_{\min}(\Omega)})$, and assume that D is sufficiently large such that $m = \alpha - \delta$ is a positive scalar. If the

exogenous input $r(t)$ is such that the state $x_{rm}(t)$ of the bounded input bounded output reference model of equation 2.7 remains bounded in the compact ball $B_m = \{x_{rm} : \|x_{rm}\| \leq m\}$ for all $t \geq 0$ then the solution $\zeta(t)$ of the closed loop system of equations 2.12 and 4.1 is uniformly ultimately bounded.

Proof Noting that the error dynamics in equation 5.8 have a similar form to that of equation 2.12, the proof can be constructed in an analogous manner to the proof of Theorem 4.1 using the Lyapunov like candidate of equation 5.22. ∎

Corollary 5.4 If the weight update law of Theorem 5.3 is used and Condition 4.1 is satisfied such that Theorem 5.3 holds, then the adaptive weights $W(t)$ will approach and remain bounded in a compact neighborhood of the ideal weights W^*.

Proof Let $c_1 = \|PB\|\bar{\epsilon}$, $c_2 = p\bar{\epsilon}\sqrt{l}$, since Theorem 5.3 holds the proof follows by noting that $\dot{V}(e, \tilde{W}) \leq 0$ when

$$\|\tilde{W}(t)\| \geq \frac{c_2 + \sqrt{(}p^2\bar{\epsilon}^2 + 4\lambda_{\min}(\Omega)(-\frac{1}{2}\lambda_{\min}(Q)\|e\|^2) + \|e\|c_1)}{2\lambda_{\min}(\Omega)}. \tag{5.30}$$

∎

5.4 Guaranteed Boundedness in Neuro-Adaptive AMI-MRAC Control with SHL NN

In this section, the concurrent learning method is extended to AMI-MRAC control with Single Hidden Layer (SHL) Neural Network (NN). As mentioned in Section 2.2.3.2, SHL NN enjoy the universal approximation property (see [38]) similar to RBF NN, with the main difference being that SHL NN are nonlinearly parameterized.

We being with the following assumptions:

Assumption 5.3 The norm of the ideal weights (W^*, V^*) is bounded by a known positive value,

$$0 < \|Z\|_F \leq \bar{Z}. \tag{5.31}$$

Where $\|.\|_F$ denotes the Frobenious norm, and

$$Z \triangleq \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix} \tag{5.32}$$

The following assumption characterizes the structure of the concurrent learning adaptive law.

Assumption 5.4 Let $\dot{W}_t$, $\dot{V}_t$ denote the weight update based on current data and let $\dot{W}_b$, $\dot{V}_b$ denote the weight updates based on past data. Furthermore, let $W_c(t)$ and $V_c(t)$ be orthogonal projection operators, then the structure of the concurrent learning adaptive law is assumed to have the form

$$\dot{W}(t) = \dot{W}_t(t) + W_c(t)\dot{W}_b(t), \tag{5.33}$$

$$\dot{V}(t) = \dot{V}_t(t) + V_c(t)\dot{V}_b(t), \tag{5.34}$$

Let $i \in \aleph$ denote the index of a stored data point z_i, define $r_{b_i}(t) = \nu_{ad}(z_i) - \hat{\Delta}(z_i)$, where $\hat{\Delta}(z) = \dot{x}_i - \nu_i$. Furthermore, define $\tilde{W}(t) = W(t) - W^*, \tilde{V}(t) = V(t) - V^*$ as the difference between the approximated NN weights and the ideal NN weights. We will use equations 5.17 and 5.18 for online learning, hence consider the following operators $W_c(t)$ and $V_c(t)$

$$\begin{aligned} W_c &= I - \frac{(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})^T}{(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})^T(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x})}, \\ V_c &= I - \frac{\Gamma_V\bar{x}\bar{x}^T\Gamma_V}{\bar{x}^T\Gamma_V\Gamma_V\bar{x}}. \end{aligned} \tag{5.35}$$

Lemma 5.5 $W_c(t)$ and $V_c(t)$ are orthogonal projection operators projecting into the nullspace of $\dot{W}_t(t), \dot{V}_t(t)$ given by equations 5.17 and 5.18 respectively.

Proof Since $W_c(t)$ and $V_c(t)$ are symmetric and idempotent they are orthogonal projection operators [5]. The proof for showing that $W_c(t)$ and $V_c(t)$ project into the nullspace of $\dot{W}_t(t), \dot{V}_t(t)$ follows by noting that $W_c(t)\dot{W}_t(t) = 0$ and $V_c(t)\dot{V}_t(t) = 0$. ∎

Let $r^T = e^T PB$ for ease of exposition, where P is the positive definite solution to the Lyapunov equation 5.9 for a given positive definite Q. Let Γ_W, and Γ_V be a positive definite matrices containing the learning rates, $\zeta(t) = (e(t), W(t), V(t))$ be a solution to the closed loop system of equations 5.8 and 5.36 for $t \geq 0$. Let $\beta = \sqrt{\frac{e^T(0)Pe(0)+\tilde{W}^T(0)\Gamma_W^{-1}\tilde{W}(0)+\tilde{V}^T(0)\Gamma_V^{-1}\tilde{V}(0)}{\min(\lambda_{\min}(P),\lambda_{\min}(\Gamma_W^{-1}),\lambda_{\min}(\Gamma_V^{-1}))}}$. The following theorem shows that $\zeta(t)$ is uniformly ultimately bounded.

Theorem 5.6 Consider the system in equation 5.1, the inverting controller of equation 5.2, assumptions 5.1, 5.3, and 5.4. Assume that the structure of the plant uncertainty is unknown and the uncertainty is approximated over a compact domain D by a SHL NN whose output ν_{ad} is given by equation 5.15. Let $W_c(t)$ and $V_c(t)$ be given by equations 5.35 and consider the following weight update law

$$\begin{aligned}
\dot{W}(t) &= -(\sigma(V^T(t)\bar{x}(t)) - \sigma'(V^T(t)\bar{x}(t))V^T(t)\bar{x}(t))r^T(t)\Gamma_w - k\|e(t)\|W(t) \\
&\quad -W_c(t)\sum_{i=1}^{p}(\sigma(V^T(t)\bar{x}_i) - \sigma'(V^T(t)\bar{x}_i)V^T(t)\bar{x}_i)r_{b_i}^T(t)\Gamma_w, \qquad (5.36) \\
\dot{V}(t) &= -\Gamma_V\bar{x}(t)r^T(t)W^T(t)\sigma'(V^T(t)\bar{x}(t)) - k\|e(t)\|V(t) - \\
&\quad V_c(t)\sum_{i=1}^{p}\Gamma_V\bar{x}_i r_{b_i}^T(t)W^T(t)\sigma'(V^T(t)\bar{x}_i), \qquad (5.37)
\end{aligned}$$

where Γ_V, Γ_W are positive definite matrices and k is a positive constant. Let $\zeta(t) = (e(t), W(t), V(t))$ be a solution to the closed loop system of equations 5.8 and 5.36, assume that $\zeta(0) \in B_\alpha$ where $B_\alpha = \{\zeta : \|\zeta\| \leq \alpha\}$ is the largest compact ball contained in D and $\beta \leq \alpha$. If D is sufficiently large, there exists a positive scalar m such that if the states of the bounded input bounded output reference model of equation 5.5 remain bounded in the compact ball $B_m = \{x_{rm} : \|x_{rm}\| \leq m\}$ then $\zeta(t)$ is uniformly ultimately bounded.

Proof Begin by noting that the sigmoidal activation function, and its derivative can

be bounded as follows

$$\|\sigma(V^T\bar{x})\| \leq b_w + n_2, \tag{5.38}$$

$$\|\sigma'\| \leq \bar{a}(b_w + n_2)(1 + b_w + n_2) = \bar{a}k_1k_2. \tag{5.39}$$

Where $\bar{a}$ is the maximum activation potential, and $k_1 = b_w + n_2, k_2 = 1 + b_w + n_2$ are constants defined above for convenience. The Taylor series expansion of the sigmoidal activation function about the ideal weights can be given by

$$\sigma(V^{*^T}\bar{x}) = \sigma(V^T\bar{x}) + \left.\frac{\partial\sigma(s)}{\partial s}\right|_{s=V^T\bar{x}} (V^{*^T}\bar{x} - V^T\bar{x}) + H.O.T. \tag{5.40}$$

where H.O.T. denote higher order terms. A bound on the H.O.T. can be found by rearranging equation 5.40 and noting that $\tilde{Z} = Z - Z^*$ where Z is as defined in assumption 5.3

$$\begin{aligned}\|H.O.T.\| &\leq \|\sigma(V^{*^T}\bar{x})\| + \|\sigma(V^T\bar{x})\| + \|\sigma'(V^T\bar{x})\|\|\tilde{V}\|\|\bar{x}\| \\ &\leq 2k_2 + \bar{a}k_1k_2\|\bar{x}\|\|\tilde{Z}\|_F.\end{aligned}$$

Using equation 5.16 the error in the NN parametrization can be written as

$$\nu_{ad}(\bar{x}) - \Delta(z) = W^T\sigma(V^T\bar{x}) - W^{*^T}\sigma(V^{*^T}\bar{x}) + \tilde{\epsilon}(x). \tag{5.41}$$

This can be further expanded to

$$\nu_{ad}(\bar{x}) - \Delta(z) = W^T\sigma(V^T\bar{x}) - W^{*^T}\left(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})\tilde{V}^T\bar{x} + H.O.T.\right) + \tilde{\epsilon}(x), \tag{5.42}$$

$$= \tilde{W}^T\left(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x}\right) + W^T\sigma'(V^T\bar{x})\tilde{V}^T\bar{x} + w.$$

Where w is given by,

$$w = \tilde{W}^T\sigma'(V^{*^T}\bar{x})V^{*^T}\bar{x} - W^{*^T}(H.O.T.) + \tilde{\epsilon}, \tag{5.43}$$

bounds on w can now be found,

$$\begin{aligned}\|w\| &\leq \|\tilde{W}^T\|\|\sigma'(V^T\bar{x})\|\|V^*\|\|barx\| + \|W^*\|\|(H.O.T.)\| + \bar{\epsilon}, \\ &\leq \bar{a}k_1k_2\bar{Z}\|\tilde{Z}\|_F\|\bar{x}\| + \bar{Z}(2k_1 + \bar{a}k_1k_2\|\bar{x}\|\|\tilde{Z}\|_F) + \bar{\epsilon}.\end{aligned} \tag{5.44}$$

Letting,

$$c_0 = \bar{\epsilon} + 2\bar{Z}k_1, \tag{5.45}$$

$$c_1 = \bar{a}k_1k_2\bar{Z} + \bar{Z}\bar{a}k_1k_2. \tag{5.46}$$

we have

$$\|w\| \leq c_0 + c_1\|\tilde{Z}\|\|\bar{x}\|. \tag{5.47}$$

To show boundedness of the reference model errors and the NN weights we use a Lyapunov like analysis [34]. A radially unbounded and positive definite [34] Lyapunov like function candidate is

$$L(e, \tilde{W}, \tilde{V}) = \frac{1}{2}e^TPe + \frac{1}{2}tr\left\{\left(\tilde{W}\Gamma_W^{-1}\tilde{W}^T\right)\right\} + \frac{1}{2}tr\left\{\tilde{V}^T\Gamma_V^{-1}\tilde{V}\right\}, \tag{5.48}$$

where $tr\{.\}$ denotes the trace operator. Note that $L(0,0,0) = 0$ and $L(e, \tilde{W}, \tilde{V}) \geq 0 \ \forall (e, \tilde{W}, \tilde{V}) \neq 0$. Differentiating the Lyapunov candidate along the trajectory of equations 5.8 and 5.36, using equation 5.42 and 5.9, and adding and subtracting $\sum_{i=1}^{p}(\nu_{ad}(\bar{x}_i) - \Delta(z_i))^T\|e\|(\nu_{ad}(\bar{x}_i) - \Delta(z_i))$, $tr\left\{k\|e\|W\tilde{W}^T\right\}$, and $tr\left\{k\|e\|V\tilde{V}^T\right\}$ we have

$$\begin{aligned}\dot{L}(e, \tilde{W}, \tilde{V}) = -\frac{1}{2}e^TQe + r^T\left(\tilde{W}^T\left(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x}\right) + W^T\sigma'(V^T\bar{x})\tilde{V}^T\bar{x} + w\right) \\ + tr\left\{\left((\dot{W}_t + W_c\dot{W}_b)\Gamma_w^{-1}\tilde{W}^T\right)\right\} + tr\left\{\left(\tilde{V}^T\Gamma_v^{-1}(\dot{V}_t + V_c\dot{V}_b)\right)\right\} \\ - \sum_{i=1}^{p}(\nu_{ad}(\bar{x}_i) - \Delta(z_i))^T(\nu_{ad}(\bar{x}_i) - \Delta(z_i)) + \sum_{i=1}^{p}(\nu_{ad}(\bar{x}_i) - \Delta(z_i))^T(\nu_{ad}(\bar{x}_i) - \Delta(z_i)) \\ + tr\left\{k\|e\|W\tilde{W}^T\right\} - tr\left\{k\|e\|W\tilde{W}^T\right\} + tr\left\{k\|e\|V\tilde{V}^T\right\} - tr\left\{k\|e\|V\tilde{V}^T\right\}.\end{aligned} \tag{5.49}$$

Using 5.42 to expand $\nu_{ad}(\bar{x}_i) - \Delta(x_i)$ and collecting terms we can set the following terms to zero

$$\begin{aligned}&tr\left\{\left(\left(\sigma(V^T\bar{x}) - \sigma'(V^T\bar{x})V^T\bar{x}\right)r^T + k\|e\|W + \dot{W}_t\Gamma_W^{-1}\right)\tilde{W}^T\right\} = 0, \\ &and \\ &tr\left\{\tilde{V}^T\left(\bar{x}r^TW^T\sigma'(V^T\bar{x}) + k\|e\|V + \Gamma_V^{-1}\dot{V}_t\right)\right\} = 0.\end{aligned} \tag{5.50}$$

and,

$$tr\left\{\left(\sum_{i=1}^{p}\left(\sigma(V^T\bar{x}_i)-\sigma'(V^T\bar{x}_i)V^T\bar{x}_i\right)r_{b_i}^T+W_c\dot{W}_b\Gamma_W^{-1}\right)\tilde{W}^T\right\}=0,$$
$$and \tag{5.51}$$
$$tr\left\{\tilde{V}^T\left(\sum_{i=1}^{p}\bar{x}_i r_{b_i}^T W^T\sigma'(V^T\bar{x}_i)+\Gamma_V^{-1}V_c\dot{V}_b\right)\right\}=0.$$

This leads to

$$\dot{W}_t=(-\left(\sigma(V^T\bar{x})-\sigma'(V^T\bar{x})V^T\bar{x}\right)r^T-k\|e\|W)\Gamma_W, \tag{5.52}$$

and

$$\dot{V}_t=\Gamma_V(-\bar{x}r^TW^T\sigma'(V^T\bar{x})-k\|e\|V). \tag{5.53}$$

$$W_c\dot{W}_b=-\sum_{i=1}^{p}\left(\sigma(V^T\bar{x}_i)-\sigma'(V^T\bar{x}_i)V^T\bar{x}_i\right)r_{b_i}^T\Gamma_W, \tag{5.54}$$

$$V_c\dot{V}_b=-\Gamma_V\sum_{i=1}^{p}\bar{x}_i r_{b_i}^T W^T\sigma'(V^T\bar{x}_i). \tag{5.55}$$

Noting that orthogonal projectors are idempotent and multiplying both sides of equation 5.54 with W_c and V_c respectively we have,

$$W_c\dot{W}_b=W_c\sum_{i=1}^{p}\left(\sigma(V^T\bar{x}_i)-\sigma'(V^T\bar{x}_i)V^T\bar{x}_i\right)r_{b_i}^T\Gamma_W, \tag{5.56}$$

and

$$V_c\dot{V}_b=V_c\Gamma_V\sum_{i=1}^{p}\bar{x}_i r_{b_i}^T W^T\sigma'(V^T\bar{x}_i). \tag{5.57}$$

Summing equation 5.52 with 5.56 and 5.53 with 5.57 we arrive at the required training law of Theorem 5.6. The derivative of the Lyapunov like candidate along the trajectories of the system is now reduced to,

$$\dot{L}(e,\tilde{W},\tilde{V})=-\frac{1}{2}e^TQe+r^Tw-\sum_{i=1}^{p}r_{b_i}^Tr_{b_i}+\sum_{i=1}^{p}r_{b_i}^Tw_i \tag{5.58}$$
$$-tr\left\{k\|e\|W\tilde{W}^T\right\}-tr\left\{k\|e\|V\tilde{V}^T\right\}.$$

which can be further bounded as:

$$\dot{L}(e,\tilde{W},\tilde{V}) \leq -\frac{1}{2}\lambda_{\min}Q\|e\|^2 + \|r\|\|w\| - \sum_{i=1}^{p}\|r_{b_i}\|^2 + \sum_{i=1}^{p}\|r_{b_i}\|\|w_i\| \\ -k\|e\|\|\tilde{Z}\|_F^2 + k\|e\|\|\tilde{Z}\|_F\bar{Z}. \tag{5.59}$$

using previously computed bounds,

$$\dot{L}(e,\tilde{W},\tilde{V}) \leq -\frac{1}{2}\lambda_{\min}Q\|e\|^2 + \|e\|\|PB\|(c_0 + c_1\|\tilde{Z}\|_F\|\bar{x}\|)) - \sum_{i=1}^{p}\|r_{b_i}\|^2 \\ + \sum_{i=1}^{p}\|r_{b_i}\|(c_0 + c_1\|\tilde{Z}\|_F\|\bar{x}\|) - k\|e\|\|\tilde{Z}\|_F^2 + k\|e\|\|\tilde{Z}\|_F\bar{Z}. \tag{5.60}$$

hence, when $\lambda_{\min}(Q)$, and k are sufficiently large, $\dot{L}(e,\tilde{W},\tilde{V}) \leq 0$ everywhere outside of a compact set. Therefore, the inputs to the NN can be bounded as follows:

$$\|[b_v, x^T]^T\| \leq b_v + x_c. \tag{5.61}$$

With this bound, let $\hat{c}_1 = \bar{a}k_1k_2\bar{Z} + \bar{Z}\bar{a}k_1k_2(b_v + x_c)$, therefore $\|w\| \leq c_0 + \hat{c}_1\|\tilde{Z}\|$.

To see that the set is indeed compact, consider that $\dot{L}(e,\tilde{W},\tilde{V}) \leq 0$ when

$$\|e\| \geq \frac{-a_0 + \sqrt{a_0^2 + 2\lambda_{\min}(Q)(-\sum_{i=1}^{p} -\|r_{b_i}\|^2 + \sum_{i=1}^{p}\|r_{b_i}\|(c_0 + \hat{c}_1\|\tilde{Z}\|_F))}}{\lambda_{\min}(Q)} \tag{5.62}$$

where

$$a_0 = \|PB\|((c_0 + \hat{c}_1\|\tilde{Z}\|_F)) - k\|\tilde{Z}\|_F^2 + k\|\tilde{Z}\|_F\bar{Z}. \tag{5.63}$$

Or

$$\|e\| = 0, \|w_i\| = 0, \tag{5.64}$$

or $\|e\| \neq 0, \sum_{i=1}^{p}\|r_{b_i}\| \neq 0$, and

$$\|\tilde{Z}\| \geq \frac{-b_0 + \sqrt{b_0^2 + 4k\|e\|(-\frac{1}{2}\lambda_{\min}(Q)\|e\|^2 + \|PB\|\|e\|c_0 - \sum_{i=1}^{p}\|r_{b_i}\|^2 + \sum_{i=1}^{p}\|r_{b_i}\|c_0)}}{2k\|e\|} \tag{5.65}$$

where

$$b_0 = (\|e\|\|PB\|\hat{c}_1 + \sum_{i=1}^{p} \|r_{b_i}\|\hat{c}_1 + k\|e\|\bar{Z}). \tag{5.66}$$

Or $\|e\| \neq 0, \|\tilde{Z}\| \neq 0$, and

$$\sum_{i=1}^{p} \|r_{b_i}\| \geq \frac{-(c_0 + \hat{c}_1\|\tilde{Z}\|_F) + \sqrt{(c_0 + \hat{c}_1\|\tilde{Z}\|_F)^2 + 4d_0}}{2}, \tag{5.67}$$

where

$$d_0 = -\frac{1}{2}\lambda_{\min}Q\|e\|^2 + \|e\|\|PB\|(c_0 + \hat{c}_1\|\tilde{Z}\|_F - k\|e\|\|\tilde{Z}\|_F^2 + k\|e\|\|\tilde{Z}\|_F\bar{Z}). \tag{5.68}$$

The curves represented by equations 5.62, 5.65, and 5.67 are guaranteed to intersect. Let Ω_γ denote the compact set formed by the intersection of the curves 5.62, 5.65, and 5.67 and note that Ω_γ is positively invariant. Let $B_\gamma = \{\zeta : \|\zeta\| \leq \gamma\}$ be the smallest compact ball containing Ω_γ. Let $\delta = \max(\beta, \gamma)$, if D is sufficiently large, then $m = \alpha - \delta$ is positive, and guarantees that if $x_{rm} \in B_m \; \forall t$ then $x(t) \in D \; \forall t \geq 0$ the NN approximation of equation 5.16 holds and the solution $\zeta(t)$ of the closed loop system of equations 5.8 and 5.36 is uniformly ultimately bounded. ∎

Remark 5.5 When a data point is added or removed, the discrete change in the Lyapunov function is zero, allowing the Lyapunov candidate to serve as a common Lyapunov function for any number of recorded data points [62]. Hence, addition or removal of data points does not affect the uniform ultimate boundedness.

Remark 5.6 It should be noted that if no concurrent points are stored, then the NN weight adaptation law reduces to that of the traditional NN weight adaptation law 5.17. This indicates that the purely online NN weight adaptation method can be considered as a special case of the more general online and concurrent weight adaptation method.

Remark 5.7 A key point to note is that proof of Theorem 5.6 does not require a specific form W_c, V_c as long as they are orthogonal projection operators mapping

into the nullspace of $\dot{W}_t, \dot{V}_T$ respectively. Hence similar results as those in Theorem 5.6 can be formed for other stable baseline laws and modifications, including sigma modification, Adaptive Loop Recovery (ALR) modification, and projection operator based modifications.

Remark 5.8 Equation 5.67 explicitly guarantees that the model error residual $\nu_{ad}(\bar{x}_i) - \Delta(z_i)$ stays bounded for all data points.

5.5 Illustrative Example

In this section we use the method of Theorem 5.6 for the control of an inverted pendulum system with nonlinearities that are unknown to the inverting controller. The nonlinear system is given as:

$$\ddot{x} = \delta + \sin(\pi x) - |\dot{x}|\,\dot{x} + 0.5e^{x\dot{x}}, \tag{5.69}$$

where δ is the actuator deflection, and $x, \dot{x}$ describe the angular position and the angular velocity of the pendulum respectively. The system is unstable as presented and it can be considered as a good benchmark for a variety of controllers including neuro-adaptive AMI-MRAC. Figure 5.5 shows the phase portrait of the system where the unstable equilibriums can be seen. All of the unstable equilibriums are on the right hand plane. The left-hand plane equilibriums represent the non-inverted states of the pendulum and are hence stable. The approximate inversion model has the simple form $\nu = \delta$. We assume that the measurement of $\ddot{x}$ is not available and that all system outputs are corrupted with Gaussian white noise along with high frequency sinusoidal noise. Consequently, an optimal fixed lag smoother is used to estimate the model error of equation 3.9 for points sufficiently far in the past. We use a cyclic history-stack of 10 data points where the oldest data point is bumped out with the newest data point selected based on how different each point is from the las stored point [19]. This example will serve to highlight the benefits brought out by this novel adaptive control approach.

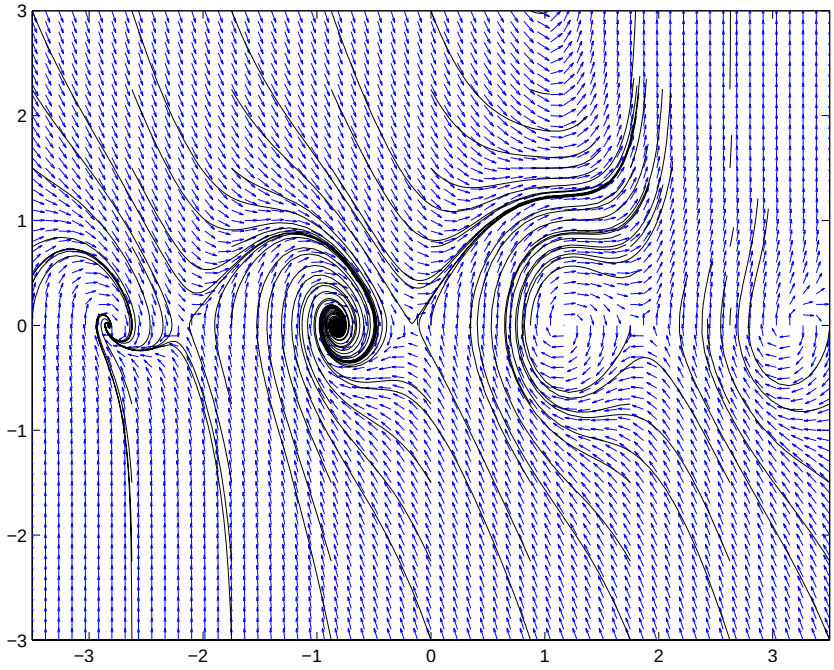
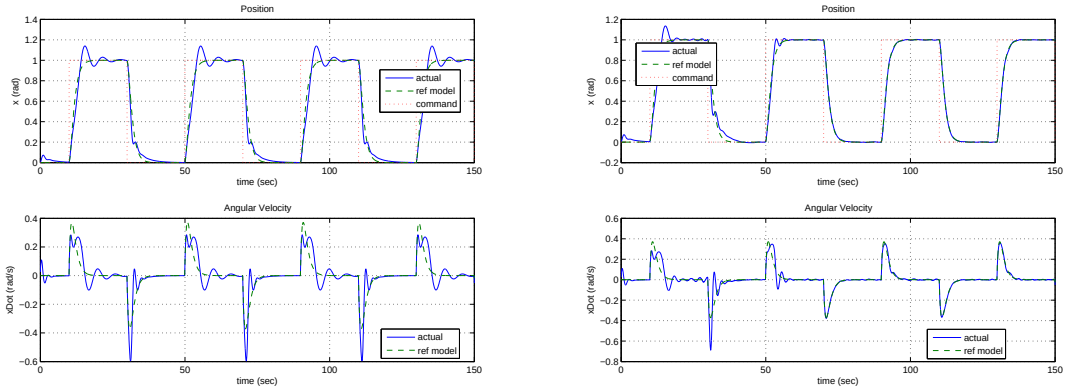


Figure 5.1: Phase Portrait Showing the Unstable Dynamics of the System

One goal of concurrent learning is to show improvement in performance on application of a repeated command. To that effect, 4 repetition of a step command in body position x are commanded to the closed loop system equipped with a SHL-NN based AMI-MRAC controller of Theorem 5.6. The performance of the concurrent learning controller is contrasted with the baseline adaptive controller in Figure 5.2. Figure 2(a) shows the reference model tracking performance of the NN based adaptive controller (without concurrent learning). It is seen that the plant states track the reference model with considerable accuracy, however, no improvement in performance is seen even as the controller tracks the same command. Particularly, the transient overshoot repeats at every step command. This indicates that the adaptive control based purely on current data has no long term memory and does not show an improvement in performance when tracking the same command repeatedly. Figure 2(b) shows the reference model tracking performance of the concurrent learning adaptive controller. It is seen that the transient performance improves over each successive step. Figure

5.3 shows the comparison of the tracking errors with and without concurrent learning controller. It can now be easily seen that without concurrent learning (Figure 3(a)) the errors follow a similar profile every time the controller tracks the step, however with concurrent learning (Figure 3(b)) the tracking error profile reduces through each successive step. Figure 5.4 compares the evolution of the NN weights. It is seen that the NN weights follow a periodic pattern when only online learning controller is used (Figure 4(a)), showing that the adaptive law has no real long term memory, and that it only adapts to the instantaneous dynamics. On the other hand, when concurrent learning adaptive control is used, it is seen that the weights tend to rapidly converge to constant values (Figure 4(b)). Figure 5.5 compares the evolution of the residual vector $r_{bi} = \nu_{ad}(x_i) - \Delta(x_i)$ for the stack of stored points. It is seen that with concurrent learning, the difference between the stored estimate of the model error and the NN estimate of the model error concurrently reduces for all stored data points. This indicates that the NN is able to concurrently adapt to the model error over multiple data points, indicating long term memory, and semi-global error parametrization. In contrast, without concurrent learning (figure 5(a)) we see that the model error residual vector exhibits cyclic behavior and shows little long term improvement.

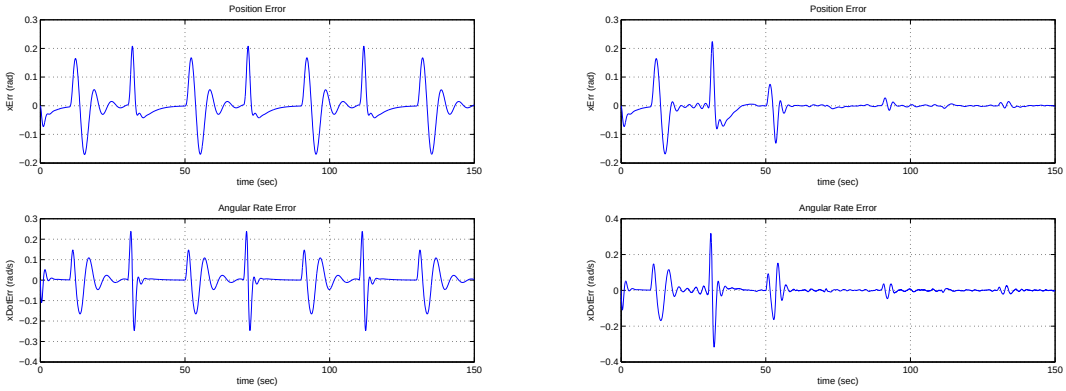
To further characterize the long term learning capabilities of concurrent learning NN, we use weights frozen at the end of the adaptation and compare the NN output (ν_{ad}) with the model error Δ as a function of the state x in Figure 5.6. This plot shows that with concurrent learning it is possible to approximate the unknown model error function with sufficient accuracy over a domain of the state space. This indicates that using concurrent learning, the concurrent learning NN training algorithm of 5.36 has been able to find the required synaptic weights such that an approximation to the nonlinearity over the range of the presented data has been formed. It should be noted that when adaptation based on only current learning is used, the post adaptation NN output is a straight line, which is a result of local learning.



(a) Comparison of States with Only Online Adaptation

(b) Comparison of States with concurrent Learning Adaptive Controller

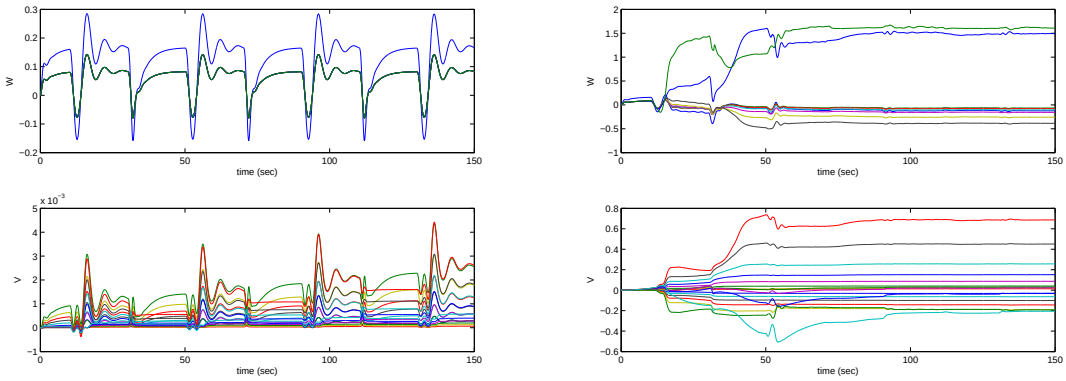
Figure 5.2: Inverted Pendulum, comparison of states vs reference model



(a) Evolution of tracking error with Only Online Adaptation

(b) Evolution of tracking error with concurrent Learning

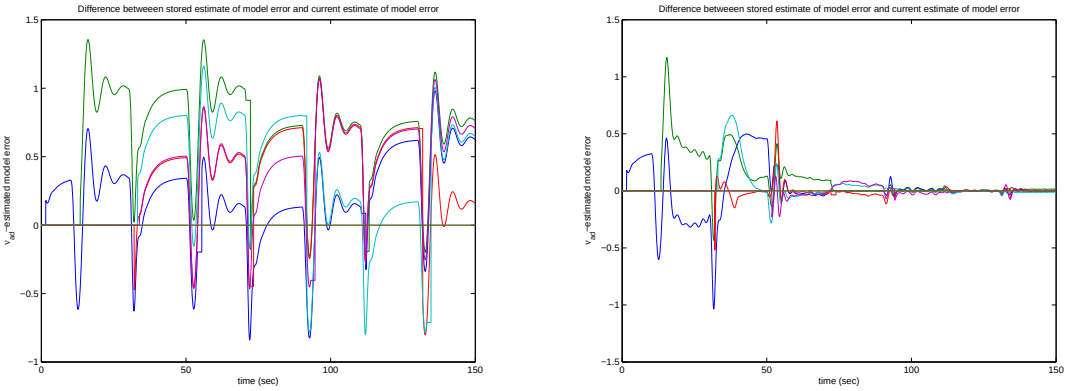
Figure 5.3: Inverted Pendulum, evolution of tracking error



(a) Evolution of NN weights with Only Online Adaptation

(b) Evolution of NN weights with concurrent Learning

Figure 5.4: Inverted Pendulum, evolution of NN weights



(a) Evolution of residual with Only Online Adaptation

(b) Evolution of residual with concurrent Learning

Figure 5.5: Inverted Pendulum, comparison of model error residual $r_{bi} = \nu_{ad}(\bar{x}_i - \Delta(z_i)$ for each stored point in the history-stack.

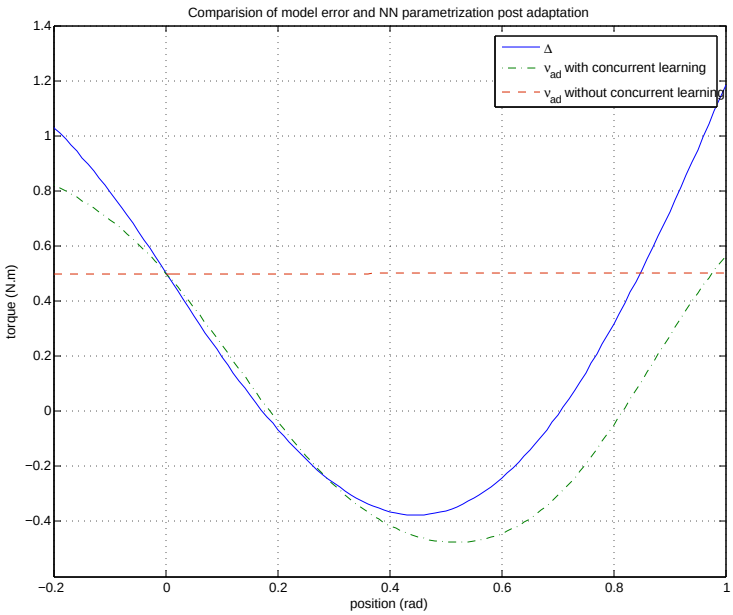


Figure 5.6: Inverted pendulum, NN post adaptation approximation of the unknown model error Δ as a function of x

CHAPTER VI

METHODS FOR RECORDING DATA FOR CONCURRENT LEARNING

The key capability brought about by concurrent learning adaptive controllers is guaranteed parameter error and tracking error convergence to zero without persistency of excitation. Concurrent learning adaptive controllers achieve this by using recorded data concurrently with current data. The recorded data include the regressor vectors $\Phi(x_j)$ which form a basis for the uncertainty $\Delta(x_j)$ in equation 2.6, stored in a matrix referred to as the history-stack, and associated information (such as $\dot{x}_j$) for estimating the model error $\Delta(x_j)$ within a finite time after a data point has been included in the history-stack. In the previous chapters, we showed that convergence can be guaranteed for the case of linearly parameterized uncertainty, if the history-stack meets a rank-condition. This condition requires that the recorded data contain as many linearly independent elements as the dimension of the basis of the uncertainty. Furthermore, in proof of Theorems 3.1 and 3.2 we saw that the rate of convergence depends on the minimum eigenvalue $\lambda_{\min}$ of the symmetric matrix $\Omega = \sum_{j=1}^{p} \Phi(x_j)\Phi^T(x_j)$. Therefore, when implementing concurrent learning adaptive controllers, we wish to record data such that Condition 3.1 is satisfied as soon as possible and that $\lambda_{\min}(\Omega)$ is maximized.

If no previous information about a system is available, or changes to the system have rendered the previously available information inapplicable, then a concurrent learning implementation must begin with no data points in the memory. In this case, a method for selecting data in real-time is needed, in which instantaneous data will be scanned at regular intervals and data points will be selected for recording if they

satisfy selection criteria. We will let $p \in \aleph$ denote the subscript of the last point stored. For ease of exposition, for a stored data point x_j, we let $\Phi_j \in \Re^m$ denote $\Phi(x_j)$, which is the data point to be stored. We will let $Z_k = [\Phi_1, ..., \Phi_p]$ denote the history-stack at time step k. The p^{th} column of Z_k will be denoted by $Z_k(:, p)$. It is assumed that the maximum allowable number of recorded data points is limited due to memory or processing power considerations. Therefore, we will require that Z_k has a maximum of $\bar{p} \in \aleph$ columns, clearly, in order to be able to satisfy Condition 3.1, $\bar{p} \geq m$. For the j^{th} data point, the associated model error $\Delta(x_j)$ is assumed to be stored in the array $\bar{\Delta}(:, j) = \Delta(x_j)$.

6.1 A Simple Method for Recording Sufficiently Different Points

For a given $\epsilon \in \Re^+$ a simple way to select the instantaneous data $\Phi(x(t))$ for recording is to require

$$\frac{\|\Phi(x(t)) - \Phi_p\|^2}{\|\Phi(x(t))\|} \geq \epsilon. \tag{6.1}$$

The above method ascertains that only those data points are selected for storage that are sufficiently different from the last data point stored. In order to meet the dimension of the history-stack, the data can be stored in a cyclic manner. That is if $p = \bar{p}$, then the next data point replaces the oldest data point (Φ_1), and so on. This method has been used previously for selecting data points for recording in Chapter 3, and Chapter 5, and was found to be highly effective.

If the mapping Φ has the properties of a logistic function (see for example [36]) then it is sufficient to pick sufficiently different x_k in order to achieve the same effect as that of equation 6.1. This property is useful when dealing with Neural Network (NN) based adaptive controllers, particularly since in these cases the dimension of Φ is often greater than the dimension of x. Furthermore, as mentioned in remark 4.3, due to Micchelli's theorem, the satisfaction of Condition 4.1 for Radial Basis

Function NN is reduced to selecting distinct points for storage [65], [36]. Hence in this particular case, the criterion in equation 6.1 is an effective and efficient way of selecting data points for recording that meet the rank-condition. However, for general cases, this method does not guarantee that the rank-condition will always be satisfied. Furthermore, this method does not guarantee that $\lambda_{\min}(\Omega)$ is maximized.

6.2 A Singular Value Maximizing Approach

In proof of Theorems 3.1 and 3.2 we saw that the rate of convergence depends on $\lambda_{\min}(\Omega)$. Letting $\sigma(\Omega)$ denote the singular values of Ω, we recall that for nonzero singular values $\sigma(\Omega) = \sqrt{\lambda(\Omega\Omega^T)}$, and Ω is full ranked only if $\sigma_{\min}(\Omega)$ is nonzero [91], [10]. This fact can be used to select data points for storage. The method presented in this section selects a data point for recording if its inclusion results in an increase in the instantaneous minimum singular value of Ω. The following fact ascertains that the singular values of Ω are the same as that of Z_k.

Fact 6.1 $\sigma_{\min}([\Phi_1, ..., \Phi_p]) = \sigma_{\min}(\sum_{j=1}^{p} \Phi_j \Phi_j^T)$

Proof Let $Z_k = [\Phi_1, ..., \Phi_p]$, then we have that $\sigma_{\min}(Z_k) = \sqrt{\lambda_{\min}(Z_k Z_k^T)}$. The proof now follows by noting that $\sum_{j=1}^{p} \Phi_j \Phi_j^T = [\Phi_1, ..., \Phi_p][\Phi_1, ..., \Phi_p]^T = Z_k Z_k^T$. ∎

The following algorithm aims to maximize the minimum singular value of the matrix containing the history-stack. The algorithm begins by using criterion in equation 6.1 to select sufficiently different points for storage. If the number of stored points increases the maximum allowable number, the algorithm seeks to incorporate new data points in such a way that the minimum singular value of Z_k is increased. To achieve this, the algorithm sequentially replaces every recorded data point in the history-stack with the current data point and stores the resulting minimum singular value in a variable. The algorithm then finds the maximum over these values, and

accepts the new data point for storage into the history-stack (by replacing the corresponding existing point) if the resulting configuration results in an increase in the instantaneous minimum singular value of Ω.

Algorithm 6.1 Singular Value Maximizing Algorithm for Recording Data Points

Require: $p \geq 1$
  if $\frac{\|\Phi(x(t))-\Phi_p\|^2}{\|\Phi(x(t))\|} \geq \epsilon$ **then**
    $p = p + 1$
    $Z_k(:, p) = \Phi(x(t))$; {store $\bar{\Delta}(:, p) = \Delta(x(t))$}
  end if
  if $p \geq \bar{p}$ **then**
    $T = Z_k$
    $S_{old} = \min SVD(Z_k^T)$
    for $j = 1$ to p **do**
      $Z_k(:, j) = \Phi(x(t))$
      $S(j) = \min SVD(Z_k^T)$
      $Z_k = T$
    end for
    find $\max S$ and let k denote the corresponding column index
    if $\max S > S_{old}$ **then**
      $Z_k(:, k) = \Phi(x(t))$, {store $\bar{\Delta}(:, k) = \Delta(x(t))$}
      $p = p - 1$
    else
      $p = p - 1$
      $Z_k = T$
    end if
  end if

The method presented in this section attempts to record data points such that $\sigma_{\min}(Z_k)$ is increased. Another interesting approach is to record data points such that the condition number of the matrix Z_k (that is $\frac{\sigma_{\max}(Z_k)}{\sigma_{\min}(Z_k)}$) is brought as close as possible to 1.

6.3 Evaluation of Data Point Selection Methods Through Simulation

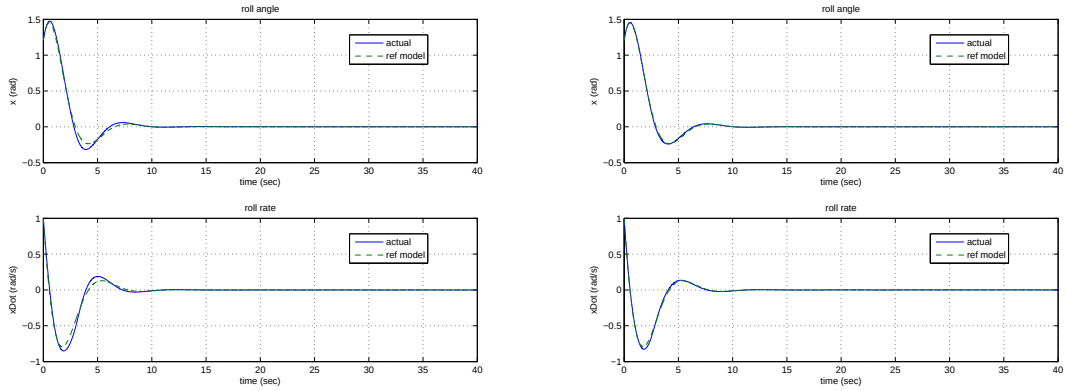
In this section we evaluate the effectiveness of the data point selection criteria through numerical simulation on a wing rock dynamics model. Wing rock is an interesting phenomena which is caused due to asymmetric stalling on lifting surfaces of agile

aircraft. If left uncontrolled, the oscillations caused by wing rock can easily grow unbounded and cause structural damage [66], [83]. Let ϕ denote the roll angle of an aircraft, p denote the roll rate, δ_a denote the aileron control input, then a simplified model for wing rock dynamics is given by [66]

$$\dot{\phi} = p \tag{6.2}$$

$$\dot{p} = \delta_a + \Delta(x), \tag{6.3}$$

where $\Delta(x) = W_0 + W_1\phi + W_2 p + W_3|\phi|p + W_4|p|p + W_5\phi^3$. The parameters for wing rock motion are adapted from [87], they are $W_0 = 0.0, W_1 = 0.2314, W_2 = 0.6918, W_3 = -0.6245, W_4 = 0.0095, W_5 = 0.0214$. Initial conditions for the simulation are arbitrarily chosen to be $\phi = 1.2deg, p = 1deg/s$. The task of the controller is to drive the state to the origin. To that effect, a MRAC controller (see Chapter 2) is used. The reference model chosen is a stable second order linear system with natural frequency of 1 $radian/second$ and damping ratio of 0.5. The linear control gains are given by $K = [2.5, 2.3]$, and the learning rate is set to $\Gamma_W = 2$. The simulation runs for a total time of 40 seconds with an update rate of 0.005 seconds using Euler integration. The reference model tracking performance of the baseline MRAC algorithm (without concurrent learning) is shown in 1(a), while the reference model tracking performance of the concurrent learning MRAC adaptive controller with singular value maximizing data point selection (algorithm 6.1) is shown in figure 1(b). For the chosen learning rate, we note that the concurrent learning adaptive controller is better at tracking the reference model. In this simulation however, we are concerned more with the impact of the selection of data points on weight convergence. To that effect, we will evaluate the different data point selection criterion separately in the following.



(a) Reference model tracking performance of the baseline MRAC adaptive controller without concurrent learning.

(b) Reference model tracking performance of the concurrent learning adaptive controller with singular value maximizing data point selection (see algorithm 6.1).

Figure 6.1: Comparison of reference model tracing performance for the control of wing rock dynamics with and without concurrent learning.

6.3.1 Weight Evolution without Concurrent Learning

Figure 6.2 shows the evolution of weights when using the baseline MRAC controller without concurrent learning. We note that the weights do not converge to their ideal values. Furthermore, once the states arrive at the origin (that is once $\phi = 0,\ p = 0$) the weights are no longer updated. This is expected in a controller that only uses instantaneous data for adaptation.

6.3.2 Weight Evolution with Concurrent Learning using a Static history-stack

For the results presented in this section, we use a static history-stack with a fixed number of slots. The history-stack here is called static because once a data point is recorded, it permanently occupies a slot in the history-stack and cannot overwritten. The data points are selected using the criterion in equation 6.1 with $\epsilon = 0.08$. Figure 6.3 shows the evolution of the weights for a simulation run. It is interesting to note that the weights continue to be updated even after the states arrive at the origin. This is an effect of concurrent training on recorded data. In fact, it can

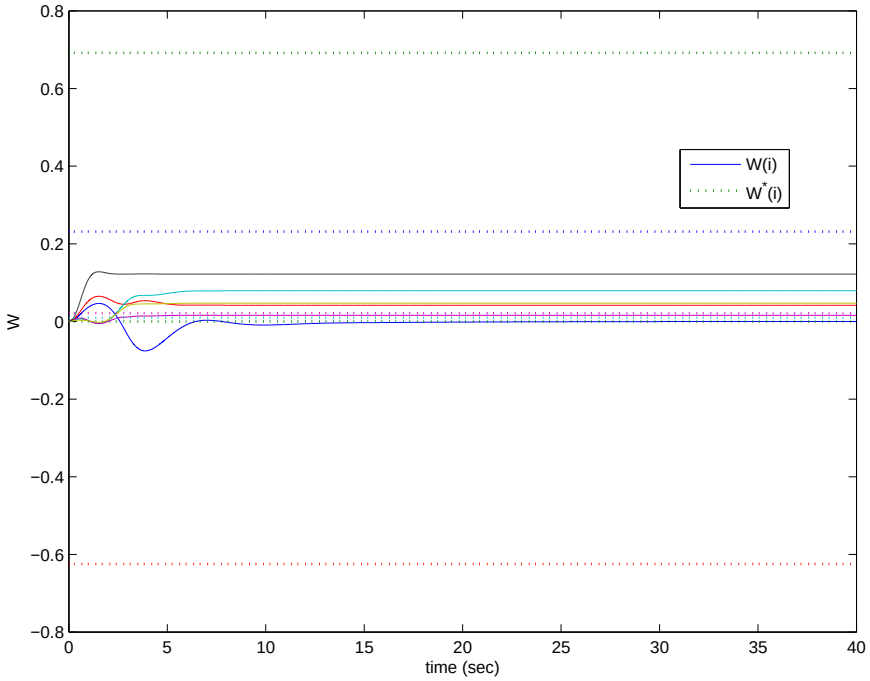


Figure 6.2: Evolution of weight when using the baseline MRAC controller without concurrent learning. Note that the weights do not converge, in fact, once the states arrive at the origin weights remain constant.

be seen that for the chosen learning rate and the data point selection criterion, the weights are approaching their true values, however are not sufficiently close to the ideal values by the end of the simulation. At the end of the simulation it was found that $\sigma_{\min}(\Omega) = 0.0265$

6.3.3 Weight Evolution with Concurrent Learning using a Cyclic history-stack

The history-stack here is called cyclic because data is recorded in a cyclical manner. That is, once the history-stack is full, the newest data point bumps out the oldest data point and so on. This approach aid in guaranteeing that the history-stack reflects the most recently stored data points. The data points are selected using the criterion in equation 6.1 with $\epsilon = 0.08$. Figure 6.4 shows the evolution of the weights for a simulation run. As in the previous case, concurrent learning results in weight update

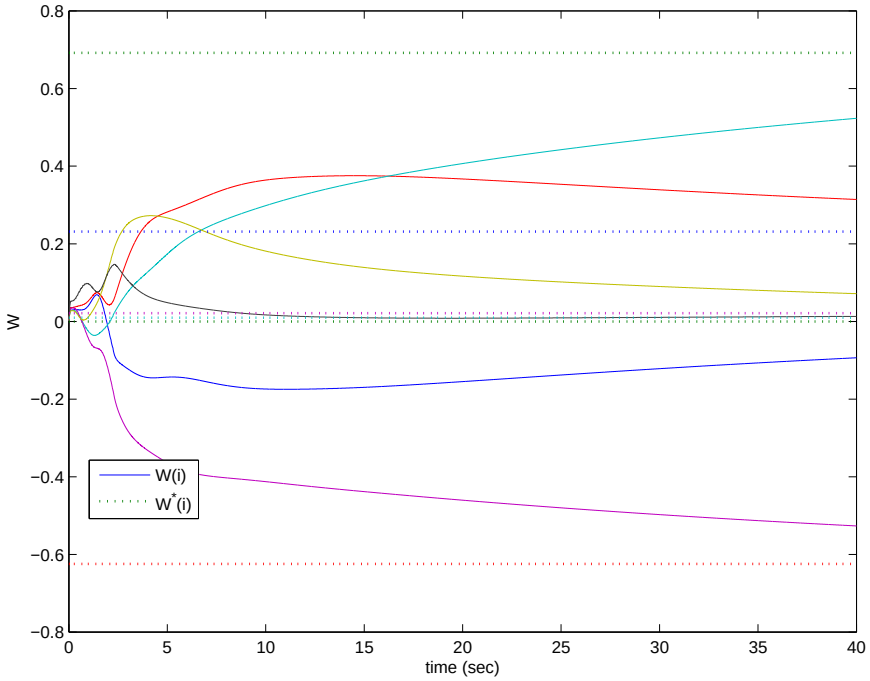


Figure 6.3: Evolution of weight with concurrent learning adaptive controller using a static history-stack. Note that the weights are approaching their true values, however are not close to the ideal value by the end of the simulation (40 seconds).

even after the states arrive at the origin. It can be seen that the weights are closer to their true values than when using a static history-stack. At the end of the simulation it was found that $\sigma_{\min}(\Omega) = 0.0980$.

6.3.4 Weight Evolution with Concurrent Learning using Singular Value Maximizing Approach

In this simulation run, the data points are recorded using algorithm 6.1. Figure 6.5 shows the evolution of the weights for this case. It can be seen that the weights converge to their true values within 20 seconds of the simulation. Furthermore, convergence occurs even when the states have arrived at the origin and are no longer persistently exciting. At the end of the simulation it was found that $\sigma_{\min}(\Omega) = 0.3519$. Figure 6.6 compares $\sigma_{\min}(\Omega)$ at every time step for the three data point selection

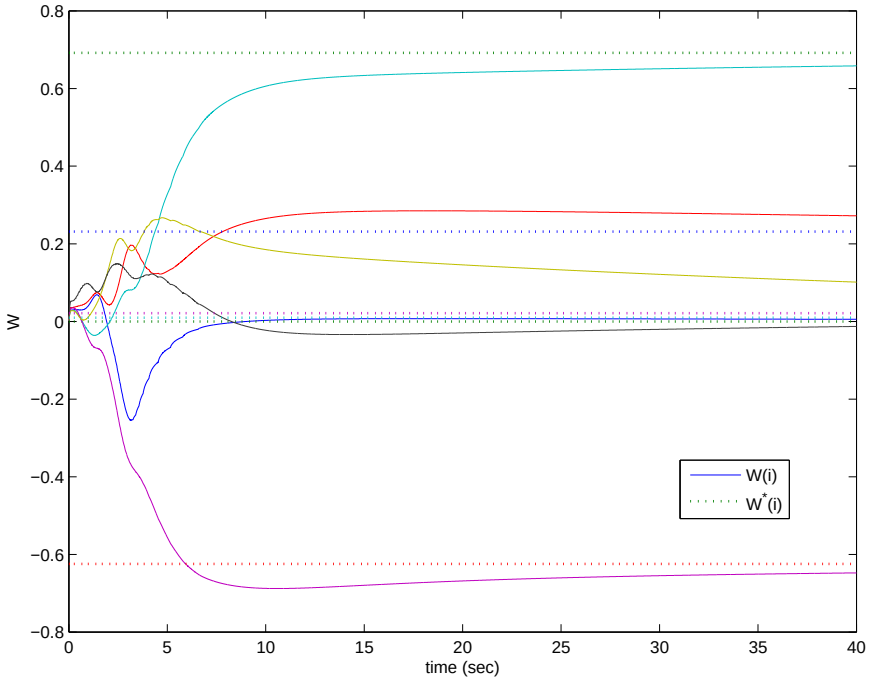


Figure 6.4: Evolution of weight with concurrent learning adaptive controller using a cyclic history-stack. Note that the weights are approaching their true values, and they are closer to their true values than when using a static history-stack within the first 20 seconds of the simulation.

algorithms discussed in this chapter. It can be seen that when using a static history-stack, $\sigma_{\min}(\Omega)$ reaches a constant value and remains there once the history-stack is full. Whereas, when a cyclic history-stack is used, $\sigma_{\min}(\Omega)$ changes as new data replaces old data and occasionally even drops below $\sigma_{\min}(\Omega)$ achieved when using a static history-stack, however by the end of the simulation $\sigma_{\min}(\Omega)$ with a cyclic history-stack is larger than $\sigma_{\min}(\Omega)$ when using a static history-stack. The singular value maximizing algorithm (algorithm 6.1) outperforms both these methods. It can be seen that new data points are selected and old data points removed such that the minimum singular value is maximized. This improvement in the quality of the data is also reflected in weight convergence, with the weights updated by the singular value maximizing approach arriving at their true values faster than the other two approaches.

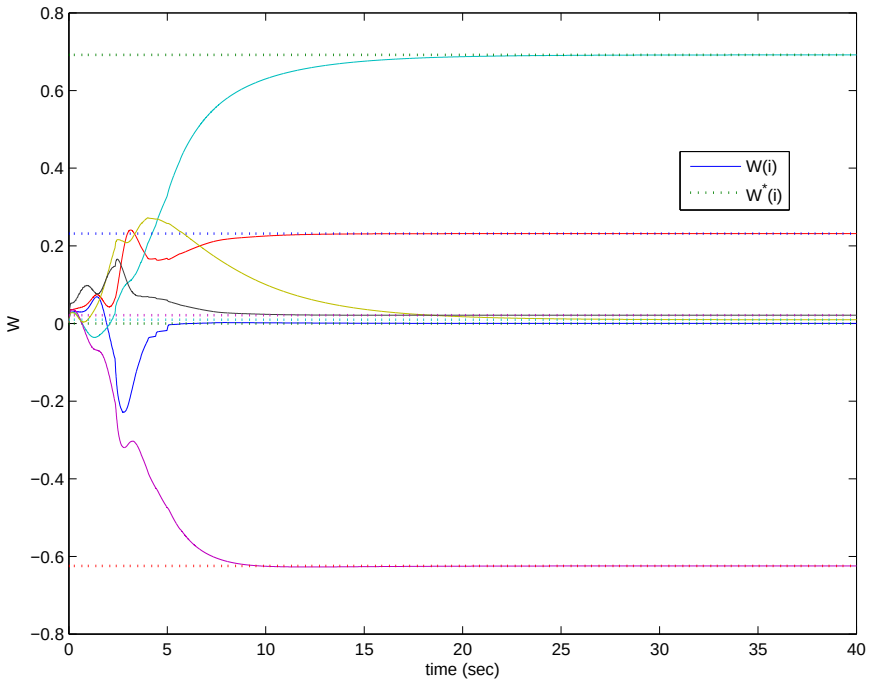


Figure 6.5: Evolution of weight with concurrent learning adaptive controller using the singular value maximizing algorithm (algorithm 6.1). Note that the weights approach their true values by the end of the simulation (40 seconds).

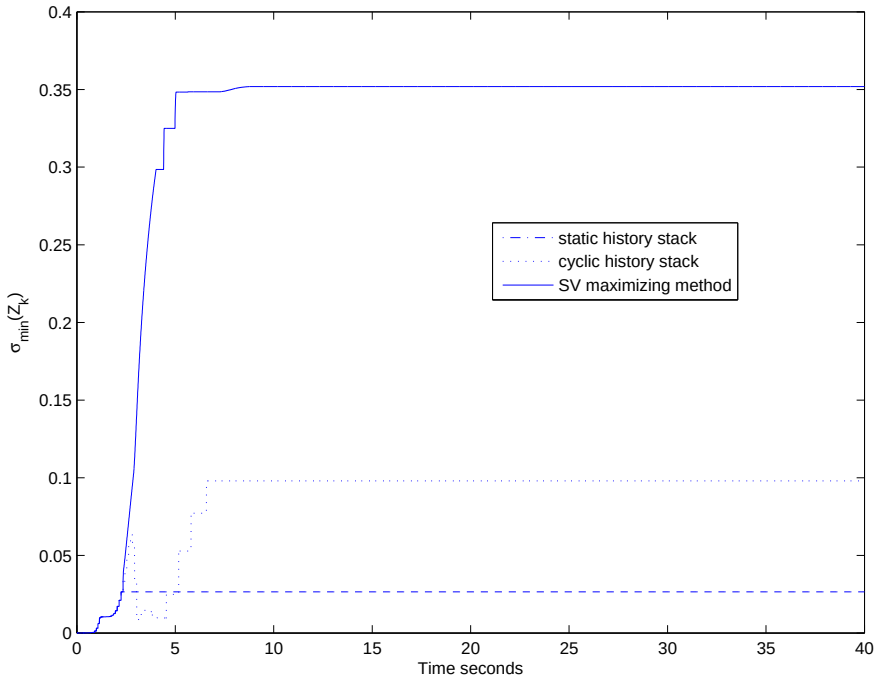


Figure 6.6: Plot of the minimum singular value $\sigma_{\min}(\Omega)$ at every time step for the three data point selection criteria discussed. Note that in case of the static history-stack, $\sigma_{\min}(\Omega)$ stays constant once the history-stack is full, in case of the cyclic history-stack, $\sigma_{\min}(\Omega)$ changes with time as new data replace old data, occasionally dropping below that of the $\sigma_{\min}(\Omega)$ for the static history-stack. When the singular value maximizing algorithm (algorithm 6.1) is used, data points are only selected such that $\sigma_{\min}(\Omega)$ increases with time. This results in faster weight convergence.

CHAPTER VII

LEAST SQUARES BASED CONCURRENT LEARNING ADAPTIVE CONTROL

In this chapter we maintain the idea of using past and current data concurrently for adaptation, however, the adaptation on past data is now performed using an optimal least squares based approach rather than gradient descent. It is well known in the literature that the best linear fit for a given set of data can be obtained by solving the linear least squares problem [10]. Consequently, least squares based method have been widely used for real time parameter estimation [3], [93]. The main contribution of this chapter is the development of a modification term that brings the desirable parameter estimation properties of least squares based algorithms to any baseline gradient based adaptive laws in the framework of model reference adaptive control. The presented least squares based modification term ensures that the adaptive weights converge smoothly to an optimal unbiased estimate of the ideal weights. We show that the modified adaptive law guarantees that exponential tracking error and exponential weight convergence if the stored data are linearly independent. It is interesting to note that both, the gradient based weight update laws studied in Chapters 3 to 5, and the least squares modification studied in this chapter, guarantee convergence subject to an equivalent rank-condition on the recorded data.

7.1 Least Squares Regression

We begin by describing a method by which least squares Regression can be performed online for the MRAC problem studied in Chapter 2. Let N denote the number of recorded state measurements at time t, and θ denote an estimate of the ideal weighs

W^*. For a given data point $k \in 1, 2, ..., N$, the model error $\Delta(k)$ can be observed using the method described in remark 3.3. Furthermore, if the Fourier Transform Regression [67] method is used for solving the least squares problem, then estimation of $\dot{x}$ is further simplified. Details of this method follow.

Define the error $\epsilon(k) = \Delta(x(k)) - \Phi(x(k))^T\theta$, then the error for N discrete data points can be written in vector form as $\epsilon = [\epsilon(1), \epsilon(2), ..., \epsilon(N)]^T$. In order to arrive at the ideal estimate θ of the true weights W^* we must solve the following least squares problem

$$\min_W \epsilon^T \epsilon. \tag{7.1}$$

Let $Y = [\Delta(1), \Delta(2), ..., \Delta(N)]^T$ and define the following matrix

$$X = \begin{bmatrix} \phi_1(x(1)) & \phi_2(x(1)) & ... & \phi_m(x(1)) \\ \phi_1(x(2)) & \phi_2(x(2)) & ... & \phi_m(x(2)) \\ & & & \\ \phi_1(x(N)) & \phi_2(x(N)) & ... & \phi_m(x(N)) \end{bmatrix}. \tag{7.2}$$

A closed form solution to the least squares problem is given as [46]

$$\theta = (X^T X)^{-1} X^T Y. \tag{7.3}$$

Equation 7.3 presents a standard way of solving the Least Squares problem online, however, it suffers from numerical inefficiencies. Fourier Transform Regression (FTR) is a method for solving the least squares problem in the frequency domain [67]. The three main benefits of the FTR approach are: 1) The matrix containing frequency domain information about the stored data has constant dimensions, 2) Available information about the expected frequency range of the data can be used to implicitly filter unwanted frequencies in the data, 3) Fixed point smoothing is not required for the estimation of the model error $\Delta(x)$. Let w denote the independent frequency variable, then the Fourier transform of an arbitrary signal $x(t)$ is given by

$$F[x(t)] = \tilde{x}(w) = \int_{-\infty}^{+\infty} x(t) e^{-jwt} dt. \tag{7.4}$$

Let N be the number of available measurements, and Δt denote the sampling interval, then the discrete Fourier transform can be approximated as

$$X(w) = \sum_{k=0}^{N-1} x(k) e^{-jwk\Delta t}. \tag{7.5}$$

The Euler approximation for the Fourier transform in equation 7.4 is given by

$$\tilde{x}(w) = X(w)\Delta t. \tag{7.6}$$

This approximation is suitable if the sampling rate $1/\Delta t$ is much higher than any of the frequencies of interest w. The discrete version of the Fourier transform can be recursively propagated as follows

$$X_k(w) = X_{k-1}(w) + x(k) e^{-jwk\Delta t}. \tag{7.7}$$

Consider a standard regression problem with complex data, where $\tilde{Y}(w)$ denotes the dependent variable, $\tilde{X}(w)$ denotes the independent variables, $\tilde{\epsilon}$ denotes the regression error in the frequency domain, and Θ denotes the unknown weights

$$\tilde{Y}(w) = \tilde{X}(w)\theta + \tilde{\epsilon}. \tag{7.8}$$

For the problem at hand, given a measurement k and a given frequency range $\omega = 1..l$ the matrix of independent variables is given as

$$\tilde{X}(w) = \begin{bmatrix} \phi_1(x(1)) & \phi_2(x(1)) & ... & \phi_m(x(1)) \\ \phi_1(x(2)) & \phi_2(x(2)) & ... & \phi_m(x(2)) \\ & & & \\ \phi_1(x(l)) & \phi_2(x(l)) & ... & \phi_m(x(l)) \end{bmatrix}. \tag{7.9}$$

The vector of dependent variables is given as $\tilde{Y}(w) = [\Delta(1), \Delta(2), ..., \Delta(l)]^T$. A benefit of using regression in the frequency domain is that the state derivative $\dot{x}_k$ in the frequency domain can be simply given as $\dot{x}_k(w) = jw\tilde{x}_k(w)$. This greatly simplifies the estimation of model error $\Delta(x)$, using equation 3.9, and letting $x(w)$

and $u(w)$ denote the Fourier transform of the state and the input signals, the model error for a data point k in the frequency domain can be found as

$$\Delta_k(w) = B^T[x_k(w)jw - Ax_k(w) - Bu_k(w)]. \tag{7.10}$$

The least squares estimate of the weight vector θ is then given by

$$\theta = [Re(\tilde{X}^*\tilde{X})]^{-1}Re(\tilde{X}^*\tilde{Y}), \tag{7.11}$$

where $*$ denotes the complex conjugate transpose. Note that, forgetting factors can be used to discount older data when the Fourier transform is recursively computed [67].

7.1.1 Least Squares Based Modification Term

We now describe a method by which the least squares estimate of the ideal weights can be incorporated in the adaptive control law. Let $r^T = e^T PB$ where e, P, B are as defined in Section 2.2, let Γ_W, Γ_θ be positive definite matrices denoting the learning rate, and let θ be the solution to the least squares problem of equation 7.3.

The adaptive law for weight estimates W is chosen as

$$\dot{W} = -(\Phi(x)r^T - \Gamma_\theta(W - \theta))\Gamma_W. \tag{7.12}$$

In the above equation, the term $\Gamma_\theta(W - \theta))$ denotes the least squares based modification to the adaptive law. For the case of the structured uncertainty (Section 2.2.2), we have that $\Delta(x) = W^{*T}\Phi(x)$ and the ideal weights W^* are assumed to be constant. Let $\tilde{W} = W - W^*$, then the weight error dynamics are given by

$$\dot{\tilde{W}} = -(\Phi(x)r^T - \Gamma_\theta(W - \theta))\Gamma_W. \tag{7.13}$$

In order to analyze the stability of this adaptive law, we begin with the following condition on the stored data.

Condition 7.1 Enough state measurements are available such that the matrix $\tilde{X}(w)$ of equation 7.9 has full column rank.

Recalling that the matrix $\tilde{X}(w)$ contains Fourier transform of the vector signal $\Phi(x(t))$ we note that Condition 7.1 requires that the stored data points be sufficiently different. In the following, we show that if this condition is satisfied, the adaptive law of equation 7.12 guarantees exponential convergence of tracking error and adaptive weights. We note that this condition is considerably weaker than a condition on persistency of excitation of the vector signal $\Phi(x(t))$ which is required for convergence of weights when using the baseline gradient based adaptive law of equation 2.16. Furthermore, since it is fairly simple to monitor the rank of $\tilde{X}(w)$ online, the fulfilment of this condition is much easier to verify than the condition on persistency of excitation.

Theorem 7.1 Consider the system in equation 2.6, the reference model in equation 2.7, the control law given by equation 2.8, the case of structured uncertainty with the uncertainty given by $\Delta(x) = W^{*T}\Phi(x)$, the weight update law of equation 7.12, and assume that Condition 7.1 is satisfied, then the zero solution $(e(t), W(t)) \equiv (0, W^*)$ of the closed loop system given by equations 2.12 and 7.12 is globally exponentially stable.

Proof Let tr denote the trace operator, and consider the following positive definite and radially unbounded Lyapunov candidate

$$V(e, \tilde{W}) = \frac{1}{2}e^T P e + \frac{1}{2}tr(\tilde{W}^T {\Gamma_W}^{-1} \tilde{W}). \tag{7.14}$$

Taking the time derivative of the Lyapunov candidate along the trajectories of equations 2.12 and 7.13, and using the Lyapunov equation 2.13 results in

$$\begin{aligned} \dot{V}(e, \tilde{W}) = -\frac{1}{2}e^T Q e + r^T(W^T\Phi(x) - W^{*T}\Phi(x)) \\ + tr(\dot{W}{\Gamma_W}^{-1}\tilde{W}^T). \end{aligned} \tag{7.15}$$

Let ϵ be such that $W^* = \theta + \epsilon$, adding and subtracting $(W^T - \theta)^T \Gamma_\theta (W^T - \theta)$ to

equation 7.15 and using the definition of ϵ yields,

$$\begin{aligned}\dot{V}(e,\tilde{W}) = -\frac{1}{2}e^TQe + r^T(\tilde{W}^T\Phi(x)) + tr(\dot{W}{\Gamma_W}^{-1}\tilde{W}^T) \\ + \tilde{W}^T\Gamma_\theta(W-\theta) - \tilde{W}^T\Gamma_\theta(W-\theta).\end{aligned} \tag{7.16}$$

Rearranging yields

$$\begin{aligned}\dot{V}(e,\tilde{W}) = -\frac{1}{2}e^TQe \\ + tr((\dot{W}{\Gamma_W}^{-1} + \Phi(x)r^T + \Gamma_\theta(W-\theta))\tilde{W}^T) \\ - \tilde{W}^T\Gamma_\theta(W-\theta).\end{aligned} \tag{7.17}$$

Setting $tr((\dot{W}{\Gamma_W}^{-1}+\Phi(x)r^T+\Gamma_\theta(W-\theta))\tilde{W}^T) = 0$ yields the adaptive law of equation 7.12. Consider the last term in equation 7.17, we have

$$\begin{aligned}\tilde{W}^T\Gamma_\theta(W-\theta) = &\quad (W-W^*)^T\Gamma_\theta(W-\theta) \\ = &\quad (W-W^*)^T\Gamma_\theta(W-W^*) \\ &+(W-W^*)^T\Gamma_\theta\epsilon.\end{aligned} \tag{7.18}$$

Using 7.11, the definition of ϵ, and Condition 7.1 yields

$$\epsilon = W^* - [Re(\tilde{X}^*\tilde{X})]^{-1}Re(\tilde{X}^*\tilde{X})W^* = 0, \tag{7.19}$$

letting $\lambda_{\min}(Q)$ and $\lambda_{\min}(\Gamma_\theta)$ denote the minimum eigenvalues of Q and Γ_θ we have that equation 7.17 becomes

$$\dot{V}(e,\tilde{W}) \leq -\frac{1}{2}\|e\|^2\lambda_{\min}(Q) - \|\tilde{W}\|^2\lambda_{\min}(\Gamma_\theta). \tag{7.20}$$

Hence, $\dot{V}(e,\tilde{W}) \leq \frac{\min(\lambda_{\min}(Q),2\lambda_{\min}(\Gamma_\theta))}{\max(\lambda_{\max}(P),\lambda_{\max}({\Gamma_W}^{-1}))}V(e,\tilde{W})$. establishing the exponential stability of the zero solution $(e(t),W(t)) \equiv (0,W^*)$ of the closed loop system given by equations 2.12 and equation 7.12 (using Lyapunov stability theory, see Theorem 3.1 in [34]). Since $V(e,\tilde{W})$ is radially unbounded, the result is global. ∎

Remark 7.1 The above proof guarantees exponential stability of the tracking error e and guarantees that W will approach the ideal weight W^* exponentially. This

is subject to Condition 7.1. Considering definition 3.1 it is clear that if the signal is exciting over any finite time interval then data points can be stored such that Condition 7.1 is satisfied. It is interesting to note that Condition 7.1 is similar to the rank-condition 3.1.

Remark 7.2 The above proof can be extended to the case where the uncertainty is unstructured (Section 2.2.3 from Chapter 2) by using Radial Basis Function Neural Networks for approximating the uncertainty. For this case, it is not possible to set $\epsilon = 0$ using equation 7.19 since $Y = W^{*T}\sigma + \tilde{\epsilon}$ and the following adaptive law will result in uniform ultimate boundedness of all states:

$$\dot{W} = -(\sigma(x)r^T - \Gamma_\theta(W - \theta))\Gamma_W. \tag{7.21}$$

Furthermore, referring to equation 2.19 and noting that in this case $\epsilon = \tilde{\epsilon}$, it can be shown that the weights will approach a neighborhood of the best linear approximation of the uncertainty. Finally, in this case, the satisfaction of Condition 7.1 is reduced to selecting distinct points for storage due to Micchelli's theorem [36].

Remark 7.3 Note that the term $\Gamma_\theta(W-\theta)$ adds in as a modification term to the baseline adaptive law of equation 2.16. Since the above analysis is valid for any initial condition and since the baseline adaptive law is known to be uniformly ultimately bounded for the closed loop system of equation 2.12 and 7.12 with $\theta = 0$, it is possible to set $\theta = 0$ until sufficient data is collected online to satisfy Condition 7.1. This will result in a σ-modification like term until satisfaction of assumption 7.1 can be verified online [42].

Remark 7.4 This proof can be modified to accommodate any least squares solution method, for example the standard least squares solution of equation 7.3 can be accommodated by replacing equation 7.19 with the following:

$$\epsilon = W^* - (X^TX)^{-1}X^TXW = 0, \tag{7.22}$$

In this case, Condition 7.1 requires that matrix X has full column rank.

Remark 7.5 The increased computational burden when using the adaptive law of equation 7.12 consists mainly of evaluating equation 7.11 to obtain θ. However, θ does not need to be updated as often as the controller itself.

Remark 7.6 It is possible to imagine a switching approach in which the online estimate of the ideal weights θ is used in equation 2.15 by setting $W = \theta$ when θ becomes available. However, this approaches looses the benefit of keeping the baseline adaptive law in the control loop, namely, the adaptive weights no longer take on values to minimize $V(t) = e^T(t)e(t)$.

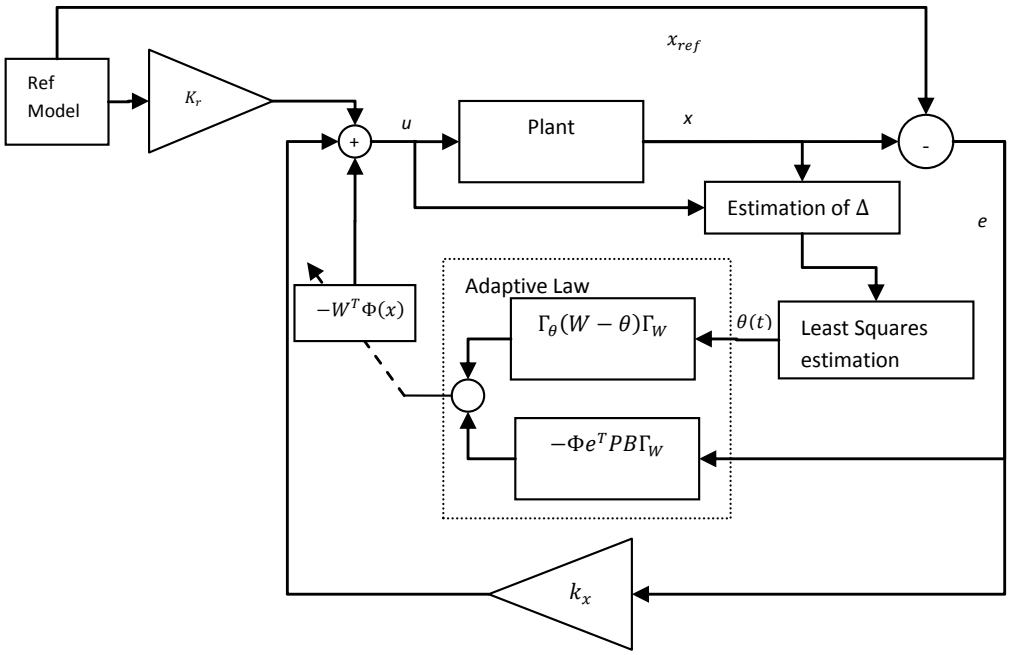


Figure 7.1: Schematics of adaptive controller with least squares Modification

Figure 7.1 shows the schematic of the presented adaptive control method with

least squares modification.

7.2 Simulation results for Least Squares Modification

In this section we use the method of Theorem 7.1 for the control a wing rock dynamics model. Let ϕ denote the roll angle of an aircraft, p denote the roll rate, δ_a denote the aileron control input, then a model for wing rock dynamics is [66]

$$\dot{\phi} = p \tag{7.23}$$

$$\dot{p} = \delta_a + \Delta(x), \tag{7.24}$$

where $\Delta(x) = W_0^* + W_1^*\phi + W_2^* p + W_3^*|\phi|p + W_4^*|p|p + W_5^*\phi^3$. The parameters for wing rock motion are adapted from [87] and [94], they are $W_0^* = 0.0, W_1^* = 0.2314, W_2^* = 0.6918, W_3^* = -0.6245, W_4^* = 0.0095, W_5 = 0.0214$. Initial conditions for the simulation are arbitrarily chosen to be $\phi = 1deg, p = 1deg/s$. The task of the controller is to drive the state to the origin. To that effect, a stable second order reference model is used. In the following the proportional gain K_x and the feedforward gain K_r in equation 2.8 are held constant.

7.2.1 Case 1: Structured Uncertainty

Consider first the case where the structure of the uncertainty is known (Section 2.2.2, in Chapter 2). We use the Fourier Transform Regression [67] method for solving the least squares problem, the details of this method are given in appendix B. Figure 7.2 shows the performance of the baseline adaptive control law of equation 2.16 without the least squares modification. For the low gain case, a learning rate of $\Gamma_W = 3$ was used, while for the high gain case a learning rate of $\Gamma_W = 10$ was used; in both cases $\Gamma_\theta = 0.015$. It is seen that the performance of the controller in both cases is unsatisfactory. Figure 7.3 shows the phase portrait of the states when the adaptive law with least squares modification of Theorem 7.1 is used. It is seen that the system follows a smooth trajectory to the origin. Furthermore, it is interesting to note that

the performance of both the high gain and the low gain case is almost identical. Figure 7.4 shows the evolution of the adaptive control weights when only the baseline adaptive law of equation 2.16 is used. It is seen that the weights do not converge to the ideal values (W^*) and evolve in an oscillatory manner. In contrast, figure 7.5 shows the convergence of the weights when the least squares modification based adaptive law of Theorem 7.1 used. Figure 7.6 compares the reference model states with the plant states for the baseline adaptive law, while 7.7 compares the reference model and state output when the least squares modification based adaptive law is used. It can be seen that the performance of the adaptive law with least squares modification is superior to the baseline adaptive law. Finally, figure 7.8 shows that the tracking error converges exponentially to the origin when least squares modification term is used.

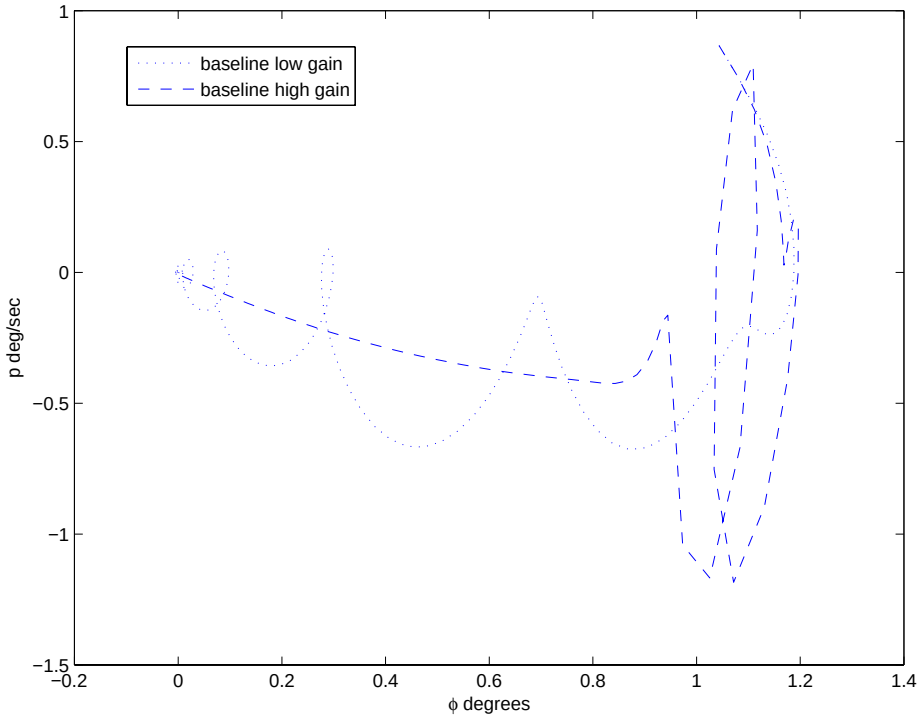


Figure 7.2: Phase portrait of system states with only baseline adaptive control

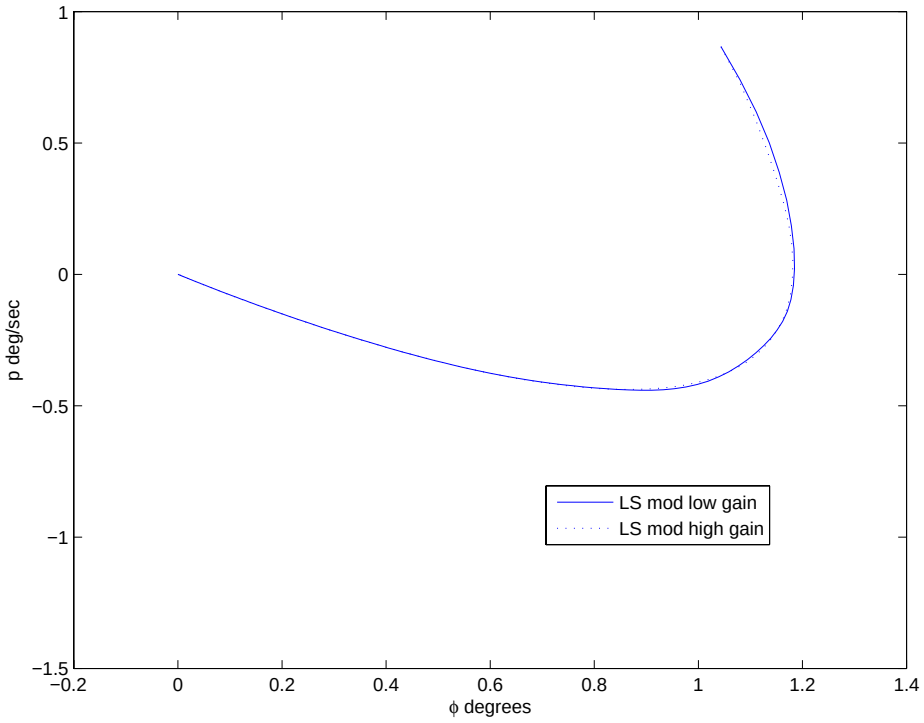


Figure 7.3: Phase portrait of system states with least squares modification

7.2.2 Case 2: Unstructured Uncertainty handled through RBF NN

For the results in this section we assume that it is only known that the structure of the uncertainty is unknown (Section 2.2.3, Chapter 2). Hence, RBF NN with 6 nodes and uniformly distributed centers over the expected range of the state space are used to capture the model uncertainty. Figure 7.9 shows the trajectory of the system in the phase space when the baseline adaptive control law of equation 2.16 is used. The performance can be contrasted with smooth convergence to the origin seen in figure 7.10 when adaptive law with least squares modification is used. Since the ideal weights W^* in this case are not known, we evaluate the performance of the adaptive law by comparing the output of the RBF NN with the actual model uncertainty with weights *frozen* after the simulation run is over. Figure 7.11 shows the comparison. It is clearly seen that the NN weights obtained with the least squares modification based

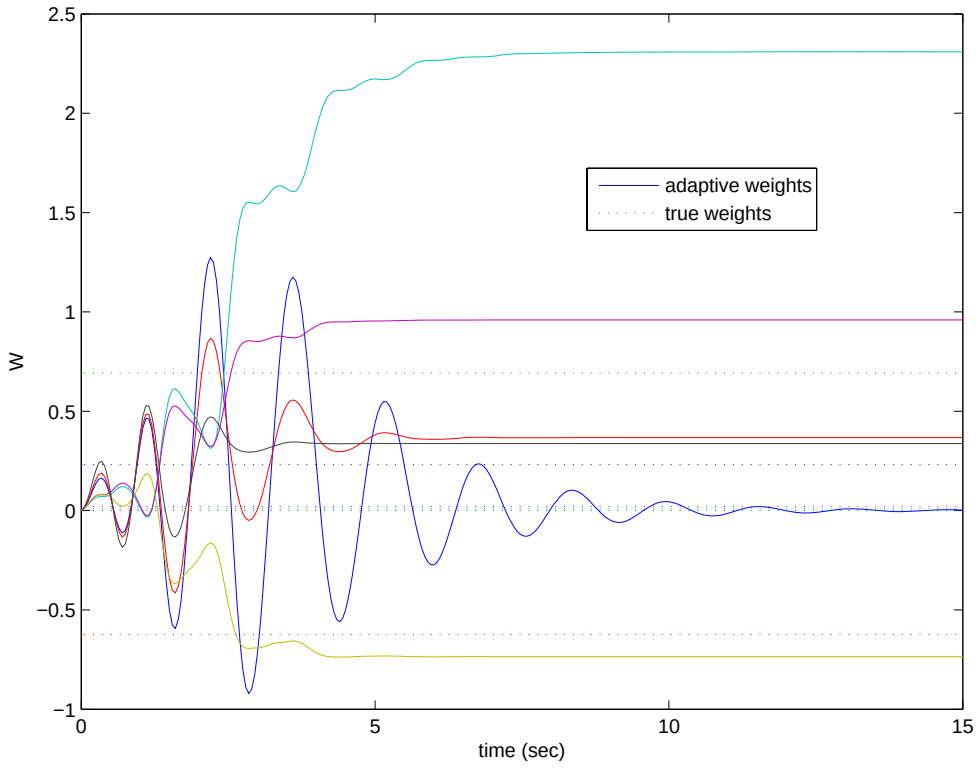


Figure 7.4: Evolution of adaptive weights with only baseline adaptive control

adaptive law are able to successfully and accurately capture the uncertainty, this is a clear indication that the weights have converged very close to their ideal values.

7.3 A Recursive approach to Least Squares Modification

The least squares modification presented in the previous sections requires the inversion of a matrix (7.11 or in 7.3). This inversion can prove cumbersome to perform online, especially if multiple input cases are considered. An alternative way to solve the least squares problem is to use a recursive approach. In this section we describe a recursive approach to least squares modification.

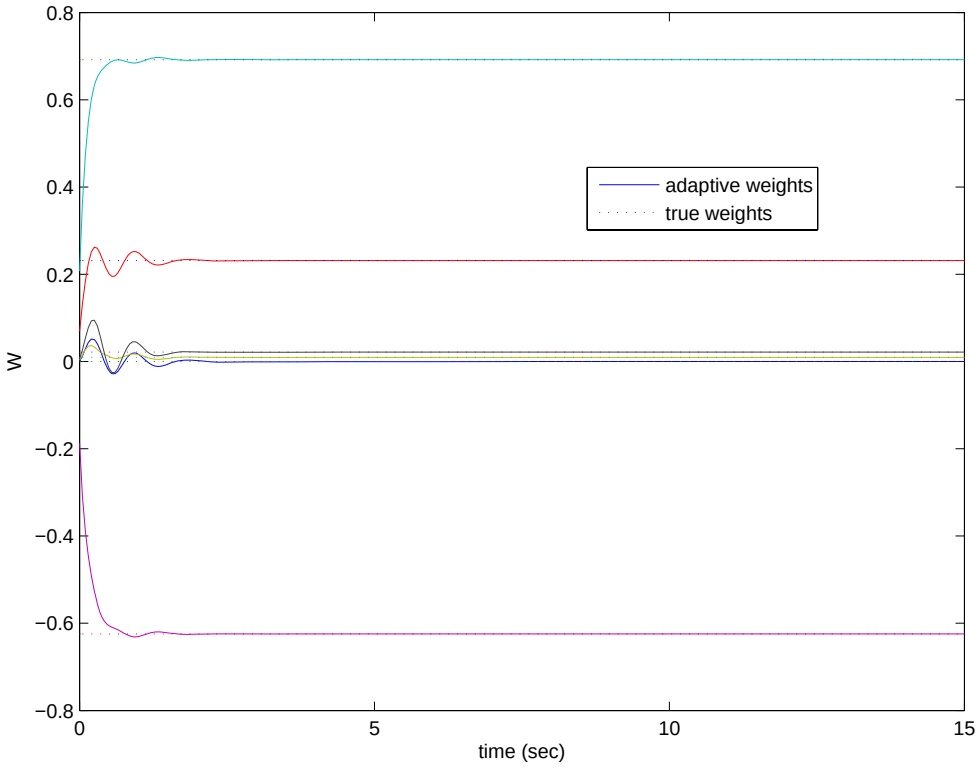


Figure 7.5: Evolution of adaptive weights with least squares modification

7.3.1 Recursive Least Squares Regression

A solution to the least squares problem can be found through Kalman filtering theory by casting the least squares problem as parameter estimation problem. Since the ideal weights are assumed to constant, the following model can be used for an estimate of the ideal weights θ,

$$\theta(k) = \theta(k-1), \tag{7.25}$$

$$\Delta(k) = \Phi^T(x(k))\theta(k). \tag{7.26}$$

Let $S(k)$ denote the Kalman filter error covariance matrix, $\hat{\theta}$ denote the estimate of the ideal weights θ, then setting the Kalman filter process noise covariance matrix $Q(k) = 0$, and the measurement covariance $R > 0$, the Kalman filter based least

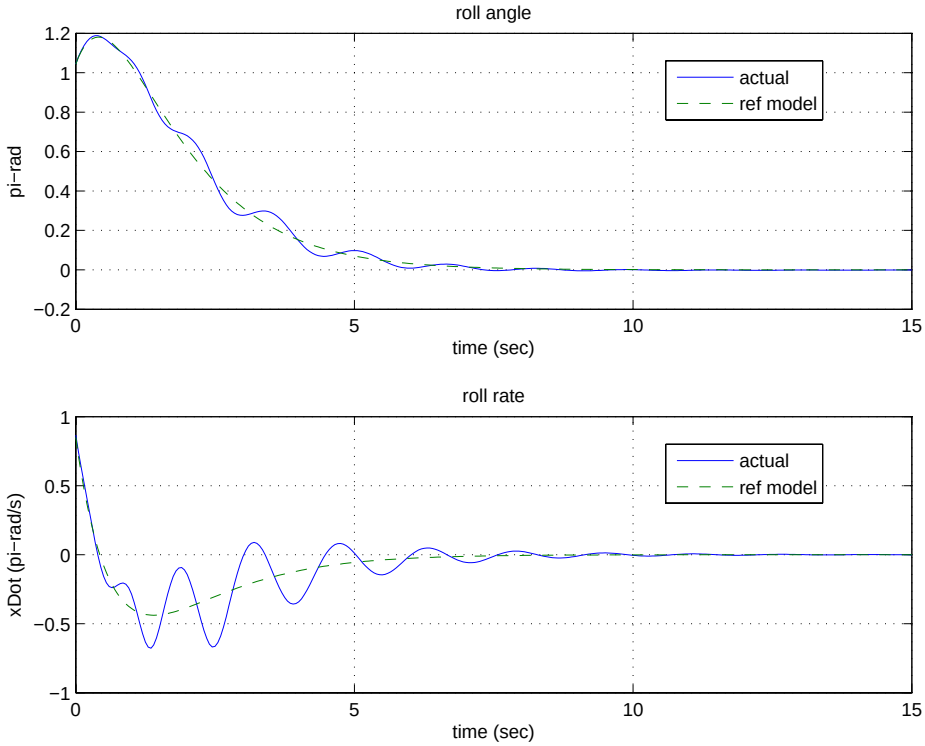


Figure 7.6: Performance of adaptive controller with only baseline adaptive law

squares estimate can be updated in the following manner

$$\hat{\theta}(k+1) = \hat{\theta}(k) + K(k+1)[\Delta(k+1) - \Phi^T(k+1)\hat{\theta}(k)], \tag{7.27}$$

$$K(k+1) = S(k)\Phi^T(k+1)[R + \Phi^T(k+1)S(k)\Phi(k+1)]^{-1}, \tag{7.28}$$

$$S(k+1) = [I - K(k+1)\Phi(k+1)]S(k). \tag{7.29}$$

7.3.2 Recursive Least Squares Based Modification

We now describe a method by which the least squares estimate of the ideal weights can be incorporated in the adaptive control law. Let $r^T = e^T PB$ where e, P, B are as in Chapter 2.2, Γ_W, Γ_θ are positive definite matrices denoting the learning rate. Let $\delta(t)$ denote the interval between two successive samples k and $k+1$, let T denote the time when sample k was obtained, for the current instant in time t, define the piece wise continuous sequence $\theta(t) = \hat{\theta}(k)$ for $T \leq t < T + \delta(t)$, where $\hat{\theta}(k)$ is as in 7.27.

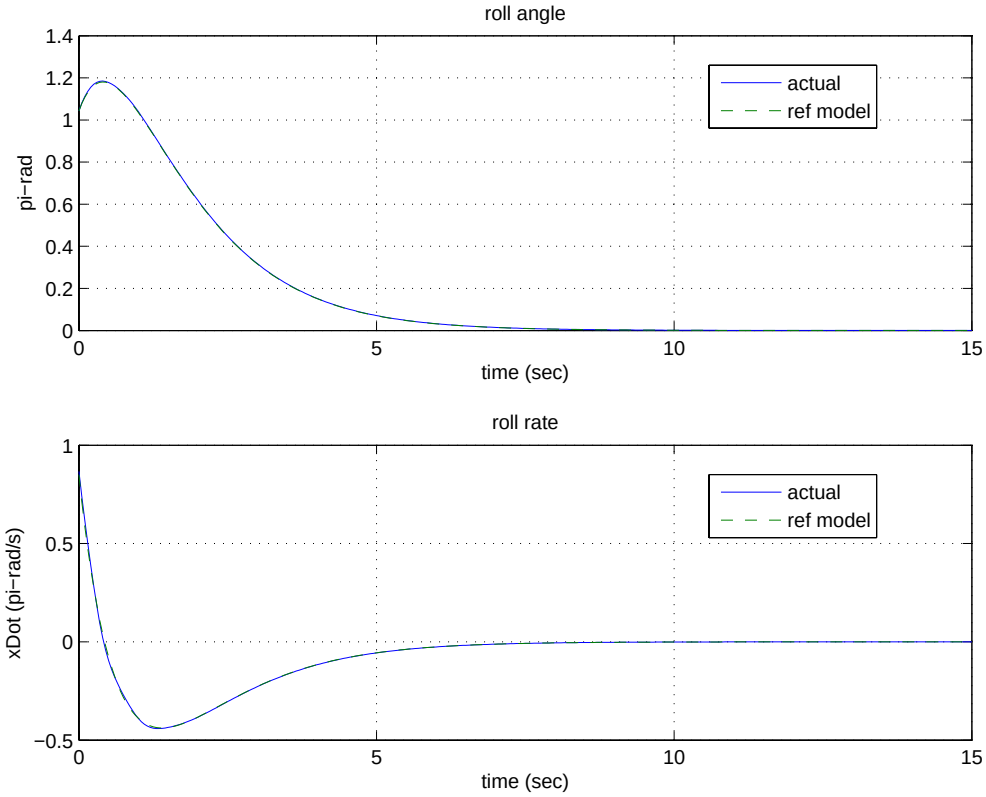


Figure 7.7: Performance of adaptive controller with least squares modification

The adaptive law for updating the weights W is chosen as

$$\dot{W}(t) = -(\Phi(x(t))r^T(t) - \Gamma_\theta(W(t) - \theta(t)))\Gamma_W. \tag{7.30}$$

In the above equation, the term $\Gamma_\theta(W(t) - \theta(t)))$ serves to combine the indirect recursive least based estimate of the ideal weights smoothly into the baseline direct adaptive training law of equation 2.16. This term acts as a modification term to the baseline adaptive law.

In the following, we present Lyapunov based stability analysis for the chosen adaptive law.

Theorem 7.2 Consider the system in equation 2.6, the reference model in equation 2.7, the control law given by equation 2.8, the case of structured uncertainty with the uncertainty given by $\Delta(x) = W^{*T}\Phi(x)$, the weight update law of equation

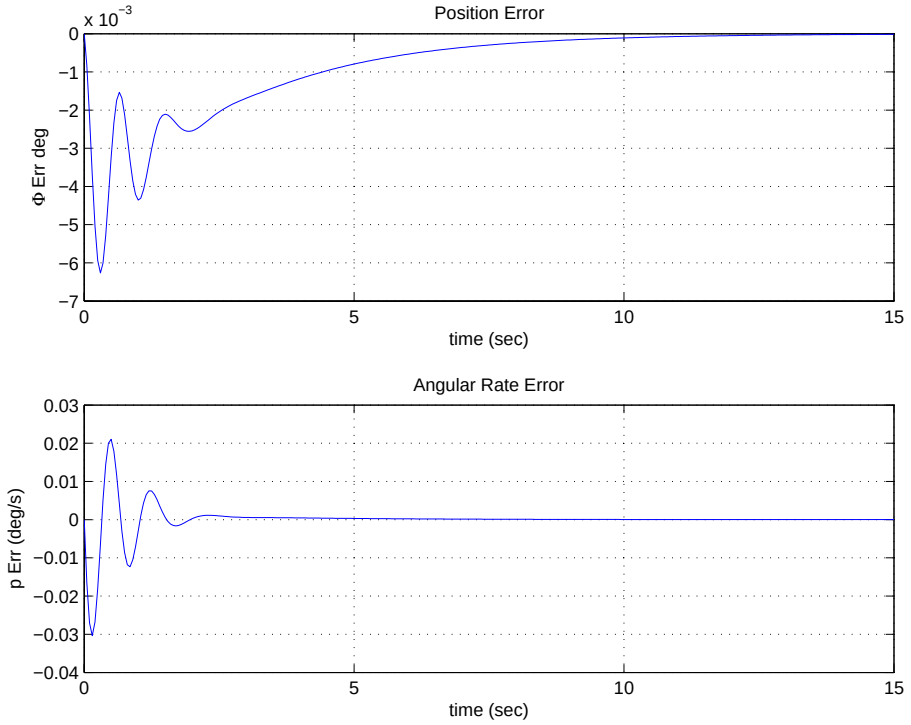


Figure 7.8: Evolution of tracking error with least squares modification

7.30, and assume that Condition 7.1 is satisfied, then the solution $(e(t), W(t))$ of the closed loop system given by equations 2.12 and 7.30 is uniformly ultimately bounded.

Proof Let $\tilde{W} = W - W^*$, let tr denote the trace operator, and consider the following positive definite and radially unbounded Lyapunov like candidate

$$V(e, \tilde{W}) = \frac{1}{2} e^T P e + \frac{1}{2} tr(\tilde{W}^T {\Gamma_W}^{-1} \tilde{W}). \tag{7.31}$$

Taking the time derivative of the Lyapunov candidate along the trajectories of equations 2.12 and 7.13, and using the Lyapunov equation 2.13 results in

$$\begin{aligned} \dot{V}(e, \tilde{W}) = -\frac{1}{2} e^T Q e + r^T (W^T \Phi(x) - W^{*T} \Phi(x)) \\ + tr(\dot{W} {\Gamma_W}^{-1} \tilde{W}^T). \end{aligned} \tag{7.32}$$

Let ϵ be such that $W = \theta + \epsilon$, adding and subtracting $(W^T - \theta)^T \Gamma_\theta (W^T - \theta)$ to

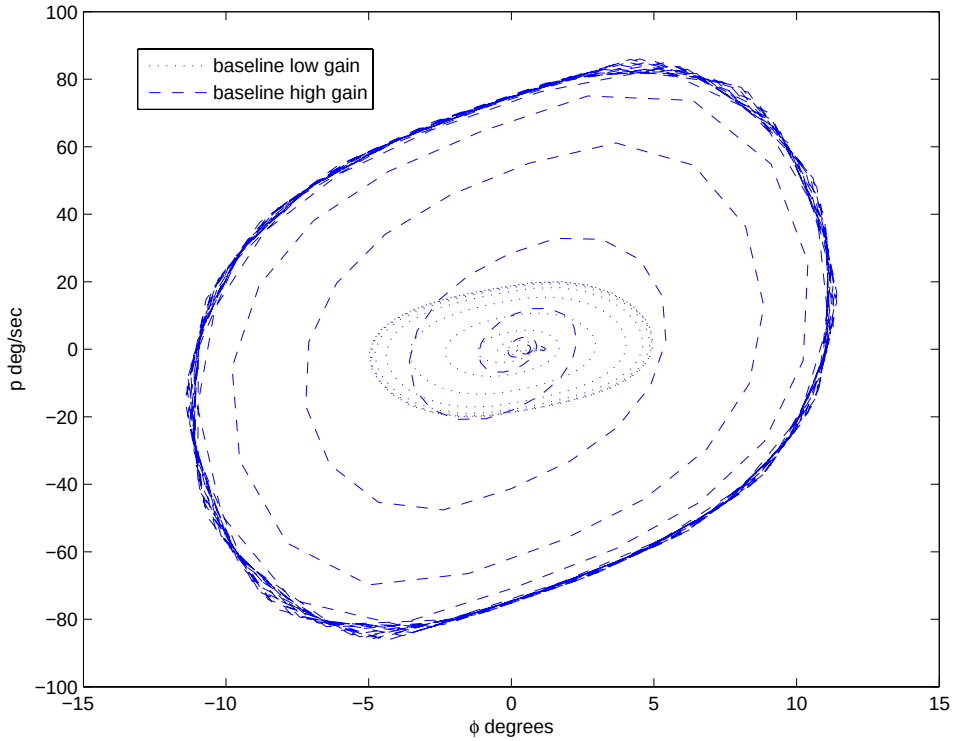


Figure 7.9: Phase portrait of system states with only baseline adaptive control while using RBF NN

equation 7.32 and using the definition of ϵ yields,

$$\begin{aligned}\dot{V}(e,\tilde{W}) = -\frac{1}{2}e^T Q e + r^T(\tilde{W}^T\Phi(x)) + tr(\dot{W}{\Gamma_W}^{-1}\tilde{W}^T) \\ + \tilde{W}^T\Gamma_\theta(W-\theta) - \tilde{W}^T\Gamma_\theta(W-\theta).\end{aligned} \tag{7.33}$$

Rearranging yields

$$\begin{aligned}\dot{V}(e,\tilde{W}) = -\frac{1}{2}e^T Q e \\ + tr((\dot{W}{\Gamma_W}^{-1} + \Phi(x)r^T + \Gamma_\theta(W-\theta))\tilde{W}^T) \\ - \tilde{W}^T\Gamma_\theta(W-\theta).\end{aligned} \tag{7.34}$$

Setting $tr((\dot{W}{\Gamma_W}^{-1}+\Phi(x)r^T+\Gamma_\theta(W-\theta))\tilde{W}^T) = 0$ yields the adaptive law of equation

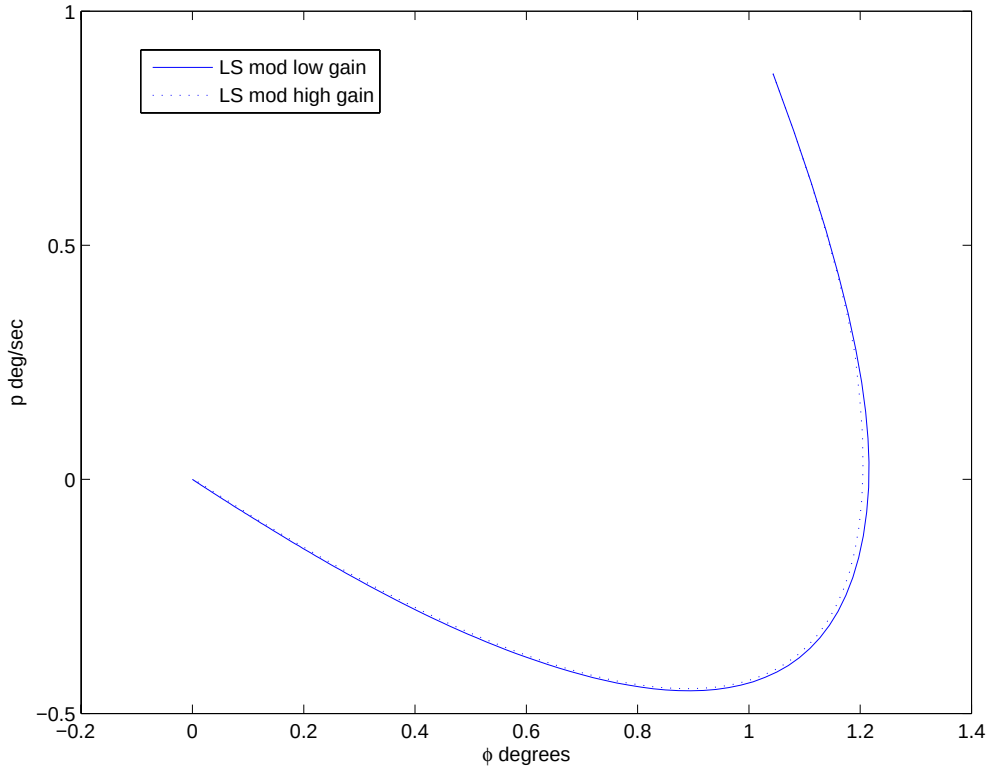


Figure 7.10: Phase portrait of system states with least squares modification while using RBF NN

7.30. Consider the last term in 7.34,

$$\begin{aligned}\tilde{W}^T\Gamma_\theta(W-\theta) = &\quad (W-W^*)^T\Gamma_\theta(W-\theta)\\ = &\quad (W-W^*)^T\Gamma_\theta(W-W^*)\\ &+(W-W^*)^T\Gamma_\theta\epsilon. \end{aligned} \tag{7.35}$$

Letting $\lambda_{\min}(Q)$ and $\lambda_{\min}(\Gamma_\theta)$ denote the minimum eigenvalues of Q and Γ_θ we have that equation 7.34 becomes

$$\dot{V}(e,\tilde{W}) = -\frac{1}{2}\|e\|^2\lambda_{\min}(Q) - \|\tilde{W}\|^2\lambda_{\min}(\Gamma_\theta) - \tilde{W}^T\Gamma_\theta\epsilon. \tag{7.36}$$

With appropriate choice of $S(0)$ and R, the Kalman filter estimation error $\theta(k)-\hat{\theta}(k)$ and $S(k)$ of equation 7.27, 7.29 remain bounded, hence ϵ remains bounded. Therefore, for a given choice of Q and Γ_θ, $\dot{V}(e,\tilde{W}) < 0$ outside of a compact set, which shows

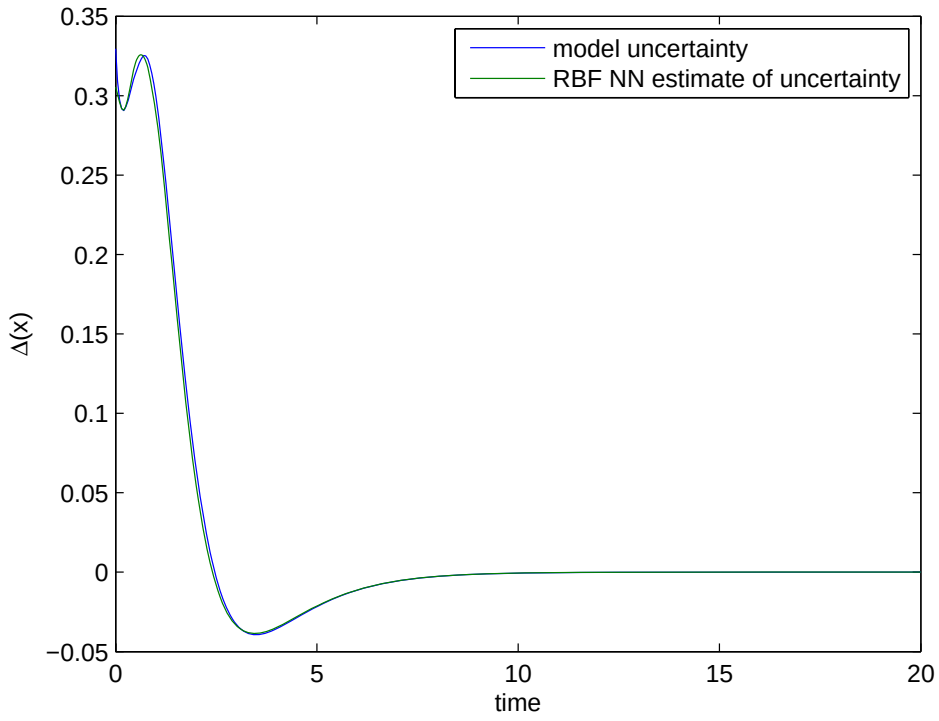


Figure 7.11: RBF NN model uncertainty approximation with weights frozen post adaptation

that the solution $(e(t), W(t))$ of the closed loop system given by equations 2.12 and 7.30 is uniformly ultimately bounded. ∎

Remark 7.7 The above proof shows uniform ultimate boundedness of the tracking error and adaptive weights. Furthermore, note that since A_{rm} is Hurwitz, x_{rm} is bounded for bounded $r(t)$, therefore it follows that x is bounded. It can be clearly seen that if $\epsilon \to 0$ then tracking error $e \to 0$. This condition will be achieved when $\theta \to W^*$, that is when the Kalman filter estimate of the ideal weights in 7.27 converges. The convergence of the Kalman filter estimate is related to choice of $S(0), R$ and the presence of excitation in the system stats [31].

Remark 7.8 The above proof can be easily extended to the case where the structure of the uncertainty is unknown (Section 2.2.3 from Chapter 2) by using Radial

Basis Function Neural Networks for approximating the uncertainty. The following adaptive law will result in uniform ultimate boundedness of all states

$$\dot{W} = -(\sigma(x)r^T - \Gamma_\theta(W - \theta))\Gamma_W. \tag{7.37}$$

Furthermore, referring to equation 2.19 and noting that in this case $\epsilon = \tilde{\epsilon}$, it can be shown that if the Kalman filter estimates of the ideal weights converge, then the weights will approach a neighborhood of the best linear approximation of the uncertainty.

Remark 7.9 The increased computational burden when using the adaptive law of equation 7.30 consists mainly of evaluating equations 7.27,7.28, and 7.29. It should be noted that since $\Phi(x) \in \Re^m$, the inversion in equation 7.28 is reduced to a division by a scalar.

7.4 Simulation results

In this section we use the method of Theorem 7.2 for the control a wing rock dynamics model. The dynamics of the model are described in equation 7.23. Initial conditions for the simulation are arbitrarily chosen to be $\phi = 1$ degree, $p = 1$ degree/second. The task of the controller is to drive the state to the origin. To that effect, a stable second order reference model is used with a natural frequency and a damping ratio of 1. The proportional gain K_x and the feedforward gain K_r in equation 2.8 are held constant for all of the presented simulation results.

The structure of the uncertainty and the ideal weights W^* are known for the wing rock dynamics model, hence the performance of the adaptive law can be accurately evaluated in terms of convergence of adaptive weights W to the ideal weights. The least squares problem is solved recursively using equations 7.27, 7.28, and 7.29. It is assumed that no *a priori* information is available about the ideal weights,hence we choose $\hat{\theta}(0) = 0$, consequently, the initial Kalman filter error covariance matrix $S(0)$

is chosen to have diagonal elements with large positive values. Figure 7.12 shows the performance of the baseline adaptive control law of equation 2.16 without the recursive least squares modification. The learning rate used was $\Gamma_W = 3$ for the low gain case, and $\Gamma_W = 10$ for the high gain case. It is seen that the performance of the controller in both cases is unsatisfactory. Figure 7.13 shows the phase portrait of the states when the adaptive law of equation 7.30 is used. It is seen that in both the low gain and the high gain case the system follows a smooth trajectory to the origin. Figure 7.14 shows the evolution of the adaptive control weights when only the baseline adaptive law of equation 2.16 is used. It is seen that the weights do not converge to the ideal values (W) and evolve in an oscillatory manner. In contrast, figure 7.15 shows the convergence of the weights when the adaptive law of equation 7.30 is used. Figure 7.16 compares the reference model states with the plant states for the baseline adaptive law, while 7.17 compares the reference model and state output when the adaptive law of equation 7.30 is used. It can be seen that the performance of the adaptive law of Theorem 7.2 is superior to that of the baseline adaptive law. Furthermore, we note that parameter convergence was observed despite using a non-persistently exciting reference input ($r(t) = 0 \forall t$).

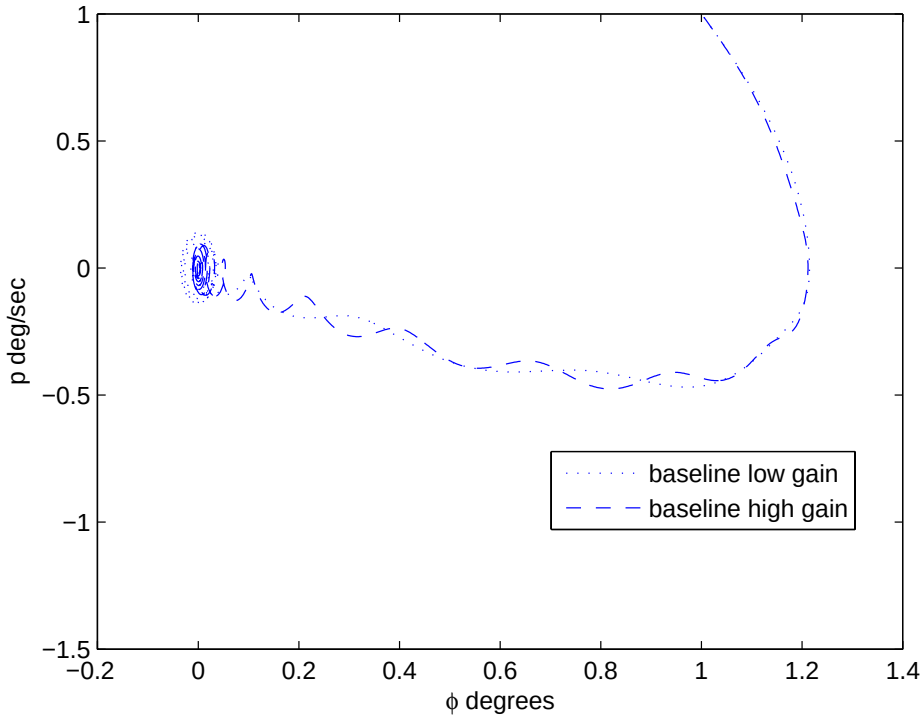


Figure 7.12: Phase portrait of system states with only baseline adaptive control

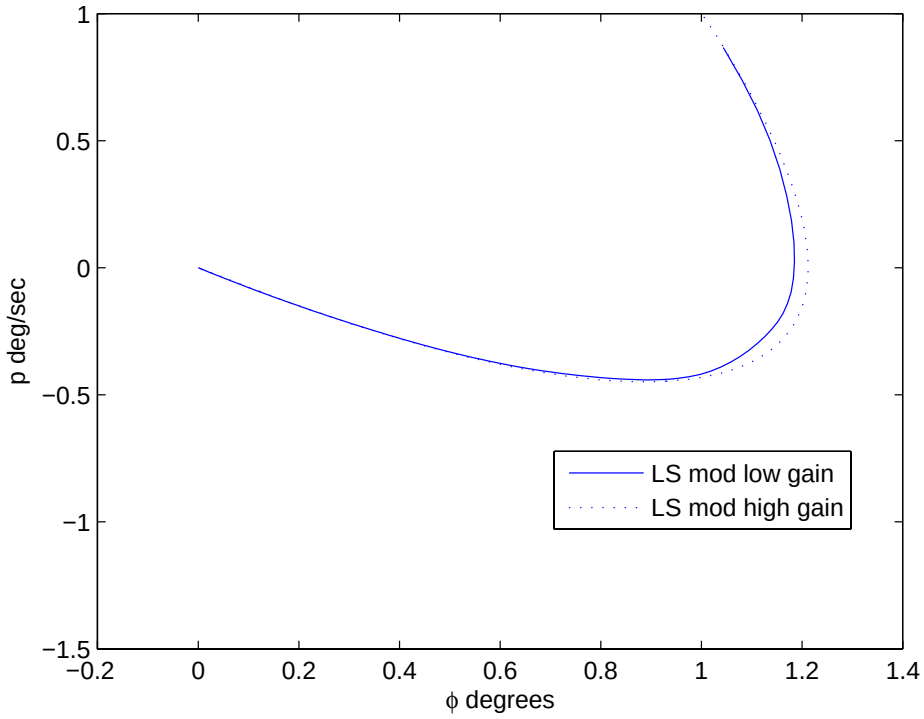


Figure 7.13: Phase portrait of system states with recursive least squares modification of equation 7.30

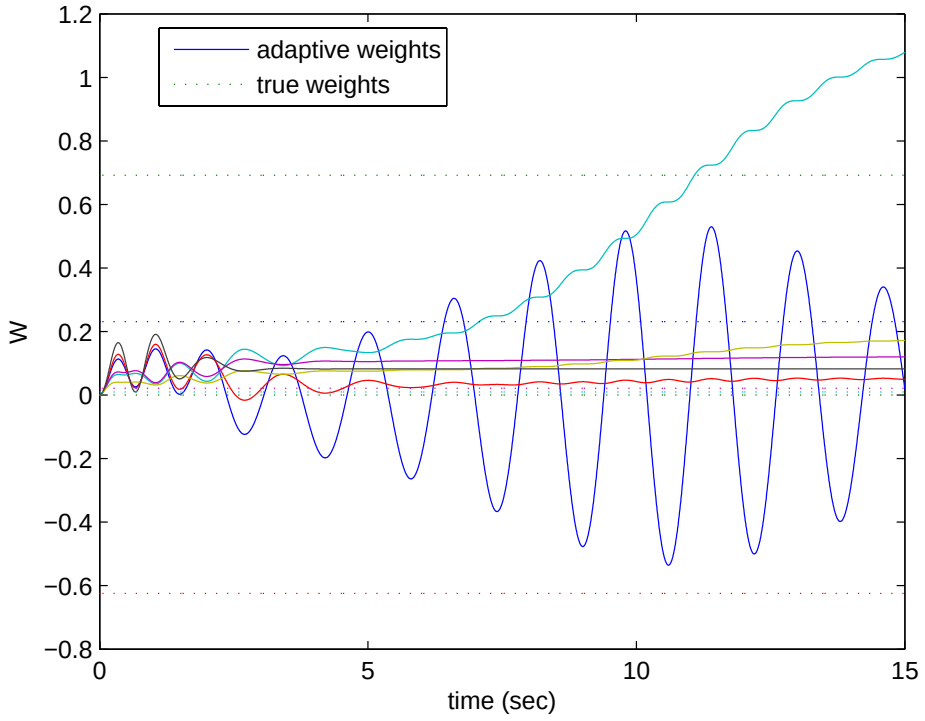


Figure 7.14: Evolution of adaptive weights with only baseline adaptive control

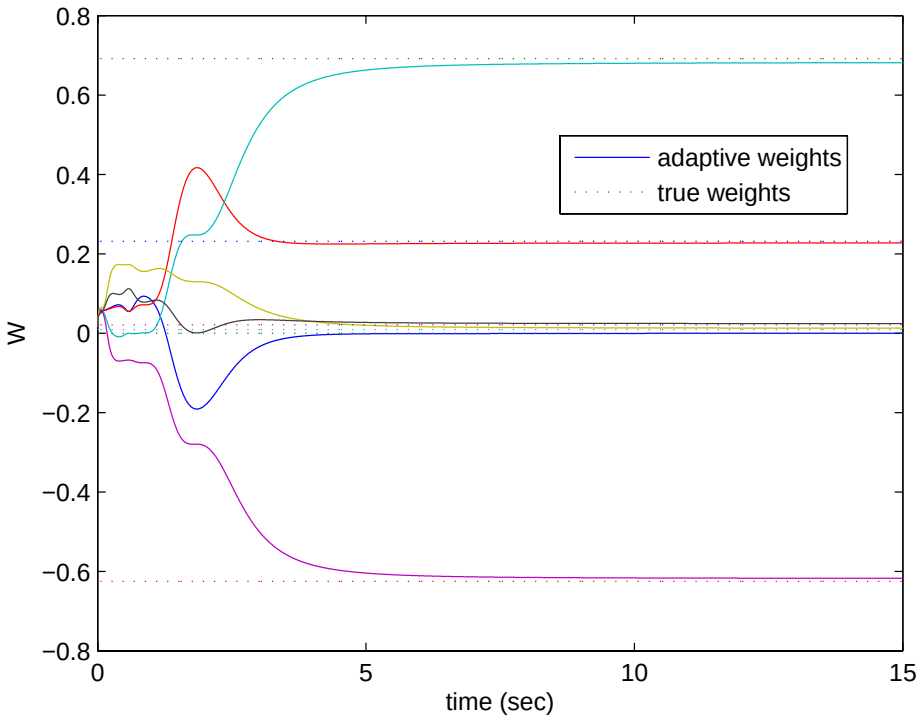


Figure 7.15: Evolution of adaptive weights with recursive least squares modification of equation 7.30

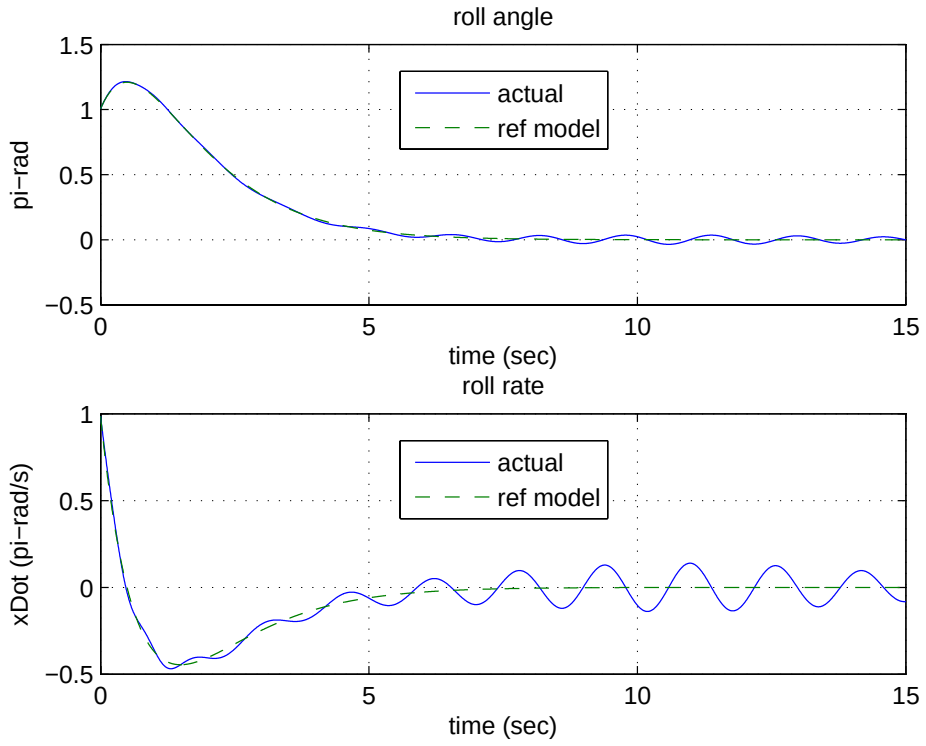


Figure 7.16: Performance of adaptive controller with only baseline adaptive law

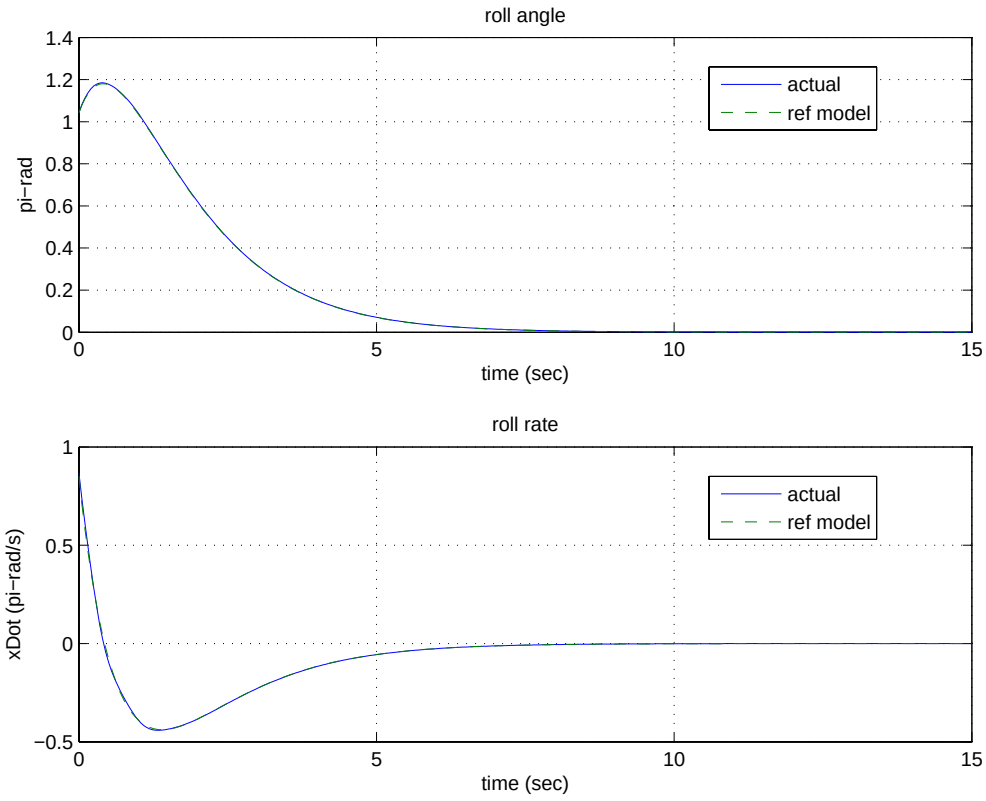


Figure 7.17: Tracking performance of the recursive least squares modification based adaptive law of equation 7.30

CHAPTER VIII

FLIGHT IMPLEMENTATION OF CONCURRENT LEARNING NEURO-ADAPTIVE CONTROL ON A ROTORCRAFT UAS

8.1 Motivation

Unmanned Aerial Systems (UAS) represent emerging technology that has already seen various successful applications around the globe. The interest in this technology is fueled by the ability of UAS to perform tasks autonomously that are dangerous to human operators, are of a repetitive nature, or demand high endurance and reliability beyond that of human capability. Currently, UAS are used mainly for surveillance and reconnaissances missions. UAS designed for these tasks are often remotely controlled, and are incapable of performing highly aggressive maneuvers. However, as the technology matures, UAS are expected to take on increasingly challenging roles in both the civil and military sectors. Some possible examples include Unmanned Combat Air Vehicles (UCAV), and highly agile Vertical Take Off and Landing (VTOL) air vehicles. Hence, developing flight control systems for UAS that perform as well as (or better) than human pilots has become an active technological challenge.

The capabilities of modern UAS are limited by their ability to track demanding trajectories which include high speed dashes, break turns, and other such aggressive maneuvers. Furthermore, UAS must also be capable of handling unmodeled disturbances, structural changes, partial system failures, and transitioning seamlessly through different flight domains. For example a rotorcraft VTOL UAS must demonstrate seamless transition through hover, forward flight, turning flight domains [7, 63, 30, 16, 80, 64]. Recent research has shown that adaptive control methodologies

are one approach that can address this challenge in a robust and efficient manner. For example Johnson, Kannan, and others have demonstrated that a VTOL UAS can be controlled effectively through its entire flight envelop using Neural Network (NN) based adaptive control laws similar to those in equation 2.28 [53],[50]. Furthermore, Johnson, Turbe, Kannan, Wu and others have also shown that adaptive controllers can be used to control a fixed-wing UAS to perform autonomous transitions to and from hover [51]. However, it was noted that the traditional instantaneous error minimizing adaptive control laws (e.g. equation 2.28) suffered from short-term learning. That is, the adaptive controller did not exhibit improvement in performance even when the aircraft performed the same maneuvers repeatedly. On analyzing flight test data, it was noted that if the adaptive element weights were to approach their ideal values, long term improvement in performance could be realized. In this thesis we developed a method that uses both current and recorded data concurrently to improve the convergence properties of NN based adaptive controllers. In this chapter, we will apply the results for the control of a rotorcraft UAS.

8.2 Flight Test Vehicle

The concurrent learning adaptive controllers have been implemented on the Georgia Tech GTMax UAS (figure 8.2). The GTMax is based on the Yamaha RMAX platform and weighs around 66 Kg with a 3 meter rotor diameter. The vehicle has been equipped with two high speed flight computers, multiple redundant data links, an in-house developed Ground Control Station communication software, and has flown over 450 flights since March 2002. The baseline controller on the GTMax is a SHL NN based AMI-MRAC and uses the update laws of equation 5.17 and has been extensively proven in flight. Further details on the baseline controller can be found in [50] and in [53]. The concurrent learning adaptive law used is from Theorem 5.6, which guarantees that the solution $(e(t), W(t), V(t))$ will stay uniformly ultimately

bounded.

Figure 8.1: The Georgia Tech GTMax UAV in Flight

We begin with presenting results on a High Fidelity flight simulation of the GTMax Simulation. These results are important due to their reproducibility, controlled environment, and repeatability of commands. We then proceed to present flight test results on the GTMax.

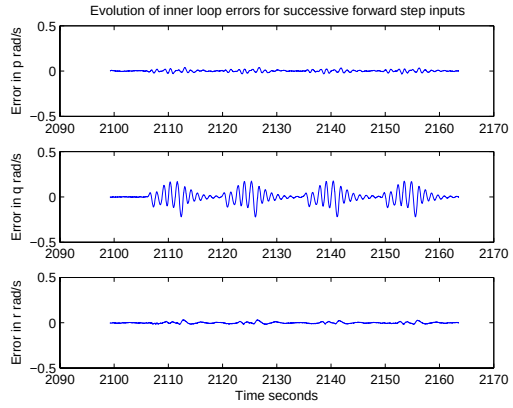
8.3 Implementation of concurrent Learning NN controllers on a High Fidelity Simulation

The Georgia Tech UAV lab maintains a high fidelity Software In the Loop (SITL) flight simulator for the GTMax UAS. The simulation is complete with sensor emulation, detailed actuator models, external disturbance simulation, and a high fidelity dynamical model.

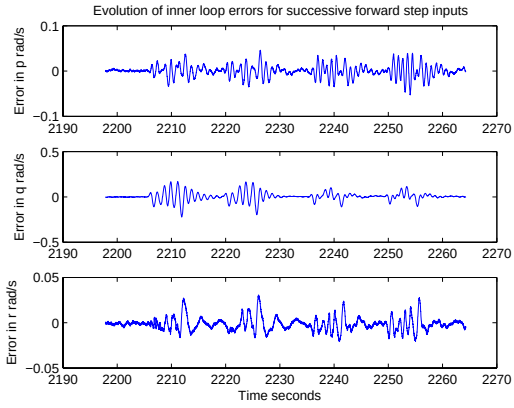
We command four successive forward step inputs with an arbitrary period of no command activity between any two successive steps. This type of input is used to mimic control tasks which involve commands that are repeated after an arbitrary time interval. Through these maneuvers, the UAS is expected to transition through forward flight and hover domain repeatedly. The performance of the inner loop controller is characterized by the errors in the three body angular rates (namely roll rate p, pitch rate q and yaw rate r), with the dominating variable being pitch rate q as

the rotorcraft accelerates and decelerates in forward step inputs. Figure 2(a) shows the performance of the inner loop controller with only instantaneous adaptation in the NN. It is clearly seen that there is no considerable improvement in the pitch rate error as the controller follows successive step inputs. The forgetting nature of the controller is further characterized by the evolution of NN weights in W and V matrices. Figure 2(c) and Figure 2(c) clearly show that the NN weights do not converge to a constant value, in fact as the rotorcraft performs the successive step maneuvers the NN weights oscillate accordingly, clearly characterizing the instantaneous (forgetting) nature of the adaptation.

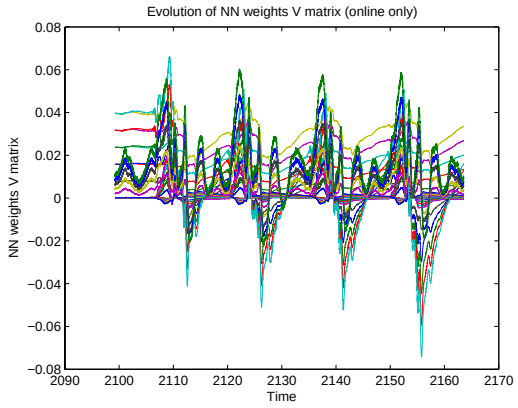
On the other hand, when both instantaneous and concurrent learning NN learning law of Theorem 5.6 is used a clear improvement in performance is seen characterized by the reduction in pitch rate error after the first two step inputs. Figure 2(b) shows the tracking performance of the concurrent learning augmented controller. The long term adaptation nature of the concurrent learning augmented adaptive controller is further characterized by the tendency the of NN weights to converge. Figure 2(d) and Figure 2(f) show that when concurrent learning is used along with instantaneous learning the NN weights do not exhibit periodic behavior and tend to converge to constant values. This indicates that the NN learns faster and retains the learning even when there is a lack of persistent excitation. This indicates that the combined instantaneous learning and concurrent learning controller will be able to perform better when performing a maneuver that it has previously performed, a clear indication of long term memory and semi-global learning.



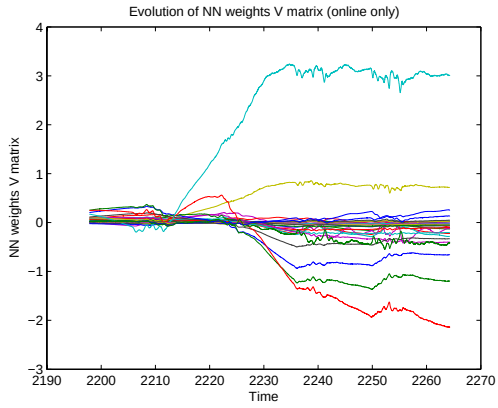
(a) Evolution of inner loop errors with Only Online Adaptation



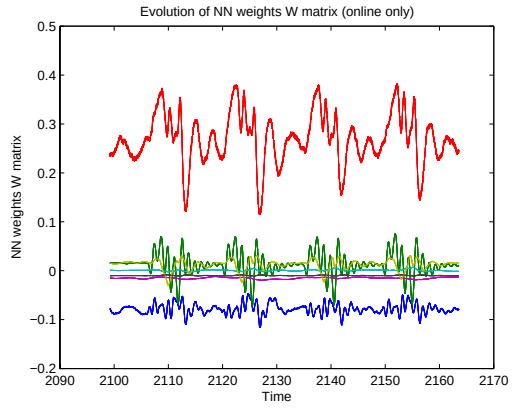
(b) Evolution of inner loop errors with concurrent Adaptation



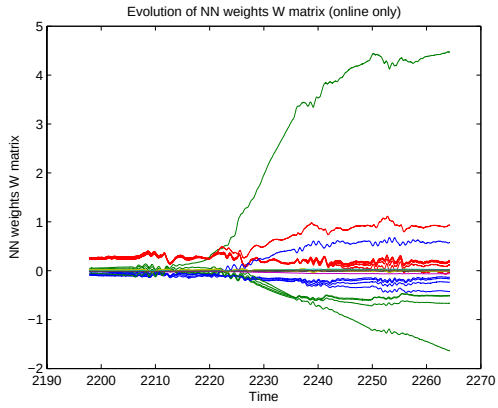
(c) Evolution of V matrix weights with Only Online Adaptation



(d) Evolution of V matrix weights with concurrent Adaptation



(e) Evolution of W matrix weights with Only Online Adaptation



(f) Evolution of W matrix weights with concurrent Adaptation

Figure 8.2: GTMax Simulation Results for Successive Forward Step Inputs with and without concurrent learning

8.4 Implementation of Concurrent Learning Adaptive Controller on a VTOL UAV

In this section we present some flight test results that characterize the benefits of using combined online and concurrent learning adaptive control. The flight tests presented here were executed on the Georgia Tech GTMax rotorcraft UAV (8.2). We begin by presenting flight test results for a series of forward steps. This series of maneuvers serves to demonstrate explicitly the effect of concurrent learning by showing improved weight convergence and reduction in the tracking error. We then present results from more complicated and aggressive maneuvers where it is highly desirable to have long term learning in order to improve performance. For this purpose we choose an aggressive trajectory tracking maneuver, in which the rotorcraft UAV tracks an elliptical trajectory with aggressive velocity and acceleration profile. The final maneuver chosen is an aggressive reversal of direction maneuver which first exchanges the kinetic energy of the rotorcraft for potential energy by climbing up. From the apex of its trajectory the rotorcraft falls back and reverses its direction of flight by continually aligning the heading with the local velocity vector.

8.4.1 Repeated Forward Step Maneuvers

The repeated forward step maneuvers are chosen in order to create a relatively simple situation in which the controller performs a repeated task. By using combined current and concurrent learning NN we expect to see improved performance through repeated maneuvers and a faster convergence of weights. Figure 8.4.1 shows the body frame states from recorded flight data for a chain of forward step inputs. Figure 4(a) and figure 4(b) shows the evolution of inner and outer loop errors. These results assert the stability (in the ultimate boundedness sense) of the combined concurrent and online learning approach.

Figure 5(d) and Figure 5(b) show the evolution of NN W and V weights as the

rotorcraft performs repeated step maneuvers and the NN is trained using combined online and concurrent learning method of Theorem 5.6. The NN V weights (5(b)) appear to go to constant values when concurrent learning adaptation is used, this can be contrasted with Figure 5(a) which shows the V weight adaptation for a similar maneuver without concurrent learning. NN W weights for both cases remain bounded, however it is seen that with concurrent learning adaptation the NN W weights seem to separate, this indicates alleviation of the rank-1 condition experienced by the baseline adaptive law relying only on instantaneous data [22]. The flight test results indicate a noticeable improvement in the error profile. In Figure 8.4.1 we see that the UAV tends not to have a smaller component of body lateral velocity (v) through each successive step. This is also seen in Figure 4(b) where we note that the error in v (body y axis velocity) reduces through successive steps. These effects in combination indicate that the combined online and concurrent learning system is able to improve performance over the baseline controller through repeated maneuvers, indicating long term learning. These results are of particular interest, since the maneuvers performed were conservative, and the baseline adaptive MRAC controller had already been extensively tuned.

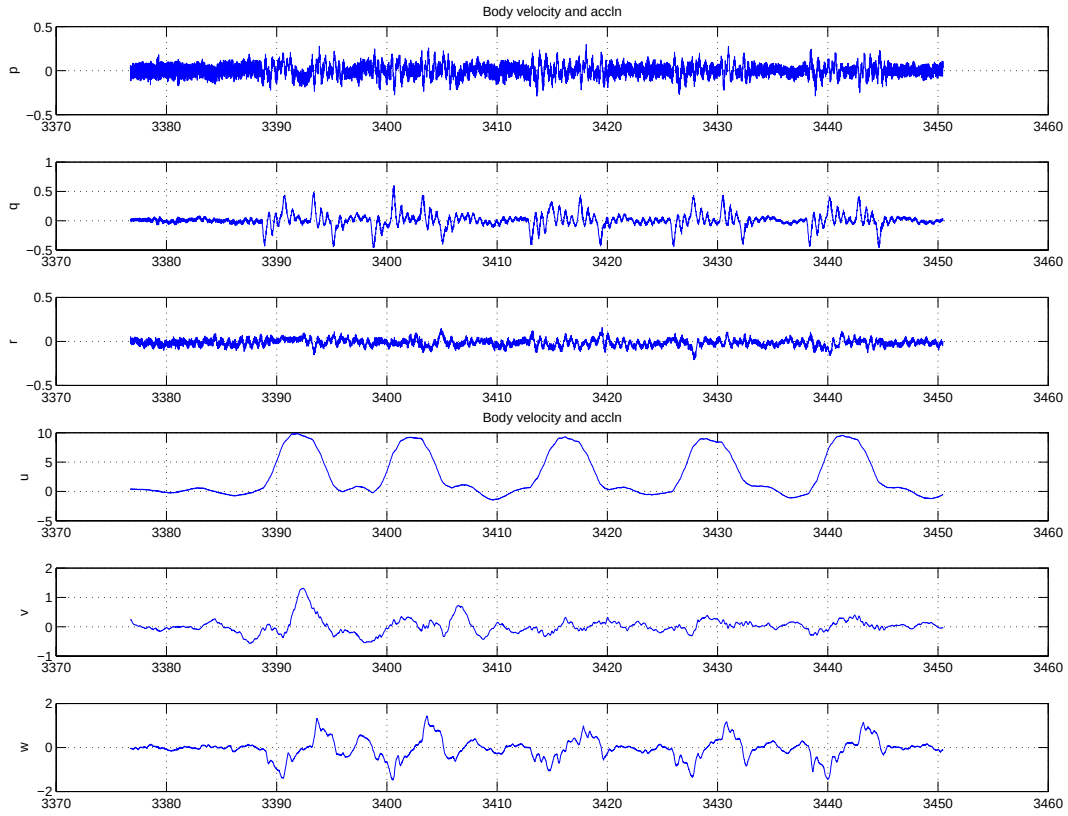
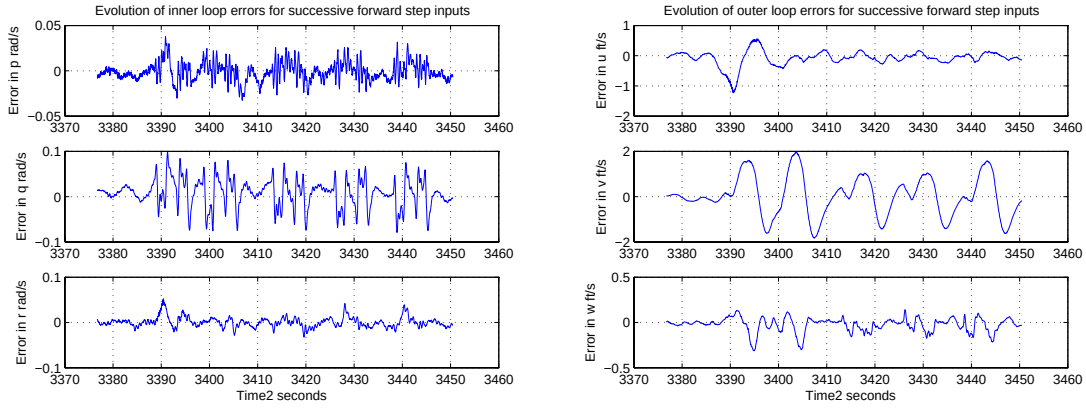


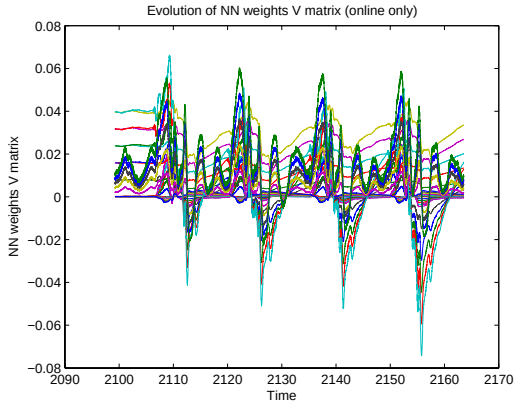
Figure 8.3: Recorded Body Frame States for Repeated Forward Steps



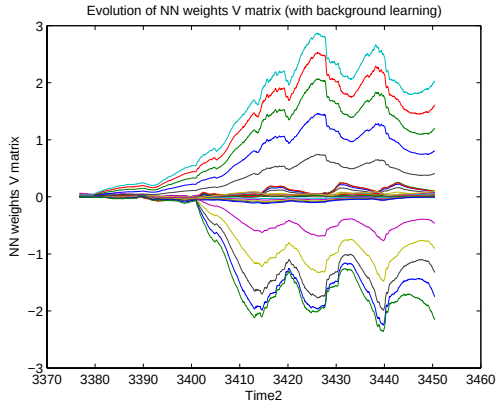
(a) Evolution of inner loop errors with concurrent Adaptation

(b) Evolution of outer loop errors with concurrent Adaptation

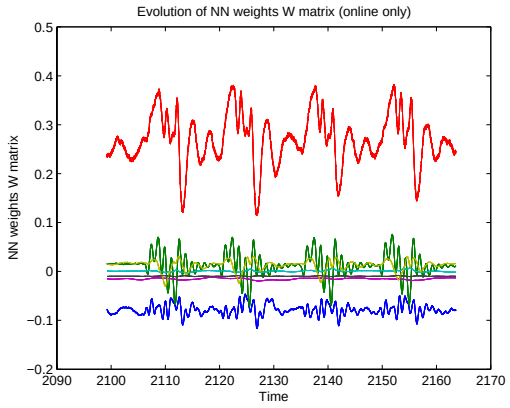
Figure 8.4: GTMax Recorded Tracking Errors for Successive Forward Step Inputs with concurrent Learning



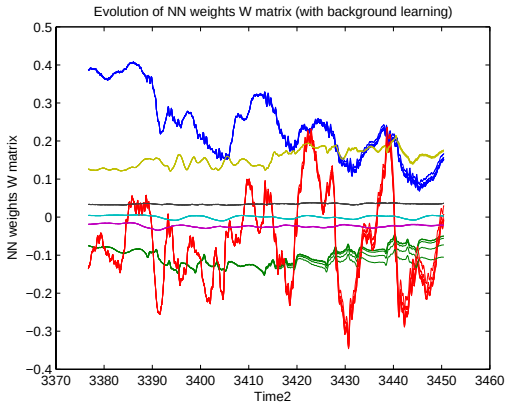
(a) Evolution of V matrix weights with Only Online Adaptation



(b) Evolution of V matrix weights with concurrent Adaptation



(c) Evolution of W matrix weights with Only Online Adaptation



(d) Evolution of W matrix weights with concurrent Adaptation

Figure 8.5: Comparison of Weight Convergence on GTMax with and without concurrent Learning

8.4.2 Aggressive Trajectory Tracking Maneuvers

Forward step maneuvers serve as a great test pattern due to their decoupled nature; however in the real world the UAV is expected to perform more complex maneuvers. In order to demonstrate the benefits of using the combined current and concurrent learning NN we present flight test results for trajectory tracking maneuver in which the UAV repeatedly tracks an elliptical trajectory with aggressive velocity ($50 ft/s$)

and acceleration ($20ft/s^2$) profile. Since these maneuvers involve state commands in more than one system state it is harder to visually inspect the data and see whether an improvement in performance is seen. In this thesis we address this issue by using the Euclidian norm of the error signal at each time step as a rudimentary metric. Further research needs to be undertaken in determining a suitable metric for this task. Figure 8.4.2.1 shows the recorded inner and outer loop states as the rotorcraft repeatedly tracks an oval trajectory pattern. In this flight, the first two ovals (until t = 5415 s) are tracked with a commanded acceleration of $30ft/sec^2$, while the rest of the ovals are tracked at $20ft/sec^2$. In the following we treat both these parts of the flight test separately.

8.4.2.1 Aggressive Trajectory Tracking with Saturation in the Collective Channel

Due to the aggressive acceleration profile of $30ft/s^2$ the rotorcraft collective channels were observed to saturate while performing high velocity turns. This leads to an interesting challenge for the adaptive controller. Figure 8.7 shows the evolution of the innerloop and outerloop tracking error. It can be clearly seen that the tracking error in the u (body x axis velocity) channel reduces in the second pass through the ellipse indicating long term learning by the combined online and concurrent learning adaptive control system. This result is further characterized by the noticeable reduction in the norm of the tracking error at every time step as shown in Figure 24.

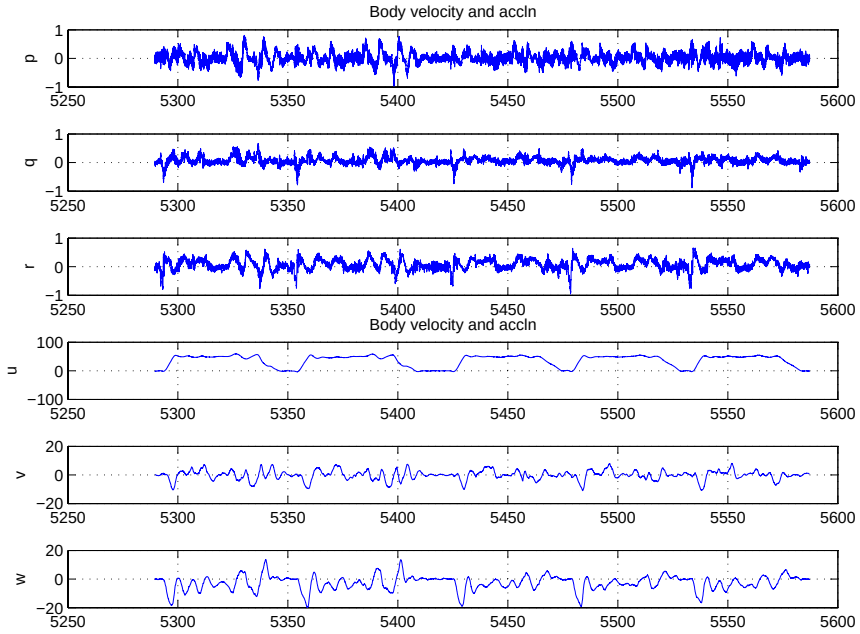
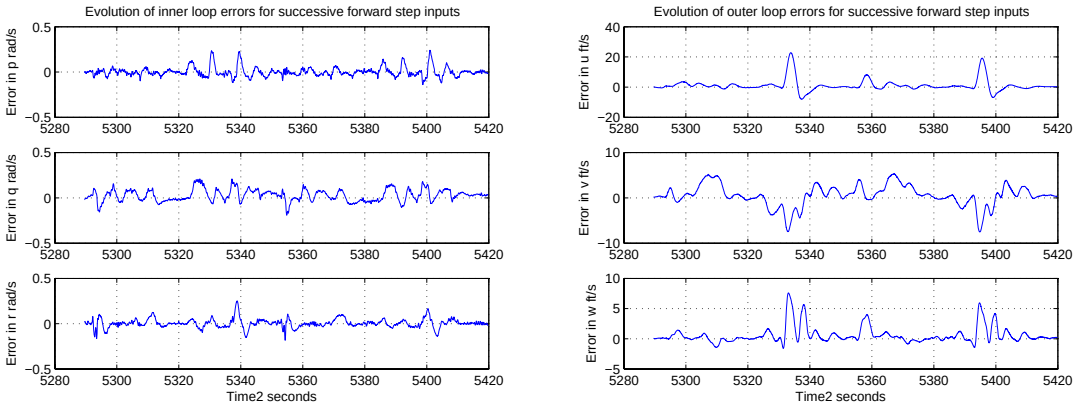


Figure 8.6: Recorded Body Frame States for Repeated Oval Maneuvers



(a) Evolution of inner loop errors with concurrent Adaptation

(b) Evolution of outer loop errors with concurrent Adaptation

Figure 8.7: GTMax Recorded Tracking Errors for Aggressive Maneuvers with Saturation in Collective Channels with concurrent Learning

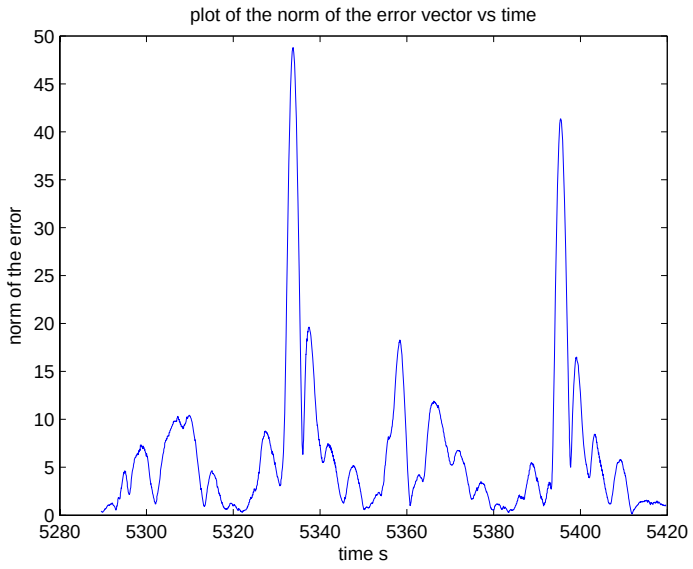
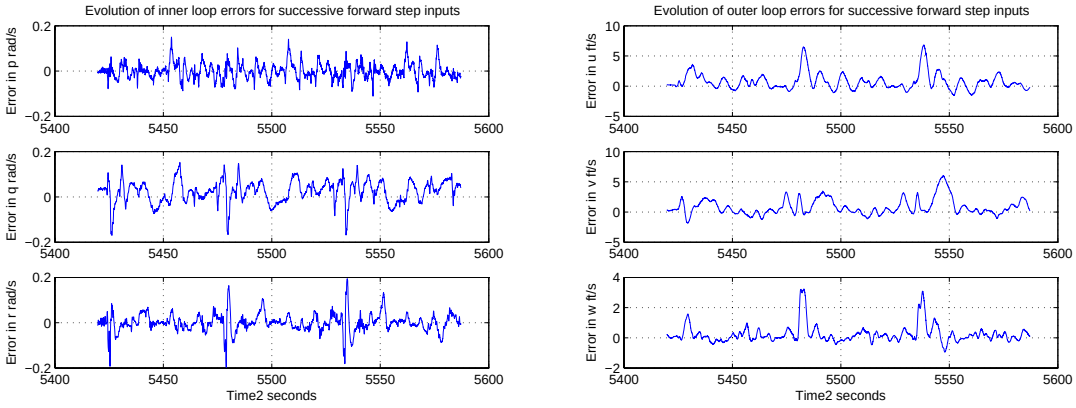


Figure 8.8: Plot of the norm of the error at each time step for aggressive trajectory tracking with collective saturation

8.4.2.2 Aggressive Trajectory Tracking Maneuver

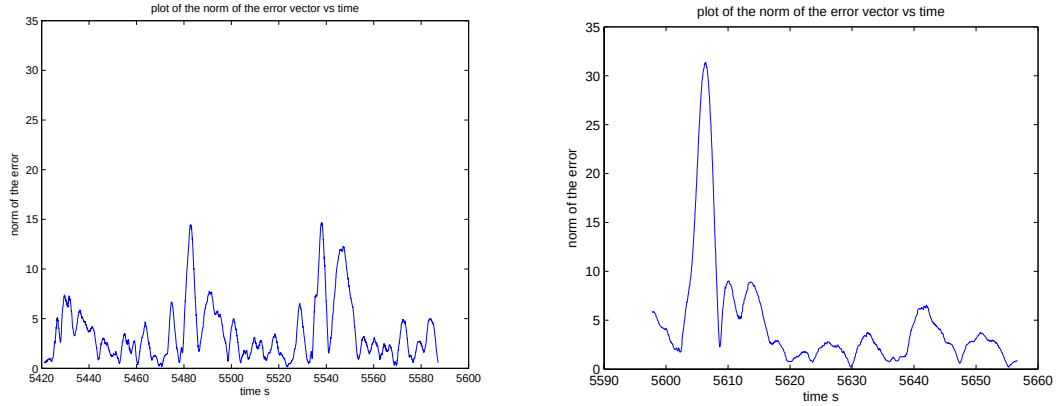
In this part of the maneuver the acceleration profile was reduced to $20 ft/sec^2$. At this acceleration profile, no saturation in the collective input was noted. Figure 8.9 shows the evolution of tracking error, and Figure 10(a) shows the plot of the norm of the tracking error at each time step.



(a) Evolution of inner loop errors with concurrent Adaptation

(b) Evolution of outer loop errors with concurrent Adaptation

Figure 8.9: GTMax Recorded Tracking Errors for Aggressive Maneuvers with concurrent Learning



(a) Evolution of the norm of the tracking error with concurrent Adaptation

(b) Evolution of the norm of the tracking error with only online Adaptation

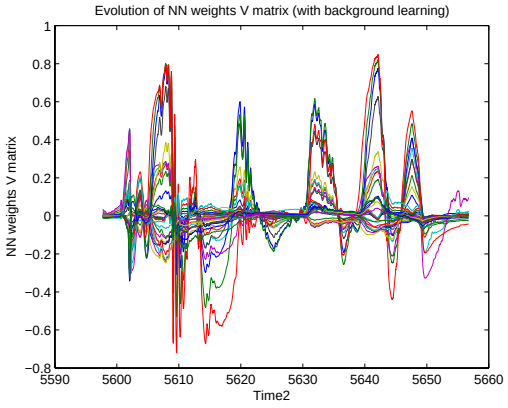
Figure 8.10: Comparison of norm of GTMax Recorded Tracking Errors for Aggressive Maneuvers

8.4.2.3 Aggressive Trajectory Tracking Maneuvers with Only Online Learning NN

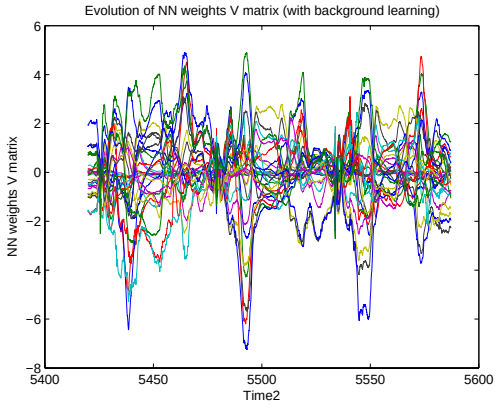
In order to illustrate the benefit of the combined online and concurrent learning adaptive controller we present flight test results as the rotorcraft tracks the same

trajectory command as in Section 8.4.2.1 , but with only online learning NN.

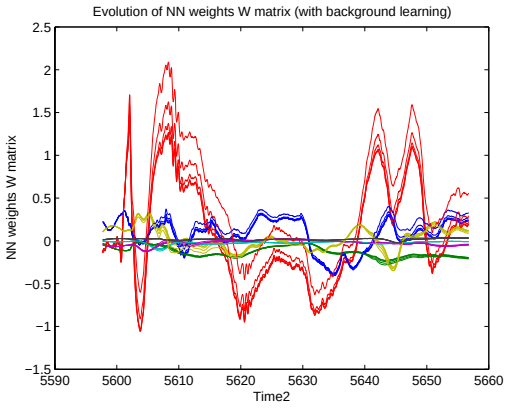
It is instructive to compare Figure 11(b), and Figure 11(d) which show the evolution of the NN weights with only online learning with Figure 11(a), and Figure 11(c) which show evolution of the NN weights with combined online and concurrent learning. Although absolute convergence of weights is not seen, as expected due to Theorem 5.6 it is interesting to see that when combined online and concurrent learning is on, the weights tend to be less oscillatory than when only online learning is on. Also, with combined online and concurrent learning, the weights do not tend to go to zero as the rotorcraft hovers between two successive tracking maneuver. Figure 10(b) shows the plot of the tracking error norm as a function of time without concurrent learning. Comparing this figure with Figure 10(a) it can be clearly seen that the norm of the error vector is much higher when only online learning is used. This indicates that the combined online and concurrent learning adaptive controller has improved trajectory tracking performance.



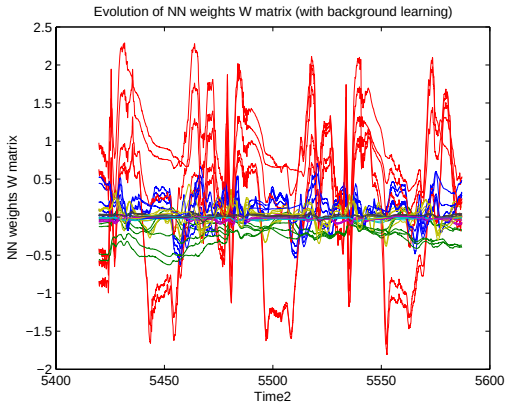
(a) Evolution of V matrix weights with Only Online Adaptation



(b) Evolution of V matrix weights with concurrent Adaptation



(c) Evolution of W matrix weights with Only Online Adaptation



(d) Evolution of W matrix weights with concurrent Adaptation

Figure 8.11: Comparison of Weight Convergence as GTMax tracks aggressive trajectory with and without concurrent Learning

In summary, the flight test results were in agreement with Theorem 5.6, which guarantees that the closed loop solution $(e(t), W(t), V(t)$ will remain uniformly ultimately bounded. Ongoing flight testing work on the GTMax includes developing techniques for improved implementation of concurrent learning adaptive controllers.

CHAPTER IX

FLIGHT IMPLEMENTATION OF CONCURRENT LEARNING NEURO-ADAPTIVE CONTROLLER ON A FIXED WING UAS

In this chapter, we present results from flight implementation of a concurrent learning Neuro-Adaptive controller onboard the Georgia Tech Twinstar UAS. The implementation uses a Radial Basis Function Neural Networks as the adaptive element and uses the adaptive control law developed in Theorem 5.3.

9.1 Flight Test Vehicle: The GT Twinstar

The GT Twinstar (Figure 9.1) is a foam built, twin engine aircraft that has been equipped with the Adaptive Flight Inc. (AFI, www.adaptiveflight.com) FCS 20®. The FCS 20 embedded autopilot system comes with an integrated navigation solution that fuses information using an extended Kalman filter from six degree of freedom inertial measurement sensors, Global Positioning System, air data sensor, and magnetometer to provide accurate state information [21]. The available state information includes velocity and position in global and body reference frames, accelerations along the body x, y, z axes, roll, pitch, yaw rates and attitude, barometric altitude, and air speed information. These measurements can be further used to determine the aircraft's velocity with respect to the air mass, and the flight path angle. The Twinstar can communicate with a Ground Control Station (GCS) using a 900 MHz wireless data link. The GCS serves to display onboard information as well as send commands to the FCS20. Flight measurements of airspeed and throttle setting are used to estimate thrust with this model. An elaborate simulation environment has also been

designed for the GT Twinstar. This environment is based on the Georgia Tech UAS Simulation Tool (GUST) environment [52]. A linear model for the Twinstar in nominal configuration (without damage) has been identified using the FTR method [23]. A linear model with 25% left wing missing has also been identified [17].

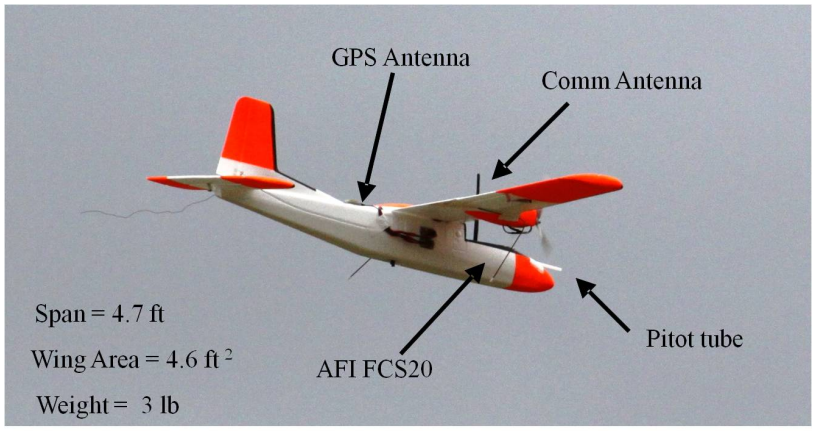


Figure 9.1: The Georgia Tech Twinstar UAS. The GT Twinstar is a fixed wing foam-built UAS designed for fault tolerant control work.

9.2 Flight Test Results

The guidance algorithm for GT Twinstar is designed to ensure that the aircraft can track feasible trajectories even when it has undergone severe structural damage [49]. The control algorithm has a cascaded inner and outer loop design. The outerloop, which is integrated with the guidance loop, commands the desired roll angle (ϕ), angle of attack (α), and sideslip angle (β) to achieve desired waypoints. The details of the outerloop design are discussed in detail in reference [49]. The innerloop ensures that the states of the aircraft track these desired quantities using the control architectures described in Chapter 5. Results from two flight tests are presented. The aircraft is commanded to track an elliptical pattern while holding altitude at 200 ft. The baseline implementation uses a RBF NN with 10 radial basis functions whose centers are spaced with a uniform distribution in the region of expected operation. The RBF width is kept constant at 1. The baseline adaptive controller uses the following

adaptive law

$$\dot{W}(t) = -\Gamma_W \sigma(\bar{x}(t)) e^T(t) P - \kappa \|e(t)\| W(t). \tag{9.1}$$

In the above equation, $\kappa = 0.1$ denotes the gain of the e-mod term[69]. The concurrent learning adaptive controller uses the learning law of Theorem 5.3. A nominal e-mod term with $\kappa = 0.01$ is also added to the concurrent learning adaptive law ensure boundedness of weights until Condition 4.1 is met. The ground tracks of both controllers are compared in figure 9.2. In that figure, the circles denote the commanded way points, the dotted line connecting the circles denotes the path the aircraft is expected to take, except while turning at the waypoints. While turning at the waypoints, the onboard guidance law smooths the trajectory [49] by commanding circles of 80 feet radius. From that figure, it is clear that the concurrent learning adaptive controller has better cross-tracking performance. Figure 9.3 shows that the altitude tracking performance of the two controllers are similar. The inner loop tracking error performance of the baseline adaptive controller is shown in figure 4(a), while the innerloop tracking error performance of the concurrent learning controller is shown in figure 4(b). The transient performance is comparable, however, it was found that the concurrent learning controller is better at eliminating steady-state errors than the baseline adaptive controller. This is one reason why the concurrent learning controller has better cross-tracking performance than the baseline. The actuator input required for the baseline adaptive controller is shown in figure 5(a), while the actuator input required for the concurrent learning adaptive controller is shown in figure 5(b). While the peak magnitude of control input requires is comparable for both controllers, it was found that the concurrent learning adaptive controller is better as estimating steady-state trims. Hence, we conclude that the improved performance of the concurrent learning controller is mostly due to better estimation of steady state constants, which should be a result of improved weight convergence.

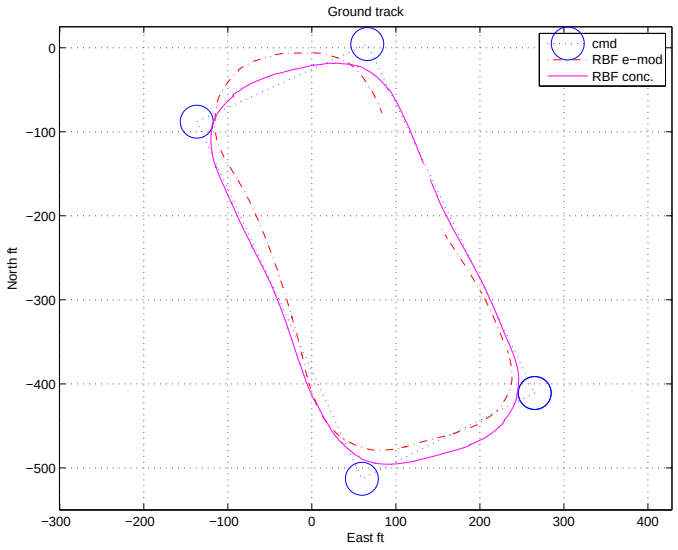


Figure 9.2: Comparison of ground track for baseline adaptive controller with concurrent learning adaptive controller. Note that the concurrent learning controller has better cross-tracking performance than the baseline adaptive controller

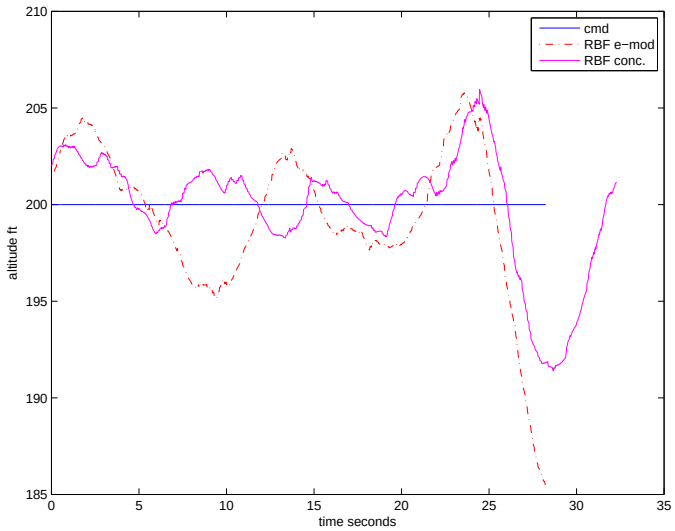
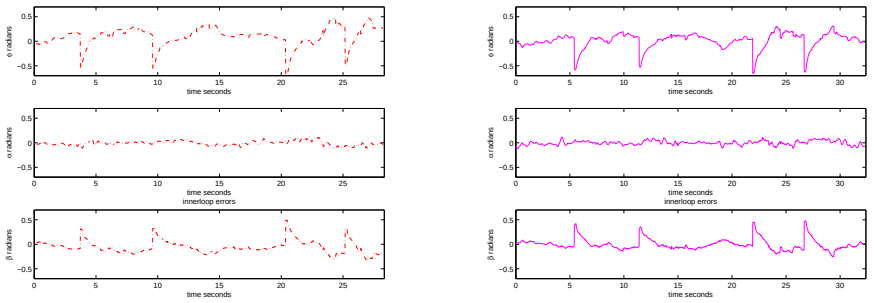


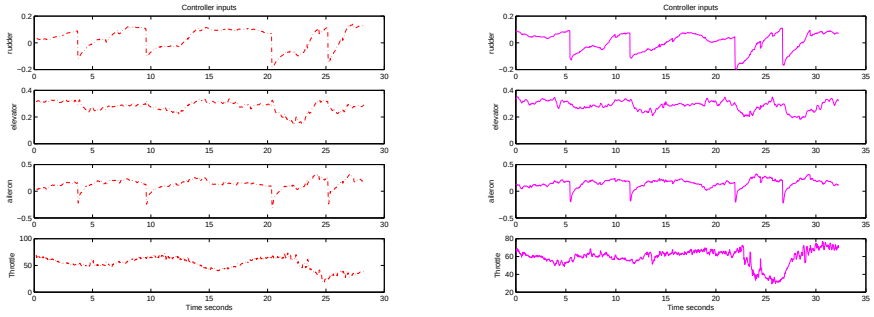
Figure 9.3: Comparison of altitude tracking for baseline adaptive controller with concurrent learning adaptive controller.



(a) Inner loop tracking errors for baseline adaptive controller

(b) Inner loop tracking errors for concurrent learning adaptive controller

Figure 9.4: Comparison of inner loop tracking errors. Although the transient performance is similar, the concurrent learning adaptive controller was found to have better trim estimation



(a) Actuator inputs for baseline adaptive controller

(b) Actuator inputs for concurrent learning adaptive controller

Figure 9.5: Comparison of actuator inputs. The concurrent learning adaptive controller was found to have better trim estimation. Note that the aileron, rudder, and elevator inputs are normalized between -1 and 1, while the throttle input is given as percentage.

CHAPTER X

APPLICATION OF CONCURRENT GRADIENT DESCENT TO THE PROBLEM OF NETWORK DISCOVERY

In this chapter, the problem of network discovery is formulated and the concurrent gradient descent method of Theorem 3.1 (Section 3.3) is proposed as a method for arriving at a solution.

10.1 MOTIVATION

Successful negotiation of real world missions often requires diverse teams to collaborate and synergistically combine different capabilities. The problem of controlling such networked teams has become highly relevant as advances in sensing and processing enable compact distributed systems with wide ranging applications, including networked Unmanned Aerial Systems (UAS), decentralized battlefield negotiation, decentralized smart-grid technology, and internet based social-networking (see for example [75], [68], [11], [27], and [74]). The development of these systems however, present many challenges as the presence of a central controlling agent with access to all the information cannot be assumed.

There have been significant advances in control of networked systems using information available only at the agent level, including reaching consensus in networked systems, formation control, and distributed estimation (see for example [75], [27]). The emphasis has been to rely only on local interactions to avoid the need for a central controlling agent. However, there are many applications where the knowledge of the global network topology is needed for making intelligent inferences. Inferences

such as identifying the interactions between agents, identifying faulty or misbehaving agents, or identifying agents that enjoy high connectivity and are in a position to influence the decisions of the networked system. This information in turn, can allow agents to make intelligent decisions about how to control a network and how to build optimal networks in real-time. The key problem that needs to be addressed for enabling the needed intelligence is: How can an agent use only information available at the agent level to make global inferences about the network topology? We term this problem as Network Discovery, and formulate the problem in the framework of estimation theory.

The idea of using measured information to gather information about the network characteristics was explored by Franceschelli et al. through the estimation of the eigenvalues of the network graph Laplacian [28]. They proposed a decentralized method for Laplacian eigenvalue estimation by providing an interaction rule that ensured that the state of the agents oscillate in such a manner such that the problem of eigenvalue estimation can be reduced to a problem of signal processing. The eigenvalues are then estimated using Fast Fourier Transforms. The Laplacian eigenvalues contains useful information that can be used to characterize the network, particularly the second eigenvalue of the Laplacian contains information on the connectivity of the network and how fast it can reach agreement. However, the knowledge of eigenvalues does not yield information about other details of the topology, including the degree of connectivity of individual agents and the graph adjacency matrix.

Agent level measurements of other agents states was used by Franceschelli, Egerstedt, and Giua for fault detection through the use of motion probes [29]. The idea behind motion probes is that individual agents perform in a decentralized way a maneuver that leaves desirable properties of the consensus protocol invariant and analyze the response of others to detect faulty or malicious agents. This work emphasized the importance of excitation in the network states for network property discovery.

It may be possible to approach the network discovery problem through the use of communication, where each agent relays the information about its connectivity to other agents, and the graph Laplacian is formed using relayed information in a decentralized manner. Muhammad and Jabdabaie have proposed using Gossip-like algorithms for minimizing communications overhead in discovering network properties through relayed information [68]. However there are various situations where communication may not be possible or cannot be trusted. For example, communications based approach may not work if some of the agents have become faulty, are unable to communicate, are maliciously relaying wrong information, or if the agent that wants to discover the network wishes to operate covertly. Hence, we restrict our attention to the development of algorithms that use information that is measured or otherwise gathered only at the agent level. Clearly the addition of communications would compliment any of the presented approaches.

Finally, we mention that the problem we are concerned with is quiet different from that of distributed estimation (see for example reference [32] and the references therein). In distributed estimation the purpose is to reach consensus about the value of an external global quantity in a decentralized manner through distributed measurements over different agents. Whereas, we are concerned with the estimation of internal network properties (particularly the rows of the graph Laplacian) through measurements.

In this section We show that under a number of assumptions the problem of network discovery can be related to that of parameter estimation. Furthermore, we propose and compare various methods that an agent can use for network discovery. We rely heavily on an algebraic graph theoretic representation of networked systems, where the network and its interconnections are represented through sets. The section is organized as follows, we begin by showing that the problem of identifying a particular agents degree of connectivity and neighbors can be reduced to that of estimating

that agent's linear consensus protocol. We then show that subject to certain assumptions, namely static network, and complete availability of information, this problem can be cast as that of parameter estimation and propose three different methods to solve the problem online. We also consider a case when the assumption of complete availability of information is relaxed.

10.2 The Network Discovery Problem

Consider a network consisting of N independent agents enabled with limited communication capabilities and operating under a protocol to reach consensus [75]. We assume that the information available to an agent is composed entirely of what it can sense, measure, or otherwise gather. A network such as this is capable of representing a wide variety of decentralized networked dynamical systems, including a collaborating group of mobile ground robots or unmanned aerial vehicles communicating through wireless datalinks, a power grid connecting distributed sources with consumers, or computer systems connected over ethernet. Such a network can be represented as a graph $G = V \times E$, with $V = 1, ..., N$ denoting the set of vertices or nodes of the network, and E denoting the set of edges $E \subset V \times V$, with the pair $(i, j) \in E$ if and only if the agents i can communicate with or otherwise sense the state of agent j. In this case, agent j is termed as a neighbor of agent i. The total number of all neighbors of an agent at time t is termed as its degree at time t. Let $Z_i \in \Re^n$ denote the state of the i^{th} agent, with $Z_i = \{z_1, z_2, z_3, ..., z_n\}$. The elements of Z_i can represent various physical quantities of interest, such as position, velocity, voltage etc. If the elements of the edge set (that is the pairs (i, j)) are unordered, the graph is termed as undirected. We will consider undirected graphs for ease of exposition, we note that an extension to the directed case is straightforward.

In the following, we will refer to the agent whose degree and neighbors are to be estimated as the *target agent*, while the agent which wishes to estimate the consensus

protocol of the target agent as the *estimating agent*. The problem of network discovery can now be formulated:

Problem 10.1 The Network Discovery Problem Use only the information available at estimating agent to determine the degree of the target agent and identify it's neighbors.

Note that multiple target and estimating agents may be present in a network. We now introduce a simplification in the notation, namely, when only one component of z_i is under consideration its identifying subscript will be dropped. Using this convention, let the vector $x = \{x_1, x_2, ..., x_N\} \in \Re^N$ contain the i^{th} element $z_i \in \Re$ of all agents. We assume that the dynamics of the target agent (agent i) is given by the following equation [27]

$$\dot{x}_i(t) = \sum_{j \in N_i} \left[x_i(t) - x_j(t) \right], \tag{10.1}$$

where the mapping $y_i(t) = \sum_{j \in N_i} [x_i(t) - x_j(t)]$ denotes the un-weighted consensus protocol of agent i [75], [27]. The preceding equation basically states that $y_i = \dot{x}_i$, and we will often drop the subscript i on y for notational convenience. Let $\zeta \in \Re^{l+1}$ denote the vector containing the states of all of agent i's neighbors where $l < N$ denotes the degree of agent i. Note that with an arbitrary numbering of the agents, the state vector x can be written as $x = [\zeta, \xi]$, where $\xi \in \Re^{N-l}$ is the vector containing the states of all the agent's in the networks which are not agent i's neighbors. Therefore, y can be also expressed as: $y = W^T x$, where the vector $W \in \Re^N$ is the i^{th} row of the instantaneous graph Laplacian [27]. Taking advantage of this fact, we denote W as the Laplacian vector of agent i. Under conditions on connectivity of the network, the consensus protocol will result in $x \to \frac{1}{N}\mathbf{1}\mathbf{1}^T x(0)$, where $\mathbf{1} = [1, 1, 1, 1..1] \in \Re^N$ [27]. In this thesis however, we are not concerned with the convergence properties of the consensus protocol. What we are concerned with, is the problem of estimating agent i's degree and neighbors (problem 10.1. Figure 10.1 depicts a network discovery

scenario where the estimating agent can sense the states of the target agent and all of its neighbors, but not all of the agents in the network.

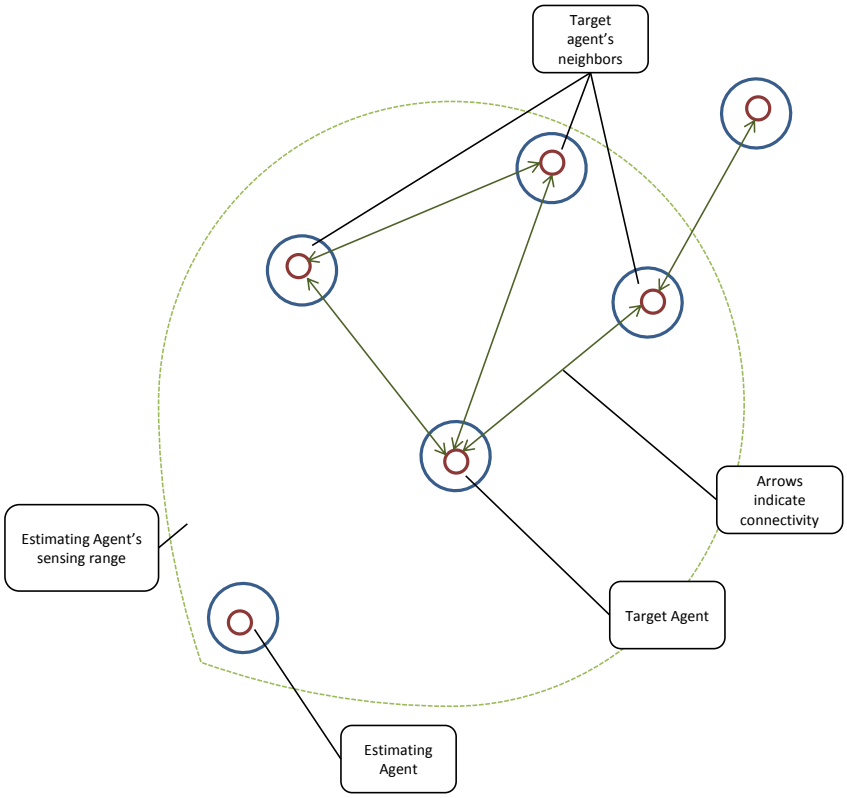


Figure 10.1: A depiction of the network discovery problem, where the estimating agent uses available measurements to estimate the neighbors and degree of the target agent. Note that the estimating agent can sense the states of the target agent and all of its neighbors, however, one agent in the target agent's network is out of the estimating agent's sensing range.

10.3 Posing Network Discovery as an Estimation Problem

Obtaining a solution to problem 10.1 in the most general case can be a quiet daunting task due to a number of reasons, including:

- The neighbors of the target agent may change with time,
- The estimating agent may not be able to sense information about all of target agent's neighbors,

- The target agent may be actively trying to avoid identification of its consensus protocol.

In order to progress, we will make the following simplifying assumption.

Assumption 10.1 Assume that the network edge set does not change for a predefined time interval $\Delta(t)$, that is the network is *slowly varying*.

The above assumption requires that within a time interval $\Delta(t)$, $W(t) = W$, that is the Laplacian vector $W(t)$ is time invariant for a predefined amount of time. That is, we require that the network topology be "slowly" varying. Such slowly varying networks can be used to model many real-world networked systems. This assumption allows us to cast the problem of network discovery as a problem of estimating the Laplacian vector of the target agent. The Laplacian vector contains the information about the degree of agent i and its adjacency to other agents in the network, information that can be used to solve the network discovery problem. The interval is expected to be sufficiently large such that estimation algorithms can arrive at a solution, and the length of the interval depends on the choice of the algorithm. Let $\bar{x} \in \Re^k$ contain the measurements of the states of agents that are available to the estimating agent. Note that without loss of generality we can assume that $k \leq N$, for if $k > N$, then we can always set $N = k$. In essence, the estimating agent assumes that all of the agents it can measure are a part of the network. Then, letting $\hat{W} \in \Re^k$ the following estimation model can be used for estimating W

$$\nu(t) = \hat{W}^T(t)\bar{x}(t). \tag{10.2}$$

Recalling that $y(t) = W^T(t)x(t)$ the estimation error can be formulated as

$$\epsilon(t) = \nu(t) - y(t) = \hat{W}^T(t)\bar{x}(t) - W^T x(t). \tag{10.3}$$

One way to approach the network discovery problem, is to design a weight law $\dot{\hat{W}}(t)$ such that $\epsilon(t) \to 0$ uniformly as $t \to \infty$, or $\epsilon(t)$ is identically equal to zero after

some time T. That is $\epsilon(t) = 0\ \forall t > T$ (it follows that $\epsilon(t) = 0\ \forall x(t)\ t > T$ if $\epsilon(t)$ is identically equal to zero). The following proposition shows that if the estimating agent cannot measure the states of all of the target agent's neighbors, then $\epsilon(t)$ cannot be identically equal to zero.

Proposition 10.1 Consider the estimation model of equation 10.2 and the estimation error ϵ of equation 10.3, and suppose $\bar{x}$ does not contain the state measurements of all of the target agent's neighbors, then $\epsilon(t)$ cannot be identically equal to zero.

Proof Ignoring the irrelevant case when the target agent has no neighbors, let $\zeta \in \Re^m$ denote the vector containing all of target agent's neighbors. Then letting i denote the identifying subscript for the target agent, and deg_i denote the degree of i we have that $y(t) = \dot{x}_i(t) = [-1, -1, ..., deg_i, ..., -1]^T \zeta(t) = \check{W}^T \zeta(t)$. Therefore the vector $\check{W} \in \Re^m$ contains only nonzero elements. Let $\bar{x} \in \Re^k$, and assume that $k < m$ (the case when $k > m$ follows in a similar manner), furthermore, let $\zeta = [\bar{x}, \xi]$, with $\xi \in \Re^{m-k}$. Suppose *ad absurdum* $\epsilon(t)$ is identically equal to zero, then we have that

$$\nu(t) - y(t) = [\hat{W}(t), 0..0]^T \begin{bmatrix} \bar{x}(t) \\ \xi(t) \end{bmatrix} - \check{W}\zeta(t) = 0. \tag{10.4}$$

Since we claim that $\epsilon(t)$ is identically equal to zero, then in the nontrivial case (i.e. $\zeta(t) \neq 0$) we must have that $[\hat{W}(t), 0..0] - \check{W} = 0$, for all $t > T$ in order to satisfy equation 10.4. Therefore $\check{W}$ must contain $m - l$ zero elements, which contradicts the fact that $\check{W}$ contains only nonzero elements. Hence, if $\bar{x}$ does not contain the state measurements of all of the target agent's neighbors, then $\epsilon(t)$ cannot be identically equal to zero. ∎

Remark 10.1 Note that in the above proof we ignored the case when $\zeta(t)$ is identically equal to zero. If $\zeta(t)$ is identically equal to zero then the states of all

agents have converged to the origin, an unlikely prospect, considering the consensus equation only guarantees $x \to span(\mathbf{1})$ as $t \to \infty$. Another unlikely but interesting case arises when $\zeta(t)$ is such that $[\hat{W}(t), 0..0] - \check{W} \perp \zeta(t)\ \forall t > T$. In both these cases, one can argue that the states $\zeta(t)$ do not contain sufficient excitation, and proposition 10.1 becomes irrelevant. The importance of excitation in the states for solving the network discovery problem is explored further in Section 10.4.

Remark 10.2 Proposition 10.1 formalizes a fundamental obstruction to obtaining a solution to the problem of network discovery: If the estimating agent cannot measure or otherwise know the states of the target agent's neighbors, then an estimation based approach alone cannot be used to solve the network discovery problem.

Therefore, we have shown that in order to use the estimation model of equation 10.2 to solve the network discovery problem, the following assumption must be satisfied:

Assumption 10.2 Assume that the estimating agent can measure or otherwise perceive the position of all of the target agent's neighbors.

The following theorem shows that if a weight update law $\dot{\hat{W}}(t)$ exists such that $\epsilon(t)$ can be made identically equal to zero, then a solution to the network discovery problem (problem 10.1) can be found.

Theorem 10.2 Consider the estimation model of equation 10.2 and the estimation error ϵ of equation 10.3, let assumption 10.2 hold, assume that the network edge set does not change for a predefined time interval (assumption 10.1), and $x(t)$ is not identically equal to zero, then finding a weight update law $\dot{\hat{W}}(t)$ such that $\epsilon(t)$ becomes identically equal to zero (that is $\epsilon(t) = 0\ \forall t > T$), is equivalent to finding a solution to the network discovery problem 10.1.

Proof Suppose there exists a weight update law $\dot{\hat{W}}(t)$ exists such that $\epsilon(t)$ becomes identically equal to zero. Since assumption 10.2 holds, we can arbitrarily reorder the states such that $\bar{x} = [\zeta, \xi]$, where ξ denote the states of the agents which are not neighbors of the target agent, hence we have

$$\nu - y = \hat{W}^T(t)\bar{x}(t) - [W, 0..0]^T \begin{bmatrix} \zeta \\ \xi \end{bmatrix} = 0. \tag{10.5}$$

Letting $\tilde{W} = \hat{W} - [W, 0..0]$, we have

$$\nu(t) - y(t) = \tilde{W}(t)\bar{x}(t) = 0. \tag{10.6}$$

Since $x(t)$ is assumed to be not identically equal to zero, in the nontrivial case we must have that $\tilde{W}(t) = 0 \ \forall t > T$. Therefore it follows that $\hat{W} = [W, 0..0]$ contains the Laplacian vector of the target agent, which is sufficient to identify the degree and neighbors of the target agent. ∎

Remark 10.3 As in the proof of proposition 10.1, an interesting but unlikely case arises when $\tilde{W}(t) \perp \bar{x}(t) \ \forall t$. Once again this relates to a notion of sufficient excitation in the system states and is further explored in Section 10.4.

To simplify the notation a little bit, we can let $\bar{x} = x$, this is equivalent to saying that the estimating agent can measure states of all of the agents that affect the target agent. Due to Theorem 10.2, this is equivalent to saying that for the purpose of the network discovery problem, the network can be assumed to be made of only the agents that either interact with the target agent or are visible to the estimating agent. Hence, this change in notation does not affect the structure of the problem, except that we now have $\epsilon(t) = \nu(t) - y(t) = \hat{W}^T(t)x(t) - W^T x(t) = \tilde{W}x$, which is simpler to deal with. In this case, the Laplacian vector of the target agent W will contain zero elements corresponding to agents that the target agent is not connected to.

Through the above discussion ,we have essentially shown that subject to assumption 10.1 and 10.2 the network discovery problem can be cast as the following simpler problem

Problem 10.2 Let an estimation model for the network discovery problem be given by equation 10.2, and the estimation error be given by equation 10.3. Design an update law $\dot{\hat{W}}$ such that $\hat{W}(t) \to W$ as $t \to \infty$.

In this way, we have reduced the network discovery problem to that of a parameter estimation problem. Various approaches have been proposed for online parameter estimation in the literature. In the following we will highlight three such approaches.

10.4 Instantaneous Gradient Descent Based Approach

In this simplest and most widely studied approach $\hat{W}$ is updated in the direction of maximum reduction of the instantaneous quadratic cost $V(\epsilon(t)) = \epsilon^2(t)$. That is, letting Γ be a positive learning rate we have $\dot{W} = -\gamma \frac{\partial V}{\partial \hat{W}}$. This results in the following update law

$$\dot{\hat{W}}(t) = -\Gamma x(t)\epsilon(t). \tag{10.7}$$

The convergence properties of the gradient descent based approach have been widely studied, it is well known that for this case persistency of excitation (see definition 3.2) in $x(t)$ is a necessary and sufficient condition for ensuring $\hat{W}(t) \to W$ as $t \to \infty$ exponentially [1],[3],[70],[93].

Note that Definition 3.2 requires that the matrix $\int_t^{t+T} x(\tau)x^T(\tau)d\tau$ be positive definite over all future predefined finite time intervals. As an example, consider that in the two dimensional case, vector signals containing a step in every component are exciting, but not persistently exciting; whereas the vector signal $x(t) = [\sin(t), \cos(t)]$ is persistently exciting. Hence, in order to ensure that $\tilde{W} \to 0$ as $t \to \infty$, we must ensure that the system states $x(t)$ are persistently exciting. However, there is no

guarantee that the network state vector $x(t)$ would be exciting if the network is only running the consensus protocol of equation 10.1. For example, the following fact shows that if the initial state of the network happens to be an eigenvector, then the system states are not persistently exciting.

Fact 10.3 The solution $x(t)$ to the consensus equation $\dot{x}(t) = -Lx(t)$, where L is the graph Laplacian, need not be persistently exciting for all choices of $x(0)$.

Proof

Let $x(0)$ and $\lambda \in \Re$ be such that $Lx(0) = \lambda x(0)$, that is let $x(0)$ be an eigenvector of L. Then we have $x(t) = e^{-\lambda t}x(0)$, hence

$$\int_t^{t+T} x(\tau)x^T(\tau)d\tau = \int_t^{t+T} e^{-2\lambda t}x(0)x^T(0), \tag{10.8}$$

which is at-most rank 1, and hence not positive definite over any interval.

∎

Therefore, an external forcing term will be needed to enforce persistency of excitation in the system. The consensus protocol can then be written as

$$\dot{x}_i(t) = \sum_{j \in N_i} x_i(t) - x_j + f(x_i(t), t), \tag{10.9}$$

where $f(x_i(t), t)$ is a known bounded mapping $\Re^2 \to \Re$ used to insert excitation into the system. In its most simplest form $f(x_i(t), t)$ can simply be a random sequence of numbers, or it could be an elaborate periodic pattern (such as in [29]) which is known over the network.

With the details of the algorithm in place, we evaluate its performance through simulation on a network containing 9 nodes with each of the nodes updated by equation 10.9, for solving the network discovery problem. It is assumed that $f(x_i(t), t)$ is a known Gaussian random sequence with an intensity of 0.01 and that $y_i(t) = \dot{x}_i(t) - f(x_i(t), t)$ can be measured. Note that the chosen $f(x_i(t), t)$ does introduce

persistent excitation in the networked system. The agents are arbitrarily labeled, and the third agent is arbitrarily picked as the estimating agent, and it estimates the consensus protocol for the second agent (which is the target agent). The Laplacian vector for the target agent is given by $W = [0, -3, 1, 0, 0, 1, 1, 0, 0]$, and its consensus protocol will have the form $y_i = W^T x$. The target agent has 3 neighbors (i.e. degree of i is 3), they are agent $3, 6$, and 7. Figure 10.2 shows the performance of the gradient descent algorithm for the network under consideration with $\Gamma = 10$. It can be seen that the algorithm is unsuccessful in estimating the Laplacian vector for W by the end of the simulation, even when persistent excitation is present. Increasing the learning rate Γ may slightly speed up the convergence, however the key condition required is that the $x(t)$ remain persistently exciting such that the scalar γ in definition 3.2 is large. That is, the convergence is dependent not only on the existence of excitation, but also on its magnitude.

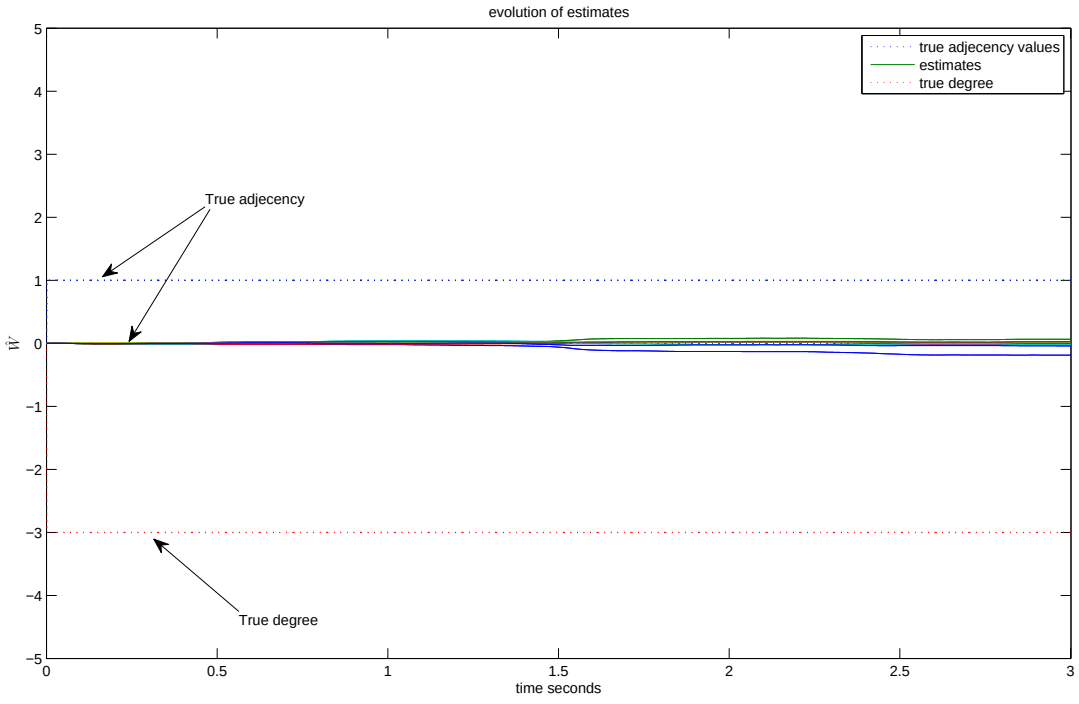


Figure 10.2: Consensus estimation problem with gradient descent

10.5 Concurrent Gradient Descent Based Approach

In the previous section we noted that the gradient descent algorithm is susceptible to being stuck at local minima, and requires persistency of excitation in the system signals to guarantee convergence. For many networked control applications the condition on persistency of excitation is infeasible to monitor online, particularly since the trajectories of individual agents are not known a-priori. On examining equation 10.7 we see that the update law uses only instantaneously available information $(x(t), \epsilon(t))$ for estimation. If the update law used specifically selected and recorded data concurrently with current data for adaptation, and if the recorded data were sufficiently rich, then intuitively it should be possible to guarantee $\hat{W} \to W$ as $t \to \infty$ without requiring persistently exciting $x(t)$.

The concurrent gradient descent algorithm of Theorem 3.1 can be used to leverage this intuitive concept. Let $j \in \{1, 2, ...p\}$ denote the index of a stored data point x_j, let $\epsilon_j = \tilde{W}^T x_j$, let denote a positive definite learning rate matrix, then the concurrent learning gradient descent algorithm for this application is given by

$$\dot{W}(t) = -\Gamma x(t)\epsilon(t) - \sum_{i=1}^{p} \Gamma x_j \epsilon_j. \tag{10.10}$$

The parameter error dynamics $\tilde{W}(t) = \hat{W}(t) - W$ for this case can be expressed as follows

$$\begin{aligned} \dot{\tilde{W}}(t) &= -\Gamma x(t)\epsilon(t) - \Gamma \sum_{j=1}^{p} x_j \epsilon_j \\ &= -\Gamma x(t)x(t))\tilde{W}(t) - \Gamma \sum_{j=1}^{p} x_j x_j^T \tilde{W}(t) \\ &= -\Gamma[x(t)x(t)) + \sum_{j=1}^{p} x_j x_j^T]\tilde{W}(t). \end{aligned} \tag{10.11}$$

The concurrent use of current and recorded data has interesting implications, as the exciting term $f(x_i, t)$ will not need to be persistently exciting, but only exciting over a finite period such that rich data can be recorded. In fact, we have already shown

that the recorded data x_j need only be linearly independent in order to guarantee weight convergence (3.1). This condition on sufficient richness of the recorded data for this application is captured in the following statement

Condition 10.1 The recorded data has as many linearly independent elements as the dimension of the basis of the uncertainty. That is, if $Z = [x_1, ..., x_p]$, then $rank(Z) = m$.

This condition is easier to monitor online and essentially requires that the recorded data contain sufficiently different elements to form the basis of the state space. The following theorem can now be proved.

Theorem 10.4 Consider the estimation model of equation 10.2, the estimation error ϵ of equation 10.3, the weight update law of equation 10.10, and assume that assumptions 10.1 and 10.2 are satisfied. If Condition 10.1 is satisfied, then the zero solution of parameter error dynamics $\tilde{W} \equiv 0$ of equation 10.11 is globally uniformly exponentially stable when using the concurrent learning gradient descent weight adaptation law of equation 10.10.

Proof A proof can be formed in an equivalent manner to proof of Theorem 3.1. ∎

We now evaluate the performance of the concurrent learning gradient descent algorithm on the networked system simulation setup described in Section 10.4. Figure 10.3 shows the performance of the concurrent gradient descent algorithm for the network under consideration with $\Gamma = 10$. The simulation began with no recorded points, at each time step, the state vector $x(t)$ was scanned online, and points satisfying the condition $\|Z^T x(t)\| < 0.5$ or $y(t) - \nu(t) > 0.3$ were selected for storage. Condition 3.1 was found to be satisfied within 0.1 seconds into the simulation. It can be seen that the algorithm is successful in estimating the Laplacian vector for W, and thus in estimating the degree of the third agent and the identity of its neighbors. Hence, the

algorithm outperforms the traditional gradient descent based method (Section 10.4) with the same level of enforced excitation. In general, the speed of convergence will be dependent on the minimum eigenvalue of the matrix ZZ^T and to a lesser extent, the learning rate Γ. That is, ideally we would like the stored data to not only be linearly independent, but also be sufficiently different in order to maximize the minimum singular value of Z. At the end of the simulation the minimum singular value was found to be 1.58.

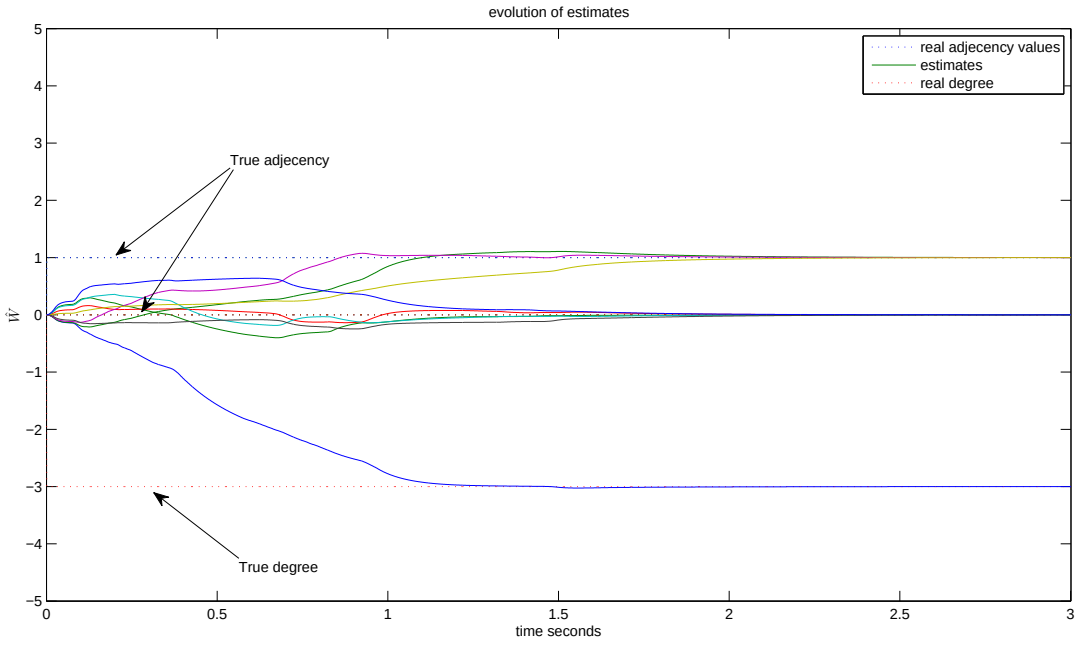


Figure 10.3: Consensus estimation problem with concurrent gradient descent

CHAPTER XI

CONCLUSIONS AND SUGGESTED FUTURE RESEARCH

The key contribution of this thesis was to show that memory (recorded data) can be used to guarantee convergence in a class of adaptive control problems without requiring Persistently Exciting (PE) exogenous inputs. To that effect we presented a method termed as concurrent learning which uses recorded data concurrently with current data to guarantee global exponential convergence to zero of the tracking error and parameter error dynamics in model reference adaptive control subject to a simple condition on linear independence of the recorded data. The presented condition requires that the recorded data have as many linearly independent elements as the dimension of the basis of the uncertainty. Lyapunov analysis was used to show that meeting this condition is sufficient to guarantee global exponential parameter convergence in parameter estimation problems with linearly parameterized estimation models when using concurrent learning. It was also shown that meeting the same condition is sufficient to guarantee global exponential stability of the zero solution of the tracking error and parameter error dynamics in adaptive control problems with structured linearly parameterized uncertainty when using concurrent learning. For this class of problems it was also shown that if the adaptive law prioritizes weight updates based on current data by restricting weight updates based on recorded data to the nullspace of weight updates based on current data, then meeting the same condition is sufficient to guarantee global asymptotic stability of the zero solution of the tracking error and parameter error dynamics. For adaptive control problems where

the structure of the uncertainty is unknown and neural networks are used to capture the uncertainty, it was shown that the same condition is sufficient to guarantee uniform ultimate boundedness of the parameter and tracking error.

Classical result for exponential convergence in adaptive control requires the exogenous input signal to have as many spectral lines as the dimension of the basis of the uncertainty (Boyd and Sastry 1986) and is well justified for adaptive controllers that use only current data for adaptation. The results in this thesis show that if both recorded and current data are used concurrently for adaptation then the condition for weight convergence relates directly to the spectrum of the recorded data. In essence, these results formalize the intuitive argument that if sufficiently rich data is available for concurrent adaptation, then weight convergence can occur without system states being *persistently* exciting. The presented condition on linear independence of the recorded data is found to be less restrictive than a condition on PE exogenous input and allows a reduction in the overall control effort required. Furthermore, unlike a condition on PE exogenous inputs, this condition is easily verified online. Finally, the additional computational overhead required for concurrent adaptation is easily handled by modern embedded computer systems. For these reasons, we believe that the presented adaptive control methods can be applied directly to improve the control performance in control of various physical plants. Furthermore, the concurrent gradient descent method described for convergence without PE states could be extended beyond adaptive control to a wide variety of control and optimization problems.

11.1 Suggested Research Directions

11.1.1 Guidance algorithms to ensure that the rank-condition is met

In this work, for the case of structured uncertainty, we showed that Condition 3.1 (Rank-Condition) is sufficient to guarantee the convergence of the adaptive weights to their ideal weights (or to a neighborhood of the ideal weights if the uncertainty is

unstructured and a neural network is used as the adaptive element). Furthermore, we showed in Theorems 3.2 and 5.1 that the rate of convergence is directly related to the minimum singular value of the history-stack $Z_k = [\Phi_1, ..., \Phi_p]$. An interesting future research direction is to design guidance laws to ensure that the rank-condition is met as soon as possible, and $\lambda_{\min}(\Omega)$ is maximized. One way to achieve this would be to find the nullspace of the recorded data points in the history-stack and generate trajectories online such that new data points can be recorded in the nullspace of the current history-stack. This approach would essentially enforce excitation in the directions that have not been recorded. The idea here differs from other ideas such as "intelligent excitation" developed by Cao and Hovakimyan [13]. In intelligent excitation, excitation is imposed as a function of the tracking error, whereas in this approach excitation would be inserted only in the direction in which it is needed, thereby minimizing unnecessary excitation.

As a simple example, assume that the mapping $\Phi : \Re^n \to \Re^m$ is invertible, and let Q be the nullspace of the history-stack, that is $Q = \{\Phi(x) : Z_k\Phi(x) = 0\}$. Then a simple guidance logic would be to select a feasible vector $\Phi_k \in Q$ and invert the mapping Φ to obtain the state x that is to be commanded by an existing guidance algorithm.

11.1.2 Extension to Dynamic Recurrent Neural Networks

Dynamic Recurrent NN DRNN, also known as differential NN, have at least one internal feedback loop. In this aspect, they differ significantly from the static NN studied in this thesis. Many authors believe that these internal feedback loops make DRNN better suited for approximating dynamical systems (see references in [78]). These NN can model dynamical systems with time-delay, internal feedback, and hysteresis. A particularly interesting application of DRNN arises in output feedback adaptive control. In these applications, it may be possible to model the dynamical system

with a DRNN and train the DRNN with the system outputs. If the estimate of the dynamic system converges, then the output feedback problem can be solved using a direct control methodology without having to solve the state estimation problem explicitly. However, the most common training laws proposed for training DRNN are gradient based, and hence, do not guarantee parameter error convergence unless conditions equivalent to persistency of excitation are met. An interesting extension of this work would be the extension of concurrent learning adaptive laws to DRNN and the development of conditions on the recorded data to guarantee parameter error convergence. Furthermore, while these NN have been studied to some extent in other control applications, not many applications of DRNN based adaptive flight control exist. It is suggested that DRNN based adaptive flight controllers be developed to realize the benefit of internal feedback.

11.1.3 Algorithm Optimization and Further Flight Testing

In this work, the developed concurrent learning adaptive controllers were implemented on a number of research aircraft. In all cases, some improvement in performance was seen, this is an encouraging sign for further testing and development of concurrent learning adaptive flight controllers. Further optimization of elements of the controller is expected to further improve this performance. Efforts should be spent on developing and optimizing algorithms for picking data points to record and to manage the history-stack. For example, in Chapter 6 we presented a brute-force algorithm for determining whether a new data point should replace an existing data point in the history-stack. This algorithm however, requires the computation of the singular values of the history stack matrix, which can be computationally expensive.

11.1.4 Quantifying the Benefits of Weight Convergence

In this work we showed that concurrent learning adaptive controllers can guarantee tracking error and weight convergence subject to a verifiable condition on the recorded

data. For the case of structured uncertainty, once the weights converge, the tracking error dynamics are linear and exponentially stable. This guarantees that the states of the plant track the states of the reference model exponentially. It remains to be shown rigorously whether this guarantees that the chosen transient response and stability properties of the reference model are recovered by the adaptive controller. Research in this direction can lead to adaptive controllers for nonlinear systems guaranteed to recover the stability and performance margins of a chosen linear system. Furthermore, such weight convergence in adaptive flight control allows one to use handling specifications such as those in reference [89], enabling a pathway to flight certification of adaptive controllers.

11.1.5 Extension to Other Adaptive Control Architectures

Another research direction of interest is to combine concurrent learning algorithms with other adaptive control methods and architectures. In Theorem 5.6 we showed that concurrent learning can be added to a baseline adaptive controller equipped with e-mod. Research is suggested in combining other modifications to adaptive control with concurrent learning algorithms, including ALR modification [12] and Kalman Filter modification [99]. Another method of particular interest is Q modification, which relies on an integral of the tracking error over a finite window of past data to drive the weights to a hypersurface that contains the ideal weights [96, 95]. Further research is suggested in exploring the similarities and differences between Q modification and concurrent learning adaptive control.

11.1.6 Extension to Output Feedback Adaptive Control

In this thesis, we assumed that the complete state of the plant was available for measurement. This is normally true for aircraft, where sensors are often available to measure all the states of interest, and the cost of instrumentation is justified to reduce risks. However, in other applications, such as active structural control, or

control of multi-joint robot arms, it may be infeasible to assume that all of the states are available for measurement. In such applications, output feedback adaptive control holds great promise. Research is therefore suggested to extend the concurrent learning framework to output feedback adaptive control.

One interesting research direction is to explore whether concurrent learning can be used in existing based output feedback adaptive control architectures. Hovakimyan et al. have presented an output feedback method applicable to non-minimum phase systems with parametric uncertainty and unmodeled dynamics whose non-minimum phase zeros are known with sufficient uncertainty (see for example references [39] and [41]). The method uses a neural network trained using the observed errors of the system for mitigating modeling error. Research is suggested to examine whether concurrent learning can bring performance gains in similar architecture.

11.1.7 Extension to Fault Tolerant Control and Control of Hybrid/Switched Dynamical Systems

In this thesis, we assumed that the plant uncertainty can be modeled using an adaptive element for which a set of static ideal weights exist. However, if the dynamics of the plant exhibit switching, this assumption no longer holds. For example, if an aircraft undergoes severe structural damage, the modeling uncertainty can change significantly, possibly voiding an existing assumed parametrization, and making the recorded set of data irrelevant. Concurrent learning algorithms that prioritize training on current data over that of training on recorded data (such as those presented in Theorems 3.3 and 5.2) ensure that under these situations the tracking error will still remain bounded. What is needed however, is a method for detecting such drastic changes in the system dynamics and a method for using this information to repopulate the history-stack. This can be achieved through further research in health monitoring. In reference [18] for example, we proposed a frequency domain method for detecting oscillations in the control loop. We also showed that this method could be used to

detect sudden loss of part of the wing. Furthermore, such health monitoring tools will also enable the extension of concurrent learning adaptive control to control of switched/hybrid dynamical systems.

11.1.8 Extension of Concurrent Learning Gradient Descent beyond Adaptive Control

Gradient descent has been widely studied as a fast and efficient method for solving optimization problems online. However, it is well known that gradient descent based method are susceptible to being stuck at local minima, and their performance depends on the richness of the information available online. In Chapter 3 we showed that concurrent learning gradient descent on quadratic cost can guarantee convergence without requiring persistency of excitation. A suggested research direction therefore is to further explore the use of concurrent learning gradient descent algorithms for applications beyond adaptive control. A particular area of interest is networked control, in which agent level information (local information) must be used to find minima of cost functions defined over the entire network (global minima). In Chapter 10 we showed that concurrent learning yields excellent result when used to solve the network discovery problem. Further research is suggested to explore development and application of concurrent learning theory for problems in networked control.

Another area of interest is Artificial Intelligence and Machine Learning, where NN have often been used to solve classification and estimation problems. In this thesis, we used Lyapunov framework to analyze concurrent gradient descent laws. Further research is suggested in using other frameworks, such as Reproducing Kernel Hilbert Spaces [2] to improve understanding of the benefits of inclusion of memory in control, estimation, and classification algorithms.

APPENDIX A

OPTIMAL FIXED POINT SMOOTHING

Numerical differentiation for estimation of state derivatives suffers from high sensitivity to noise. An alternate method is to use a Kalman filter based approach. Let x, be the state of the system and $\dot{x}$ be its first derivative, and consider the following system:

$$\begin{pmatrix} \dot{x} \\ \ddot{x} \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} x \\ \dot{x} \end{pmatrix} \tag{A.1}$$

Suppose x is available as sensor measurement, then an observer in the framework of a Kalman filter can be designed for estimating $\dot{x}$ from available noisy measurements using the above system. Optimal Fixed Point Smoothing is a non real time method for arriving at a state estimate at some time t, where $0 \leq t \leq T$, by using all available data up to time T. Optimal smoothing combines a forward filter which operates on all data before time t and a backward filter which operates on all data after time t to arrive at an estimate of the state that uses all the available information. This appendix presents brief information on implementation of optimal fixed point smoothing; the interested reader is referred to Gelb [31] for further details. For ease of implementation on modern avionics, we present the relevant equations in the discrete form. Let $\hat{x}_{(k|N)}$ denote the estimate of the state $x = [\ x \quad \dot{x}\ ]^T$, let Z_k denote the measurements, $(-)$ denote predicted values, and $(+)$ denote corrected values, dt denote the discrete time step, Q and R denote the process and measurement noise covariance matrices respectively, while P denotes the error covariance matrix. Then

the forward Kalman filter equations can be given as follow:

$$\Phi_k = e^{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}_{dt}}, \tag{A.2}$$

$$Z_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{pmatrix} x \\ \dot{x} \end{pmatrix}, \tag{A.3}$$

$$\hat{x}_k(-) = \Phi_k \hat{x}_{k-1}, \tag{A.4}$$

$$P_k(-) = \Phi_k P_{k-1} {\Phi_k}^T + Q_k, \tag{A.5}$$

$$K_k = P_k(-) {H_k}^T [H_k P_k(-) {H_k}^T + R_k]^{-1}, \tag{A.6}$$

$$\hat{x}_k(+) = \hat{x}_k(-) + K_k[Z_k - H_k \hat{x}_k(-)], \tag{A.7}$$

$$P_k(+) = [I - K_k H_k] P_k(-). \tag{A.8}$$

The smoothed state estimate can be given as:

$$\hat{x}_{k|N} = \hat{x}_{k|N-1} + B_N[\hat{x}_N(+) - \hat{x}_N(-)], \tag{A.9}$$

where $\hat{x}_{k|k} = \hat{x}_k$.

REFERENCES

[1] Anderson, B., "Exponential stability of linear equations arising in adaptive identification," *IEEE Transactions on Automatic Control*, vol. 22, pp. 83–88, Feb 1977.

[2] Aronszajn, N., "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, pp. 337–404, may 1950.

[3] Aström, K. J. and Wittenmark, B., *Adaptive Control*. Readings: Addison-Weseley, 2 ed., 1995.

[4] Bayard, D., Spanos, J., and Rahman, Z., "A result on exponential tracking error convergence and persistent excitation," *IEEE Transactions on Automatic Control*, vol. 43, no. 9, pp. 1334–1338, 1998.

[5] Berberian, S. K., *Introduction to Hilbert spaces*. AMS Chelsea publication, 1961.

[6] Bernstein, D. and Wassim, H., *Control-System synthesis: The Fixed Structure Approach*. Atlanta, GA: Georgia Tech Book Store, 1995.

[7] Bogdanov, A., Carlsson, M., Harvey, G., Hunt, J., Kieburtz, D., Van Der Merwe, R., and Wan, E., "State dependent riccatti equation control of a small unmanned helicopter," in *Proceedings of Guidance Navigation and Control conference*, American Institute of Aeronautics and Astronautics, 2003.

[8] Boskovich, B. and Kaufmann, R. E., "Evolution of the honeywell first-generation adaptive autopilot and its applications to f-94, f-101, x-15, and x-20 vehicles," *AIAA Journal of Aircraft*, vol. 3, no. 4, pp. 296–304, 1966.

[9] Boyd, S. and Sastry, S., "Necessary and sufficient conditions for parameter convergence in adaptive control," *Automatica*, vol. 22, no. 6, pp. 629–639, 1986.

[10] Bretscher, O., *Linear Algebra with Applications*. Prentice Hall, 2001.

[11] Bullo, F., Cortés, J., and Martínez, S., *Distributed Control of Robotic Networks*. Applied Mathematics Series, Princeton University Press, 2009. Electronically available at http://coordinationbook.info.

[12] Calise, A., Yucelen, T., Muse, J., and Yang, B. J., "A loop recoevery method for adaptive control," in *Proceedings of the AIAA Guidance Navigation and Control Conference, held at Chicago, IL*, 2009.

[13] Cao, C. and Hovakimyan, N., "Design and analysis of a novel adaptive control architecture with guaranteed transient performance," *Automatic Control, IEEE Transactions on*, vol. 53, pp. 586 –591, march 2008.

[14] Cao, C., Hovakimyan, N., and Wang, J., "Intelligent excitation for adaptive control with unknown parameters in reference input," *IEEE Transactions on Automatic Control*, vol. 52, pp. 1525 –1532, Aug 2007.

[15] Cao, C. and Hovakimyan, N., "L1 adaptive output feedback controller for systems with time-varying unknown parameters and bounded disturbances," in *Proceedings of American Control Conference*, (New York), 2007.

[16] Castillo, C., Alvis, W., Castillo-Effen, M., Valavanis, K., and W., M., "Small scale helicopter analysis and controller design for non-aggressive flights," in *58th AHS Forum*, (Montreal, Canada), 2002.

[17] Chowdhary, G., Debusk, W., and Johnson, E., "Real-time system identification of a small multi-engine aircraft with structural damage," in *AIAA Infotech@Aerospace*, 2010.

[18] Chowdhary, G., Srinivasan, S., and Johnson, E., "Frequency domain method for real-time detection of oscillations," in *AIAA Infotech@Aerospace*, 2010. *Nominated for best student paper award.*

[19] Chowdhary, G. V. and Johnson, E. N., "Adaptive neural network flight control using both current and recorded data," in *Proceedings of the AIAA Guidance Navigation and Control Conference, held at Hilton Head Island, SC*, 2007.

[20] Chowdhary, G. V. and Johnson, E. N., "Theory and flight test validation of long term learning adaptive flight controller," in *Proceedings of the AIAA Guidance Navigation and Control Conference*, (Honolulu, HI), 2008.

[21] Christophersen, H. B., Pickell, W. R., Neidoefer, J. C., Koller, A. A., Kannan, S. K., and Johnson, E. N., "A compact guidance, navigation, and control system for unmanned aerial vehicles," *Journal of Aerospace Computing, Information, and Communication*, vol. 3, May 2006.

[22] Chwodhary, G. and Johnson, E., "Theory and flight test validation of a concurrent learning adaptive controller," *Journal of Guidance Control and Dynamics*, 2010. accepted.

[23] Debusk, W., Chowdhary, G., and Eric, J., "Real-time system identification of a small multi-engine aircraft," in *Proceedings of AIAA AFM*, 2009.

[24] Dorsey, J., *Continuous and Discrete Control Systems*. Singapore: McGraw-Hill Higher Education, 2002.

[25] Duarte, M. A. and Narendra, K. S., "Combined direct and indirect approach to adaptive control," *IEEE Transactions on Automatic Control*, vol. 34, no. 10, pp. 1071–1075, 1989.

[26] Dydek, Z., Annaswamy, A., and Lavretsky, E., "Adaptive control and the nasa x-15-3 flight revisited," *Control Systems Magazine, IEEE*, vol. 30, pp. 32 –48, june 2010.

[27] Egerstedt, M. and Mesbahi, M., *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[28] Franceschelli, M., Gasparri, A., Giua, A., and Seatzu, C., "Decentralized laplacian eigenvalues estimation of the network topology of a multi-agent system," in *IEEE Conference on Decision and Control*, 2009.

[29] Franceschelli, M., M., E., and Giua, A., "Motion probes for fault detection and recovery in networked control systems," in *American Control Conference*, 2008.

[30] Frazzoli, E., Dahleh, M. A., and Feron, E., "A hybrid control architecture for aggressive maneuvering of autonomous helicopters," in *IEEE Conf. On Decision and Control*, 1999.

[31] Gelb, A., *Applied Optimal Estimation*. Cambridge: MIT Press, 1974.

[32] Gupta, V., *Distributed Estimation and Control in Networked Systems*. PhD thesis, California Institute of Technology, 2006.

[33] Haddad, W. M., Volyanskyy, K. Y., Bailey, J. M., and Im, J. J., "Neuroadaptive output feedback control for automated anesthesia with noisy eeg measurements," *IEEE Transactions on Control Systems Technology*, 2010. to appear.

[34] Haddad, W. M. and Chellaboina, V., *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*. Princeton: Princeton University Press, 2008.

[35] Hayakawa, T., Haddad, W., and Hovakimyan, N., "Neural network adaptive control for a class of nonlinear uncertain dynamical systems with asymptotic stability guarantees," *IEEE Transactions on Neural Networks*, vol. 19, pp. 80 –89, jan. 2008.

[36] Haykin, S., *Neural Networks a Comprehensive Foundation*. Upper Saddle River: Prentice Hall, USA, 2 ed., 1998.

[37] Holzel, M. S., Santillo, M. A., Hoagg, J. B., and Bernstein, D. S., "System identification using a retrospective correction filter for adaptive feedback model updating," in *Guidance Navigation and Control Conference*, (Chicago), AIAA, August 2009.

[38] Hornik, K., Stinchcombe, M., and White, H., "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[39] Hovakimyan, N., Yang, B. J., and Calise, A., "An adaptive output feedback control methodology for non-minimum phase systems," *Automatica*, vol. 42, no. 4, pp. 513–522, 2006.

[40] Hovakimyan, N., *Robust Adaptive Control*. Unpublished, 2008.

[41] Hovakimyan, N., Yang, B.-J., and Calise, A. J., "An adaptive output feedback control methodology for non-minimum phase systems," in *Conference on Decision and Control*, (Las Vegas, NV), pp. 949–954, 2002.

[42] Ioannou, P. A. and Kokotovic, P. V., *Adaptive Systems with Reduced Models*. Secaucus, NJ: Springer Verlag, 1983.

[43] Ioannou, P. A. and Sun, J., *Robust Adaptive Control*. Upper Saddle River: Prentice-Hall, 1996.

[44] Ishihara, A., Menahem, B., Nguyen, N., and Stepanyan, V., "Time delay margin estimation for adaptive outer- loop longitudinal aircraft control," in *Infotech@AIAA conference*, (Atlanta), 2010.

[45] Jankt, J. A., Scoggins, S. M., Schultz, S. M., Snyder, W. E., White, S. M., and Scutton, J. C., "Shocking: An approach to stabilize backprop training with greedy adaptive learning rates," *IEEE Neural Networks Proceedings*, vol. 3, no. 7, 1998.

[46] Jategaonkar, R. V., *Flight Vehicle System Identification A Time Domain Approach*, vol. 216 of *Progress in Astronautics and Aeronautics*. Reston: American Institute of Aeronautics and Astronautics, 2006.

[47] Johnson, E., Turbe, M., Wu, A., and Kannan, S., "Flight results of autonomous fixed-wing uav transitions to and from stationary hover," in *Proceedings of the AIAA GNC Conference*, August 2006.

[48] Johnson, E. N., *Limited Authority Adaptive Flight Control*. PhD thesis, Georgia Institute of Technology, Atlanta Ga, 2000.

[49] Johnson, E. and Chowdhary, G., "Guidance and control of an airplane under severe structural damage," in *AIAA Infotech@Aerospace*, 2010. *Invited*.

[50] Johnson, E. and Kannan, S., "Adaptive trajectory control for autonomous helicopters," *Journal of Guidance Control and Dynamics*, vol. 28, pp. 524–538, May 2005.

[51] Johnson, E., Turbe, M., Wu, A., Kannan, S., and Neidhoefer, J., "Flight test results of autonomous fixed-wing uav transitions to and from stationary hover," *AIAA Journal of Guidance Control and Dynamics*, vol. 2, March-April 2008.

[52] Johnson, E. N. and Schrage, D. P., "System integration and operation of a research unmanned aerial vehicle," *AIAA Journal of Aerospace Computing, Information and Communication*, vol. 1, pp. 5–18, Jan 2004.

[53] Kannan, S. K., *Adaptive Control of Systems in Cascade with Saturation*. PhD thesis, Georgia Institute of Technology, Atlanta Ga, 2005.

[54] Kim, N., *Improved Methods in Neural Network Based Adaptive Output Feedback Control, with Applications to Flight Control*. PhD thesis, Georgia Institute of Technology, Atlanta Ga, 2003.

[55] Kim, Y. H. and Lewis, F., *High-Level Feedback Control with Neural Networks*, vol. 21 of *Robotics and Intelligent Systems*. Singapore: World Scientific, 1998.

[56] Krstić, M., Kanellakopoulos, I., and Kokotović, P., *Nonlinear and Adaptive Control Design*. New York: John Wiley and Sons, 1995.

[57] Lavertsky, E. and Wise, K., "Flight control of manned/unmanned military aircraft," in *Proceedings of American Control Conference*, 2005.

[58] Lavretsky, E., "Combined/composite model reference adaptive control," *Automatic Control, IEEE Transactions on*, vol. 54, pp. 2692 –2697, nov. 2009.

[59] Lee, S., *Neural Network based Adaptive Control and its applications to Aerial Vehicles*. PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332, apr 2001.

[60] Leonessa, A., Haddad, W., Hayakawa, T., and Morel, Y., "Adaptive control for nonlinear uncertain systems with actuator amplitude and rate saturation constraints," *International Journal of Adaptive Control and Signal Processing*, vol. 23, pp. 73–96, 2009.

[61] Lewis, F. L., "Nonlinear network structures for feedback control," *Asian Journal of Control*, vol. 1, pp. 205–228, 1999. Special Issue on Neural Networks for Feedback Control.

[62] Liberzon, D., *Handbook of Networked and Embedded Control Systems*, ch. Switched Systems, pp. 559–574. Boston: Birkhauser, 2005.

[63] McConley, M., Piedmonte, M. D., Appelby, B. D., Frazzoli, E., D. M. A., and Feron, E., "Hybrid control for aggressive maneuvering of autonomous aerial vehicles," in *19th Digital Avionics System Conference*, 2000.

[64] Mettler, B., *Modeling Identification and Characteristics of Miniature Rotorcrafts*. USA: Kluwer Academic Publishers, 2003.

[65] Micchelli, C. A., "Interpolation of scattered data: distance matrices and conditionally positive definite functions," *Construct. Approx.*, vol. 2, pp. 11 –22, dec. 1986.

[66] Monahemi, M. M. and Krstic, M., "Control of wingrock motion using adaptive feedback linearization," *Journal of Guidance Control and Dynamics*, vol. 19, pp. 905–912, August 1996.

[67] Morelli, E. A., "Real time parameter estimation in the frequency domain," *Journal of Guidance Control and Dynamics*, vol. 23, no. 5, pp. 812–818, 2000.

[68] Muhammad, A. and Jadbabaie, A., "Decentralized computation of homology groups in networks by gossip," in *American Control Conference*, 2007.

[69] Narendra, K. and Annaswamy, A., "A new adaptive law for robust adaptation without persistent excitation," *IEEE Transactions on Automatic Control*, vol. 32, pp. 134–145, February 1987.

[70] Narendra, K. S. and Annaswamy, A. M., *Stable Adaptive Systems*. Englewood Cliffs: Prentice-Hall, 1989.

[71] Nguyen, N., "Asymptotic linearity of optimal control modification adaptive law with analytical stability margins," in *Infotech@AIAA conference*, (Atlanta, GA), 2010.

[72] Nguyen, N., Krishnakumar, K., Kaneshige, J., and Nespeca, P., "Dynamics and adaptive control for stability recovery of damaged asymmetric aircraft," in *AIAA Guidance Navigation and Control Conference*, (Keystone, CO), 2006.

[73] Ochiai, K., Toda, N., and Usui, S., "Kick-out learning algorithm to reduce the oscillation of weights," *Elsevier Neural Networks*, vol. 7, no. 5, 1994.

[74] of the Secretary of Defense, O., "Unmanned aircraft systems roadmap 2005-2030," tech. rep., Department of Defense, August 2005.

[75] Olfati-Saber, R., Fax, J., and Murray, R., "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, pp. 215 –233, jan. 2007.

[76] Park, J. and Sandberg, I., "Universal approximation using radial-basis-function networks," *Neural Computatations*, vol. 3, pp. 246–257, 1991.

[77] Patiño, H., Carelli, R., and Kuchen, B., "Neural networks for advanced control of robot manipulators," *IEEE Transactions on Neural Networks*, vol. 13, pp. 343–354, Mar 2002.

[78] Ponzyak, A. S., Sanchez, E. N., and Yu, W., *Differential Neural Networks for Robust Nonlinear Control, Identification, State Estimation, and Trajectory Tracking*. Singapore: World Scientific, 2001.

[79] Psichogios, D. C. and Ungar, L. H., "Direct and indirect model based control using artificial neural networks," *Industrial and Engineering Chemistry Research*, vol. 30, no. 12, p. 25642573, 1991.

[80] Roberts, J. M., Corke, P. I., and Buskey, G., "Low-cost flight control system for a small autonomous helicopter," in *IEEE Intl Conf. on Robotics and Automation*, 02.

[81] Rumelhart, D. E., E., H. G., and Williams, R. J., "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986.

[82] Rysdyk, R. T. and Calise, A. J., "Adaptive model inversion flight control for tiltrotor aircraft," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 22, no. 3, pp. 402–407, 1999.

[83] Saad, A. A., *SIMULATION AND ANALYSIS OF WING ROCK PHYSICS FOR A GENERIC FIGHTER MODEL WITH THREE DEGREES-OF-FREEDOM*. PhD thesis, Air Force Institute of Technology, Air University, Wright-Patterson Air Force Base, Dayton, Ohio, 2000.

[84] Santillo, M. A. and Bernstein, D. S., "Adaptive control based on retrospective cost optimization," *AIAA Journal of Guidance Control and Dynamics*, vol. 33, March-April 2010.

[85] Santillo, M. A., D'Amato, A. M., and Bernstein, D. S., "System identification using a retrospective correction filter for adaptive feedback model updating," in *American Control Conference*, (St. Louis), June 2009.

[86] Sastry, S. and Bodson, M., *Adaptive Control: Stability, Convergence, and Robustness*. Upper Saddle River: Prentice-Hall, 1989.

[87] Singh, S. N., Yim, W., and Wells, W. R., "Direct adaptive control of wing rock motion of slender delta wings," *Journal of Guidance Control and Dynamics*, vol. 18, pp. 25–30, Feb. 1995.

[88] Slotine, J.-J. E. and Li, W., "Composite adaptive control of robot manipulators," *Automatica*, vol. 25, no. 4, pp. 509–519, 1989.

[89] Standard, A. D., "Handling qualities requirements for military rotor-craft, ads-33e," tech. rep., United States Army Aviation and Missile Command, Redstone Arsenal, Alabama, march 2000.

[90] Steinberg, M., "Historical overview of research in reconfigurable flight control," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 219, no. 4, pp. 263–275, 2005.

[91] Strang, G., *Linear Algebra and its Applications*. Brooks: Thomson Learning, 1988.

[92] Suykens, J. A., Vandewalle, J. P., and Moor, B. L. D., *Artificial Neural Networks for Modelling and Control of Non-Linear Systems*. Norwell: Kluwer, 1996.

[93] Tao, G., *Adaptive Control Design and Analysis*. New York: Wiley, 2003.

[94] Volyanskyy, K. and Calise, A., "An error minimization method in adaptive control," in *Proceedings of AIAA Guidance Navigation and Control conference*, 2006.

[95] Volyanskyy, K. Y., *ADAPTIVE AND NEUROADAPTIVE CONTROL FOR NONNEGATIVE AND COMPARTMENTAL DYNAMICAL SYSTEMS*. Ph.d., Georgia Institute of Technology, Atlanta, March 2010.

[96] Volyanskyy, K. Y., Haddad, W. M., and Calise, A. J., "A new neuroadaptive control architecture for nonlinear uncertain dynamical systems: Beyond σ and e-modifications," *IEEE Transactions on Neural Networks*, vol. 20, pp. 1707–1723, Nov 2009.

[97] Xu, J.-X., Jia, Q.-W., and Lee, T. H., "On the design of nonlinear adaptive variable structure derivative estimator," *IEEE Transactions on Automatic Control*, vol. 45, pp. 1028–1033, may 2000.

[98] YU, H. and LLOYD, S., "Combined direct and indirect adaptive control of constrained robots," *International Journal of Control*, vol. 68, no. 5, pp. 955–970, 1997.

[99] Yucelen, T. and Calise, A., "Kalman filter modification in adaptive control," *Journal of Guidance, Control, and Dynamics*, vol. 33, pp. 426–439, march-april 2010.

[100] Zhou, K., Doyle, J. C., and Glover, K., *Robust and Optimal Control*. Upper Saddle River, NJ: Prentice Hall, 1996.

VITA

Girish received a Bachelor of Aerospace Engineering degree with first class honors from the Royal Melbourne Institute of Technology (RMIT), Melbourne, Australia in 2003. He then worked as a research engineer with the German Aerospace Center (DLR) at the Institute for Flight Systems Technology in Braunschweig Germany from 2004 to 2006. In Fall 2006, Girish joined the school of Aerospace Engineering at the Georgia Institute of Technology in Atlanta, GA. At Georgia Tech, he has worked with Professor Eric N. Johnson in Aerospace Guidance, Navigation, and Control as well as Autonomous Systems Technology. Girish received a Master of Science degree in Aerospace Engineering from Georgia Tech in 2008.