Concurrent Learning for Improved Convergence in Adaptive Flight Control

Girish Chowdhary[*], Eric N. Johnson[†]

Georgia Institute of Technology, Atlanta, GA, 30318

This paper presents two adaptive control laws that have improved parameter convergence properties. These adaptive control laws are derived through the framework of Model Reference Adaptive Control and are applicable to plants with structured or unstructured modeling uncertainty. The presented adaptive control laws use both recorded and current data concurrently to improve parameter convergence and are shown to have a stability proof. The first method, termed as concurrent learning adaptive control, increases the rank of standard gradient descent based adaptive laws and guarantees that the adaptive weights approach the ideal weights if the recorded data meets a verifiable condition on linear independence. The second method introduces a least squares based modification term that drives the adaptive weights to an estimate of the ideal weights computed online using recorded data. This modification term guarantees the asymptotic stability of the tracking error and convergence of adaptive weights if a condition on linear independence of the recorded data is met. Expected improvement in parameter convergence and tracking error is demonstrated through adaptive autopilot design for a fixed wing unmanned aircraft and through simulation studies of control of wing rock dynamics.

I. Introduction

Adaptive control has been widely studied for aerospace applications. Many active research direction exist. For example Calise,[3] Johnson,[22–24,26] Kannan[28,29] and others have developed Model Reference Adaptive Controllers (MRAC) for both fixed wing and rotary wing Unmanned Aerial Systems (UAS). Cao, Yang, Hovaykiman, and other have developed the L1 adaptive control method.[4,18] Lavertsky,[33] Nguyen,[40] and others have extended direct adaptive control methods to fault tolerant control and developed techniques in composite/hybrid adaptation. The increasing interest in adaptive control stems from its ability to handle changes in system dynamics, provable robustness to uncertainty, and the relatively direct extension to fault tolerant control.

Many adaptive control approaches employ an adaptive element which attempts to capture the plant uncertainty in a parameterized model with the parameters tuned online using some kind of adaptive control law. These parameters are also termed as adaptive weights. The convergence of adaptive weights to ideal values which guarantee accurate parameterization of the plant uncertainty over the entire operating domain is a problem of considerable interest in adaptive control. If the online adaptive control laws are indeed able to drive the adaptive weights to their globally optimal values then the tracking performance of the adaptive control significantly increases. Furthermore, since the globally optimal values ensure accurate uncertainty parameterization over the entire operating domain, performance of the adaptive controller improves over the entire operating domain. Most adaptive control methods are designed using a Lyapunov based approach and can be thought of as attempting to minimize an instantaneous quadratic cost function by using a gradient descent type method.[34,45] Gradient descent type methods however, are susceptible to local adaptation, and weight drift.[38] That is, gradient descent type methods tend to drive the weights to values that minimize the instantaneous cost by approximating the plant uncertainty at the given instant in the state space. Persistency of excitation in the system signals is often required to guarantee the convergence of adaptive weights when using these methods. In order to counter weight drift, Ioannou et al.[20] suggested the use of a σ modification

[*]Research Assistant, School of Aerospace Engineering, AIAA member

[†]Lockheed Martin Professor of Avionics Integration, School of Aerospace Engineering, AIAA member

term which adds damping to the adaptive law. Narendra et al.[38] suggested the e modification term which uses the norm of the tracking error to scale the added damping. These methods are primarily designed to ensure boundedness of adaptive weights and do little to ensure the convergence of the weights to ideal values. In order to improve weight convergence, Volyanskyy et al. suggested the Q modification[46] term which uses an integral over a moving window of the system uncertainty to improve weight convergence. Chowdhary and Johnson suggested the Concurrent Learning[10] method for incorporating long term learning in adaptive laws by using past and current data concurrently for adaptation. Long term learning was characterized by improvement in tracking performance over repeated maneuvers. The idea of learning concurrently on recorded and current data is explored further in this paper.

In this paper we present two adaptive control laws that improve the convergence of adaptive weights in plants with significant modeling uncertainty in the framework of Model Reference Adaptive Control. The firs law extends the concurrent learning method[10,11] to the case where plant uncertainty is captured through a linearly parameterized Radial Basis Function (RBF) Neural Network (NN). Concurrent learning constrains the weight update based on simultaneous gradient descent over a set of stored data point to the null space of the weight update based on current data. This augments the rank of the adaptive law while ensuring that it stays responsive to dynamic changes in the environment. We show that the plant states remain uniformly ultimately bounded when using the concurrent learning law in the framework of MRAC. Furthermore, a useful property of this adaptive law is that if the recorded data contain as many linearly independent elements as the number of RBFs employed, then the weights of the RBF NN are guaranteed to be bounded within a compact neighborhood of the ideal weights without requiring the states to be persistently exciting.

The second control law uses a novel modification term to traditional adaptive control laws to ensure that the adaptive weights converge smoothly to a least squares based optimal estimate of the ideal weights. The least squares based estimate is computed online using stored data employing the efficient Fourier Transform Regression method.[36] We show that this adaptive law is guaranteed to find optimal weights if the least squares problem can be solved online. The least squares problem can be solved online if the stored data points are linearly independent.[2] Leveraging this fact, we show that if the recorded data meet a condition on linear independence then the weights of the adaptive element are guaranteed to converge to their ideal values without requiring the states to be persistently exciting.

One common point between both methods presented is that they ensure that the convergence properties of adaptive controllers are improved subject to a verifiable condition on linear independence of the recorded data. This condition is found to be less restrictive than the requirement of persistence of excitation in the reference signal normally required for convergence of weights in MRAC schemes.[45]

We begin with a brief introduction of MRAC scheme. Section III discusses the concurrent learning adaptive law for improved weight convergence. Section V discusses the least squares modification based control law. Flight tests for validating concurrent learning adaptive control on the GTMAX UAS are currently in progress, and the results will be included in Section VI in the final manuscript. Section VII demonstrates the application of least squares modification based adaptive control to the design of adaptive autopilot for control of wing rock dynamics. Convergence of both tracking error and adaptive weights is observed.

II. Model Reference Adaptive Control

This section discusses the formulation of Model Reference Adaptive Control using approximate model inversion.[24,29,30,37] Let $D_x \in \Re^n$ be compact, and Let $x(t) \in D_x$ be the known state vector, let $\delta \in \Re^k$ denote the control input, and consider the following system:

$$\dot{x} = f(x(t), \delta(t)), \tag{1}$$

where the function f is assumed to be continuously differentiable in $x \in D_x$, and control input δ is assumed to be bounded and piecewise continuous. The conditions for the existence and the uniqueness of the solution to 1 are assumed to be met.

Since the exact model 1 is usually not available or not invertible, we introduce an approximate inversion model $\hat{f}(x, \delta)$ following the approach of Approximate Model Reference Adaptive Control which can be inverted to determine the control input δ:

$$\delta = \hat{f}^{-1}(x, \nu). \tag{2}$$

Where ν is the pseudo control input, which represents the desired model output $\dot{x}$ and is expected to be approximately achieved by δ. Hence, the pseudo control input is the output of the approximate inversion model:

$$\nu = \hat{f}(x, \delta). \tag{3}$$

This approximation results in a model error of the form:

$$\dot{x} = \nu + \Delta(x, \delta) \tag{4}$$

where the model error $\Delta : \Re^{n+k} \to \Re^n$ is given by:

$$\Delta(x, \delta) = f(x, \delta) - \hat{f}(x, \delta). \tag{5}$$

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

$$\dot{x}_{rm} = f_{rm}(x_{rm}, r(t)), \tag{6}$$

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 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 solution to 6 are satisfied. A tracking control law consisting of a linear feedback part $u_{pd} = Kx$, a linear feedforward part $u_{crm} = \dot{x}_{rm}$, and an adaptive part $u_{ad}(x)$ is proposed to have the following form:

$$u = u_{crm} + u_{pd} - u_{ad}. \tag{7}$$

Define the tracking error e as $e(t) = x_{rm}(t) - x(t)$, then, letting $A = -K$ the tracking error dynamics are found to be:[24]

$$\dot{e} = Ae + [u_{ad}(x, \delta) - \Delta(x, \delta)]. \tag{8}$$

The baseline full state feedback controller $u_{pd} = Kx$ is assumed to be designed 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:

$$A^T P + PA + Q = 0. \tag{9}$$

Letting $\bar{x} = [x, \delta] \in \Re^{n+k}$The following two cases for characterizing the uncertainty $\Delta(x)$ are considered:

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 1. *The uncertainty $\Delta(\bar{x})$ can be linearly parameterized, that is, there exist a vector of constants $W^* \in \Re^{m \times n}$ and a vector of continuously differentiable functions $\Phi(\bar{x}) = [\phi_1(\bar{x}), \phi_2(\bar{x}), ..., \phi_m(\bar{x})]^T$ such that*

$$\Delta(\bar{x}) = W^{*T} \Phi(\bar{x}). \tag{10}$$

In this case letting W denote the estimate W^* the adaptive law can be written as

$$u_{ad}(\bar{x}) = W^T \Phi(\bar{x}). \tag{11}$$

Case II: Unstructured Uncertainty: If it is only known that the uncertainty $\Delta(\bar{x})$ is continuous and defined over a compact domain $D \subset \Re^{n+k}$, a Radial Basis Function (RBF) Neural Network (NN) can be used as the adaptive element:

$$u_{ad}(\bar{x}) = W^T \sigma(\bar{x}). \tag{12}$$

where $W \in \Re^{n \times l}$ and $\sigma(\bar{x}) = [1, \sigma_2(\bar{x}), \sigma_3(\bar{x}), ..., \sigma_l(\bar{x})]^T$ is a vector of known radial basis functions. 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:

$$\sigma_i(x) = e^{-\|\bar{x} - c_i\|^2 / \mu_i} \tag{13}$$

Appealing to the universal approximation property of RBF NN (see reference 41 or 44) we have that given a fixed number of radial basis functions l there exists ideal weights $W^* \in \Re^{n \times l}$ and a real number $\tilde{\epsilon}(\bar{x})$ such that the following approximation holds for all $x \in D$ where D is compact:

$$\Delta(x) = W^{*T} \sigma(\bar{x}) + \tilde{\epsilon}(\bar{x}), \tag{14}$$

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

Figure 1 depicts the control architecture for MRAC control discussed in this section.

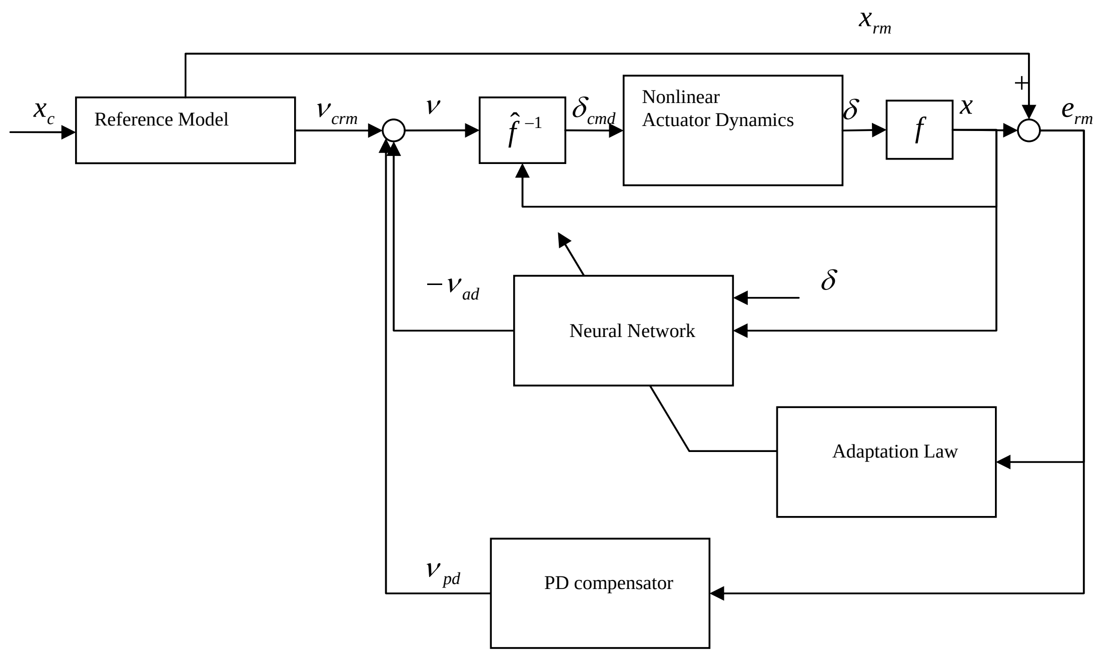


Figure 1. Neural Network Adaptive Control using Approximate Model Inversion

A. Baseline Adaptive Law

For the case of structured uncertainty it is well known that in the presence of persistently exciting input the following adaptive law

$$\dot{\hat{W}} = -\Phi(\bar{x})e^T PB\Gamma_W \tag{15}$$

where Γ_W is a positive definite matrix of appropriate dimensions results in ultimate boundedness of the weights and $e(t) \to 0$.[20,38,45] Equation 15 will be referred to as the baseline adaptive law. It can be shown that the baseline adaptive law can be arrived at by minimizing $e^T(t)e(t)$ using a gradient descent methodology,[17,45] hence this adaptive law will also be referred to as gradient based. Furthermore, replacing $\Phi(\bar{x})$ with $\sigma(\bar{x})$ in equation 15 results in the baseline gradient based adaptive law for the case of unstructured uncertainty (case 2). For this case, the baseline adaptive law guarantees uniform ultimate boundedness of tracking error e.[30,31,34] This adaptive law however, does not guarantee the boundedness of adaptive weights in both cases.

In order to ensure that adaptive weights remain bounded, Ioannou et al.[20] suggested the use of a σ modification term which adds damping to the adaptive law. Narendra et al.[38] suggested the e modification term which uses the norm of the tracking error to scale the added damping. These modifications have since been commonly used in adaptive control to ensure that the weights do not drift.[19,31,44,45] However, these modifications do little to ensure convergence of adaptive weights, in fact, these modificiations are often designed to bound the weights around a neighborhood of the origin. This can be counterproductive in situations where the adaptive element must estimate a steady-state trim value that is far away from origin. Finally, these modifications do little to ensure weight convergence. If weight convergence occurs, then the linear, exponentially stable term in the tracking error dynamics of equation 8 dominates, and exponential tracking error convergence can be guaranteed. Furthermore, if weight convergence is achieved, then the performance of the system improves over the entire domain of the state space, thus incorporating long term learning and global error parameterization. The convergence of weights $\hat{W}$ to their ideal values W^* can only be guaranteed with the gradient based baseline law of equation 15 if the signal $\Phi(\bar{x}(t))$ is Persistently Exciting (PE). The persistence of excitation of a bounded vector signal has been defined by Tao as follows:[45]

Definition 1. *A bounded signal $\Phi(t)$ is defined to be persistently exciting if for all $t > t_0$ there exists $T > 0$ and $\gamma > 0$ such that*

$$\int_t^{t+T} \Phi(x(\tau))\Phi^T(x(\tau))d\tau \geq I\gamma. \tag{16}$$

Boyd and Sastry have shown that in the framework of MRAC, the persistency of excitation of $\Phi(x(t))$ can be guaranteed through the persistency of excitation of the tracking reference signal $r(t)$.[1, 19, 45] Consequently, persistency of excitation in the tracking signal is required to guarantee convergence of parameters when using the control law of equation 15. Constant reference signals are not persistently exciting, nor are exponentially decaying reference signals. Hence, in order to ensure parameter convergence, the tracking signal must enforce persistent excitation, requiring significant control effort. Furthermore, enforcing persistency of excitation for flight vehicle control can result in loss of ride comfort. Finally, it is hard to determine whether a signal is persistently exciting online.

III. Concurrent Learning Adaptive Law when the Structure of the Uncertainty is Known

The first adaptive control law for improved parameter convergence we present is based on the novel concept of concurrent gradient descent. This adaptive law performs gradient descent on cost formulated using carefully selected and stored data points concurrently with gradient descent on instantaneous cost formed with current data. We let $\hat{W}(t)$ denote a gradient based adaptive law that attempts to minimize the instantaneous tracking error cost $e(t)^T e(t)$. Let P denote the number of stored data points and i denote the index of a stored data point, formulate the cost $v_b = \sum_{i=0}^{p}(\hat{W}^T\Phi(x_i) - \Delta(x_i))^T(\hat{W}^T\Phi(x_i) - \Delta(x_i))$, let $\hat{W}_b(t)$ denote a gradient based adaptive law that directs the weights in the direction of maximum descent of v_b. Then the concurrent learning law has the following form:

$$\dot{\hat{W}}(t) = \dot{\hat{W}}_t(t) + W_c(t)\dot{\hat{W}}_b(t). \tag{17}$$

In the above equation the term $W_c(t)$ is used to prioritize learning over current and past data. If $W_c(t) = I$, where I denotes the identity matrix of appropriate dimensions, then the training on recorded data and training on current data receive the same priority. If on the other hand, if $W_c(t)$ is an orthogonal projection matrix that projects $\dot{W}_b$ into the nullspace of $\dot{W}_t(t)$ then the training on current data receives more priority over the training on recorded data. Such prioritization is useful to ensure that the concurrent use of recorded data does not affect the direction of weight updates designed to minimize the instantaneous tracking error. This property can be particularly useful if for some reason the recorded data has become irrelevant or corrupted. The projection matrix that allows such prioritization is given by:

$$W_c(t) = I - \hat{W}_t(t)(\hat{W}_t^T(t)\hat{W}_t(t))^+\hat{W}_t^T(t). \tag{18}$$

Since the weight update based on the instantaneous data is always at most rank-1, concurrent learning can occur in an n-1 dimensional subspace, where n is the maximum possible rank of the appropriate weight matrix. Due to the enforced separability, this approach to concurrent training on instantaneous as well as stored data is not susceptible to sluggish adaptation performance as forgetting factor[39] based NN adaptation laws are. Furthermore, as in reference 42, the concurrent learning law does not require a pre-adapted bank of NN for improving global model error parameterizations properties. Finally, the concurrent learning law overcomes the rank-one limitation which allows the learning law to search for the ideal NN weights in the complete parameter space.

By utilizing recorded data concurrently with current data, concurrent leading leverages the intuitive concept that if the recorded data is sufficiently rich, then it should be possible to guarantee exponential tracking error and parameter error convergence without requiring the system states to be persistently exciting. A relevant question that arises is: How does one know if the recorded data is sufficiently rich? In reference 8 we have shown that a sufficient condition on richness of the recorded data is that it contain as many linearly independent elements as the dimension of the chosen basis of the uncertainty (which would be the dimension of $\Phi(\bar{x})$ for the case of structured uncertainty, and the dimension of $\sigma(\bar{x})$ for the case of unstructured uncertainty). This leads to the following theorem for the case of structured uncertainty

Theorem 1. *Consider the system in equation 1, the control law of equation 7, the case of structured uncertainty (case 1). For the j^{th} recorded data point let $\epsilon_j = \nu_{ad}(\bar{x}_j) - \Delta(\bar{x}_j)$, furthermore let p be the number of recorded data points $\Phi(\bar{x}_j)$ in the matrix $Z = [\Phi(\bar{x}_1), ..., \Phi(\bar{x}_p)]$, such that $rank(Z) = m$, and consider the following weight update law:*

$$\dot{W}(t) = -\Gamma_W \Phi(\bar{x}(t)) e^T(t) PB - \sum_{j=1}^{p} \Gamma_W \Phi(\bar{x}_j) \epsilon_j^T(t). \tag{19}$$

Then the zero solution $e(t) \equiv 0$ of tracking error dynamics of equation 8 is globally exponentially stable and $W(t) \to W^*$ exponentially.

Proof. A proof can be found in references 5,13. An equivalent theorem for a different class of plants is proved in 8.

□

Remark 1. *In the above theorem it is shown that a verifiable condition on the linear independence of the recorded data is sufficient to guarantee that the zero solutions of the tracking error and the parameter error are globally exponentially stable. It is important to note that the rank-condition on the recorded data that is sufficient to guarantee exponential stability is significantly different than a condition on PE states. Firstly, this condition applies only to the recorded data, which is a small subset of all past states, whereas, the condition on PE states applies to all past and future states. Secondly, since the rank of a matrix can be easily determined online, it is possible to verify whether this condition is met online, whereas it is impossible to determine whether a signal will be PE without knowing its future behavior. Hence, the rank-condition required to guarantee convergence when recorded data is concurrently used for adaptation with instantaneous data is less restrictive than a condition on PE states.*

Remark 2. *In references 5, 13 it is shown that if training on current data is prioritized over training on past data by using a projection operator such as that in equation 18 then asymptotic tracking error and parameter error convergence can be guaranteed.*

The data points required for concurrent learning can be selected and stored in a number of possible ways. For a given $\epsilon_{\bar{x}}$ a simple way to choose points that are sufficiently different than the last point stored is to enforce the following criterion:

$$\frac{\|\bar{x} - \bar{x}_p\|^2}{\|\bar{x}\|} \geq \epsilon_{\bar{x}}. \tag{20}$$

Here the subscript p denotes the index of the last data point stored. The above method ascertains that only those data points are selected for storage that are sufficiently different from the last data point stored. The model error $\Delta(\bar{x}_i)$ of equation 5 can be estimated by denoting the state estimate of x by $\hat{x}$ and rewriting equation 8 as:

$$\hat{\Delta}(\bar{x}_i) = \dot{\hat{x}}_i - \nu_i. \tag{21}$$

In this way, the problem of estimation of $\Delta(\bar{x}_i)$ can be reduced to that of estimating $\dot{\hat{x}}_i$. In cases where an explicit measurement for $\dot{x}$ is not available, for a given sample k, $\dot{x}$ can be estimated using an implementation of a fixed point smoother,[15] the details of which are given in Appendix A.

IV. Neuro-Adaptive control with Concurrent Learning for Cases when the Structure of the Uncertainty is Unknown

When the structure of the uncertainty is unknown (case II in section II), a NN can be used as the adaptive element. In this case, the following theorem shows that the tracking error and NN weights are uniformly ultimately bounded.

We being with the following assumptions:

Assumption 2. *The NN approximation $\Delta(x) = u_{ad}(x) + \epsilon$ holds in a compact domain D which is sufficiently large. Hence, $\|[b_w, x^T]^T\| \leq b_w + x_c$ for some positive constant x_c, that is, the inputs to the NN remain bounded.*

In the following, let $j \in \aleph$ denote the index of a stored data point $\bar{x}_j$, define $\epsilon_j(t) = W^T(t)\sigma(\bar{x}_j) - \Delta(\bar{x}_j)$, where $\hat{\Delta}(x) = \dot{x}_j - \nu_j$. Furthermore, define $\tilde{W}(t) = \hat{W}(t) - W^*$ as the difference between the approximated NN weights and the ideal NN weights.

Theorem 2. *Consider the system in equation 1, the control law of equation 7, let $\bar{x}(0) \in D$ where D is compact, and the case of unstructured uncertainty (Case II). For the j^{th} recorded data point let $\epsilon_j(t) = W^T(t)\sigma(\bar{x}_j) - \Delta(\bar{x}_j)$, furthermore let p be the number of recorded data points $\sigma(\bar{x}_j)$ in the matrix $Z = [\sigma(\bar{x}_1), ..., \sigma(\bar{x}_p)]$, such that $rank(Z) = l$, and consider the following weight update law:*

$$\dot{W} = -\Gamma_W \sigma(z) e^T P - \Gamma_W W_c(t) \sum_{j=1}^{p} \sigma(z_j)\epsilon_j^T. \tag{22}$$

Then the following adaptive weight update law renders the reference model tracking error e and the SHL NN weight errors $\tilde{W}$ uniformly ultimately bounded.

Proof. Consider the following positive definite and radially unbounded Lyapunov candidate

$$V(e, \tilde{W}) = \frac{1}{2} e^T P e + tr(\frac{1}{2}\tilde{W}^T {\Gamma_W}^{-1} \tilde{W}). \tag{23}$$

where tr denotes the trace operator.

Since $\epsilon_j(t) = \tilde{W}^T(t)\sigma(\bar{x})$ we have that $\dot{\tilde{W}}(t) = -\Gamma_W W_c(t) \sum_{j=1}^{p} \sigma(x_j)\sigma^T(x_j)\tilde{W}(t) - \Gamma_W \sigma(x(t)) e^T(t) P$

Differentiating 23 along the trajectory of 8, the Lyapunov equation (equation 9), we get

$$\begin{aligned} \dot{V}(e, \tilde{W}) = & -\frac{1}{2} e^T Q e + e^T P(u_{ad} - \Delta) \\ & + tr(\tilde{W}^T(-W_c(t) \sum_{j=1}^{p} \sigma(x_j)\sigma^T(x_j)\tilde{W} - \sigma(x) e^T P B)) \end{aligned} \tag{24}$$

Canceling like terms and simplifying we have

$$\dot{V}(e, \tilde{W}) = -\frac{1}{2} e^T Q e - tr(\tilde{W}^T(\sum_{j=1}^{p} \sigma(x_j)\sigma^T(x_j))\tilde{W}) + e^T P \tilde{\epsilon}(x). \tag{25}$$

Let $\Omega = \sum_{j=1}^{p} \sigma(x_j)\sigma^T(x_j)$, since $rank(Z) = l$, $\Omega > 0$, now using equation 14 we have

$$\dot{V}(e, \tilde{W}) \leq -\frac{1}{2}\lambda_{\min}(Q) e^T e - \lambda_{\min}(\Omega)\|\tilde{W}^T \tilde{W}\| + e^T P B \bar{\epsilon}, \tag{26}$$

where $\bar{\epsilon}$ denotes the supremum over all $\tilde{\epsilon}(x)$ for all $x \in D$. Hence $\dot{V}(e, \tilde{W}) \leq 0$ outside a compact neighborhood of the origin ($e = 0, \tilde{W} = 0$) for sufficiently large $\lambda_{\min}(Q)$ and $\lambda_{\min}(\Omega)$. Hence the tracking error $e(t)$ and the parameter error $\tilde{W}$ are uniformly ultimately bounded, furthermore, since $x_{rm}(t)$ is bounded for bounded $r(t)$, $x(t)$ remains bounded. Since $V(e, \tilde{W})$ is radially unbounded the result holds for all $x(0) \in D$. □

Remark 3. *In the above theorem it is shown that a verifiable condition on the linear independence of the recorded data is sufficient to guarantee that the tracking error and the parameter error are uniformly ultimately bounded. Furthermore, by using equation 26 it can be shown that the adaptive weights W stay bounded around a compact neighborhood of the ideal weights if $rank(Z) = l$*[5, 8] *when the adaptive law of theorem is used. Whereas, the baseline adaptive law augmented with a traditional implementation of e-mod or σ-mod can only guarantee that the weights stay bounded within a neighborhood of the origin (or a pre-defined constant).*

Remark 4. *Removing or adding recorded data points does not affect the Lyapunov candidate. Therefore, it can be shown that the result remains valid if data points are removed or added arbitrarily as long as $rank(Z) = l$ and assuming that this is not done infinity fast.*[16]

Remark 5. *In situations where a pre-recorded history stack (the matrix Z) is not available, data points must be recorded in real time. In order to guarantee the boundedness of weights until $rank(Z) = l$ an e-mod term can be used.*

V. Least Squares Modification to Adaptive Control

Least Squares based methods are commonly used in system identification,[21] particularly for linear regression. Nguyen showed that direct NN based adaptive control algorithm performance can be improved with recursive least squares parameter estimation algorithm, Nguyen also demonstrated that the RLS algorithm is stable in the sense that the weights and tracking error remains bounded[40] . We further explore the incorporation of least squares based estimators in the design of adaptive control laws. The novel contribution of our work is that the presented adaptive law is shown to guarantee asymptotic convergence of the parameters and tracking error. This is achieved by augmenting the baseline gradient based adaptive law of equation 15 with a modification term that drives the adaptive weights to a least squares estimate of the ideal weights formed using stored data. The resulting control law is a combined direct and indirect adaptation law that uses stored and current data concurrently for improving adaptation performance. In the case where the structure of the uncertainty is known (Case 1 in section II) this method guarantees asymptotic convergence of both the adaptive weights and the tracking error. In the case where the uncertainty is linearly parameterized and unstructured (Case 2), this method guarantees the convergence of adaptive weights to the best linear approximation in the least squares sense. In the following, stability proofs will be developed for case 1 and subsequently generalized for Case 2. Furthermore, for simplicity, we will also assume that the uncertainty is scalar, and it is only a function of the system states $x(t)$. That is $\Delta(x) = W^T\Phi(x(t))$ with $W \in \Re^m$ and $\Phi(x(t)) \in \Re^m$. The generalization to the case of vector uncertainty should be straightforward.

A. Online Least Squares Regression

We begin by describing a method by which least squares Regression can be performed online. Let N denote the number of available state measurements at time t, and θ denote an estimate of the ideal weighs W. 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:

$$\min_W \epsilon^T\epsilon. \tag{27}$$

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

$$X = \begin{bmatrix} \phi_1(x(1)) & \phi_2(x(1)) & ... & \phi_m(x(1)) \\ \phi_1(x(2)) & \phi_2(x(2)) & ... & \phi_m(x(2)) \\ & & & \\ \phi_1(x(N)) & \phi_2(x(N)) & ... & \phi_m(x(N)) \end{bmatrix}. \tag{28}$$

A closed form solution to the least squares problem is given as:[21]

$$\theta = (X^TX)^{-1}X^TY. \tag{29}$$

Equation 29 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.[36] The three main benefits of the FTR approach are: 1. the matrix to be inverted 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:

$$F[x(t)] = \tilde{x}(w) = \int_{-\infty}^{+\infty} x(t)e^{-jwt}dt. \tag{30}$$

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

$$X(w) = \sum_{k=0}^{N-1} x(k)e^{-jwk\Delta t}. \tag{31}$$

Then the Euler approximation for the Fourier transform in equation 30 is given by:

$$\tilde{x}(w) = X(w)\Delta t. \tag{32}$$

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:

$$X_k(w) = X_{k-1}(w) + x(k)e^{-jwk\Delta t}. \tag{33}$$

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 :

$$\tilde{Y}(w) = \tilde{X}(w)\theta + \tilde{\epsilon}. \tag{34}$$

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:

$$\tilde{X}(w) = \begin{bmatrix} \phi_1(x(1)) & \phi_2(x(1)) & \dots & \phi_m(x(1)) \\ \phi_1(x(2)) & \phi_2(x(2)) & \dots & \phi_m(x(2)) \\ & & & \\ \phi_1(x(l)) & \phi_2(x(l)) & \dots & \phi_m(x(l)) \end{bmatrix}. \tag{35}$$

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 5, 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:

$$\Delta_k(w) = B^T[x_k(w)jw - Ax_k(w) - Bu_k(w)]. \tag{36}$$

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

$$\theta = [Re(\tilde{X}^*\tilde{X})]^{-1}Re(\tilde{X}^*\tilde{Y}), \tag{37}$$

where $*$ denotes the complex conjugate transpose. Another benefit of the FTR method is that forgetting factors can be used to discount older data while recursively computing the Fourier Transform.[36]

B. Least Squares Modification to Adaptive Control using Concurrent Learning

We now describe a method by which the least squares estimate of the ideal weights can be incorporated in the adaptive control law. Let $r^T = e^TPB$ where e, P, B are as in section II, Γ_W, Γ_θ are positive definite matrices denoting the learning rate, θ is the solution to the least squares problem (equation 29).

The adaptive law for weight estimates $\hat{W}$ is chosen as:

$$\dot{\hat{W}} = -(\Phi(x)r^T - \Gamma_\theta(\hat{W} - \theta))\Gamma_W. \tag{38}$$

In order to analyze the stability of this adaptive law, we begin with the following assumption on the stored data which is required for guaranteeing asymptotic convergence of adaptive weights.

Assumption 3. *Assume that enough number of state measurements are available such that the matrix $\tilde{X}(w)$ of equation 35 has full column rank.*

Recalling that the matrix $\tilde{X}(w)$ contains Fourier transform of the vector signal $\Phi(x(t))$ we note that assumption 3 requires that the stored data points be sufficiently different. We note that this assumption is considerably weaker than the assumption of 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 15. Furthermore, this assumption is much easier to verify online by simply monitoring the rank of $\tilde{X}(w)$.

In the following, we present Lyapunov based stability analysis for the adaptive law of equation 38.

Theorem 3. *Consider the control law of equation 7, and the weight update law of 38, then all signals in the system 1 are bounded and $x(t)$ tracks x_{rm} asymptotically. Furthermore, if assumption 3 is satisfied, then $\hat{W}$ converges asymptotically to the ideal weights W.*

Proof. Let $\tilde{W} = \hat{W} - W$, tr denote the trace operator, and consider the following Lyapunov candidate function:

$$V(e, \tilde{W}) = \frac{1}{2}e^T Pe + \frac{1}{2}tr(\tilde{W}^T {\Gamma_W}^{-1}\tilde{W}). \tag{39}$$

The time derivative of the Lyapunov candidate along the system trajectory 8 is:

$$\begin{aligned}\dot{V}(e, \tilde{W}) = &-\frac{1}{2}e^T Qe + r^T(\hat{W}^T\Phi(x) - W^T\Phi(x)) \\ &+ tr(\dot{\hat{W}}{\Gamma_W}^{-1}\tilde{W}^T).\end{aligned} \tag{40}$$

Let ϵ be such that $W = \theta + \epsilon$, adding and subtracting $(\hat{W}^T - \theta)^T\Gamma_\theta(\hat{W}^T - \theta)$ to equation 40 and using the definition of ϵ yields,

$$\begin{aligned}\dot{V}(e, \tilde{W}) = &-\frac{1}{2}e^T Qe + r^T(\tilde{W}^T\Phi(x)) + tr(\dot{\hat{W}}{\Gamma_W}^{-1}\tilde{W}^T) \\ &+ \tilde{W}^T\Gamma_\theta(\hat{W} - \theta) - \tilde{W}^T\Gamma_\theta(\hat{W} - \theta).\end{aligned} \tag{41}$$

Rearranging yields:

$$\begin{aligned}\dot{V}(e, \tilde{W}) = &-\frac{1}{2}e^T Qe \\ &+ tr((\dot{\hat{W}}{\Gamma_W}^{-1} + \Phi(x)r^T + \Gamma_\theta(\hat{W} - \theta))\tilde{W}^T) \\ &- \tilde{W}^T\Gamma_\theta(\hat{W} - \theta).\end{aligned} \tag{42}$$

Setting $tr((\dot{\hat{W}}{\Gamma_W}^{-1} + \Phi(x)r^T + \Gamma_\theta(\hat{W} - \theta))\tilde{W}^T) = 0$ yields the adaptive law. Consider the last term in 42,

$$\begin{aligned}\tilde{W}^T\Gamma_\theta(\hat{W} - \theta) = &\ (\hat{W} - W)^T\Gamma_\theta(\hat{W} - \theta) \\ = &\ (\hat{W} - W)^T\Gamma_\theta(\hat{W} - W) \\ &+ (\hat{W} - W)^T\Gamma_\theta\epsilon.\end{aligned} \tag{43}$$

Using 37, the definition of ϵ, and *assumption 3* yields:

$$\epsilon = W - [Re(\tilde{X}^*\tilde{X})]^{-1}Re(\tilde{X}^*\tilde{X})W = 0, \tag{44}$$

letting $\lambda_{min}(Q)$ and $\lambda_{min}(\Gamma_\theta)$ denote the minimum eigenvalues of Q and Γ_θ we have that equation 42 becomes:

$$\dot{V}(e, \tilde{W}) = -\frac{1}{2}\|e\|^2\lambda_{min}(Q) - \|\tilde{W}\|^2\lambda_{min}(\Gamma_\theta). \tag{45}$$

Hence $\dot{V}(e, \tilde{W}) < 0$ which shows that $e \to 0$ and $\hat{W} \to W$, since A_{rm} is Hurwitz, x_{rm} is bounded, therefore it follows that x is bounded. □

Remark 6. *The above proof shows asymptotic stability of tracking error and shows that $\hat{W}$ will approach the ideal weight W. This is subject to 3, which is much weaker condition on $\Phi(x(t))$ than persistency of excitation. Particularly, the signal only needs to be exciting in the past when it was recorded, rather than remain persistently exciting. Furthermore, the proof does not require persistency of excitation of $r(t)$.*

Remark 7. *The above proof can be easily extended to the case where the uncertainty is unstructured (case 2 from section II) by using Radial Basis Function Neural Networks for approximating the uncertainty. For this case, it is not possible to set $\epsilon = 0$ using equation 44 since $Y = {W^*}^T\sigma + \tilde{\epsilon}$ and the following adaptive law will result in uniform ultimate boundedness of all states:*

$$\dot{\hat{W}} = -(\sigma(x)r^T - \Gamma_\theta(\hat{W} - \theta))\Gamma_W. \tag{46}$$

Furthermore, referring to equation 14 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.

Remark 8. *Note that the term* $\Gamma_\theta(\hat{W} - \theta)$ *adds in as a modification term to the baseline adaptive law of equation 15. Since the above analysis is valid for any bounded initial condition and since the baseline adaptive law is known to be stabilizing for equation 1 and 8, it is possible to set* $\theta = 0$ *until sufficient data is collected online to satisfy assumption 3. This will result in a* σ*-modification like term until satisfaction of assumption 3 can be verified online. Hence, assumption 3 can be considered as a requirement to ensure asymptotic convergence of weights.*

Remark 9. *This proof can be modified to accommodate any least squares solution method, for example the standard least squares solution of 29 can be accommodated by replacing equation 44 with the following:*

$$\epsilon = W - (X^T X)^{-1} X^T X W = 0, \tag{47}$$

In this case, assumption 3 requires the matrix X *to have full column rank.*

Remark 10. *Recursive Least Square (RLS) method can also be used to recursively update* θ *as data becomes available by defining a process model* $\dot{\theta} = 0$*, a measurement model* $y = \theta^T \Phi(x)$ *and using Kalman Filter based methods. We have performed this extension in reference 7. Convergence of* θ *to the ideal weights in this case is dependent on the persistency of excitation in* $\phi(x)$*.*[45]

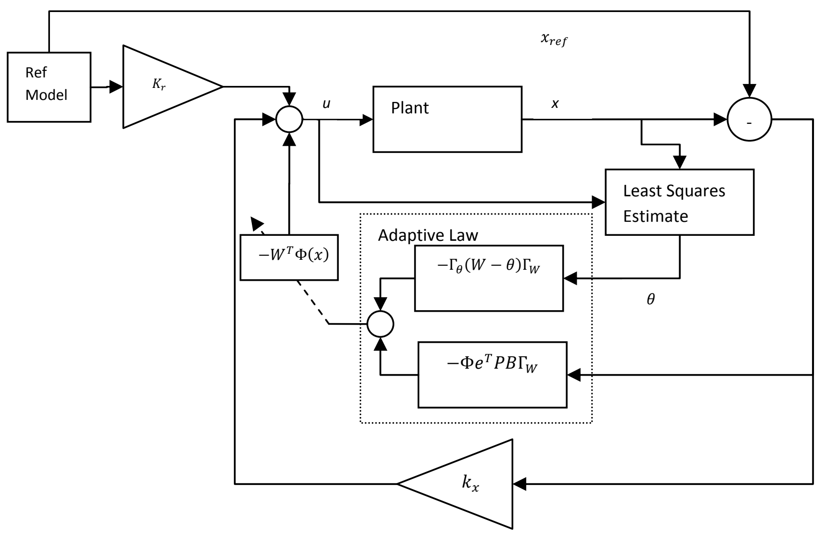


Figure 2. Schematics of adaptive controller with least squares Modification

Figure 2 shows the schematic of the presented adaptive control method with least squares modification.

VI. Flight Test Validation of a Concurrent Learning Adaptive Autopilot for a Fixed-Wing UAS

Concurrent learning adaptive control methods have been validated on various systems in the past. Notably, Chowdhary and Johnson showed improved adaptation performance when using concurrent learning for the control of an inverted pendulum system.[9] Kutay and Chowdhary showed improved performance of a fault tolerant controller implemented on a simulation model of a UCAV.[32] We have previously demonstrated concurrent learning adaptive controllers with Single Hidden Layer Neural Networks as the adaptive elements.[10,11] Improved tracking performance as well as higher separation in weights was observed in those results.

In this paper, we employ Radial Basis Function Neural Networks as the adaptive elements in the design of adaptive autopilots for the fixed wing GT Twinstar UAS. The GT Twinstar (Figure 3) 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.[12] 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.[27] A linear model for the Twinstar in nominal configuration (without damage) has been identified using the FTR method by the authors.[14] A linear model with 25% left wing missing has also been identified.[6]

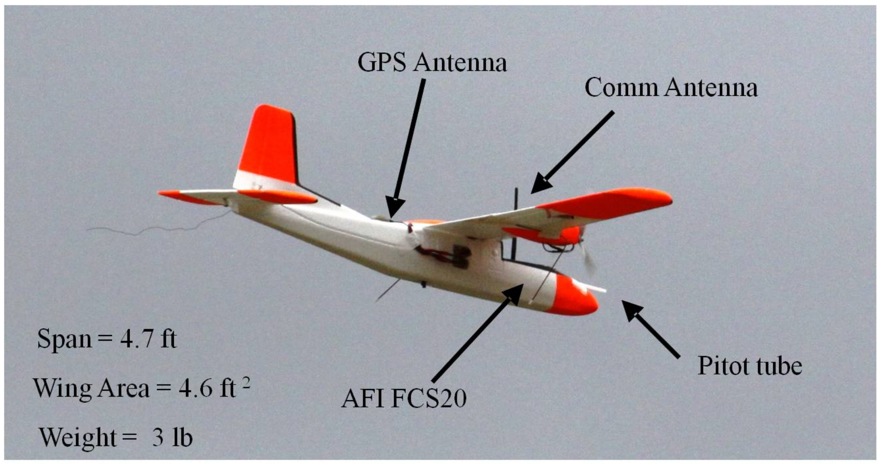


Figure 3. The Georgia Tech Twinstar UAS. The GT Twinstar is a fixed wing foam-built UAS designed for fault tolerant control work.

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.[25] 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 25. The innerloop ensures that the states of the aircraft track these desired quantities using the control architectures described in section IV. 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:

$$\dot{W}(t) = -\Gamma_W \sigma(\bar{x}(t)) e^T(t) P - \kappa \|e(t)\| W(t). \tag{48}$$

In the above equation, $\kappa = 0.1$ denotes the gain of the e-mod term.[38] The concurrent learning adaptive controller uses the learning law of equation 22. A nominal e-mod term with $\kappa = 0.01$ is also used for the concurrent learning adaptive law. The ground tracks of both controllers are compared in figure 4. 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[25] 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 5 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 6(a), while the innerloop tracking error performance of the concurrent learning controller is shown in figure 6(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 7(a), while the actuator input required for the concurrent learning adaptive controller is shown in figure 7(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.

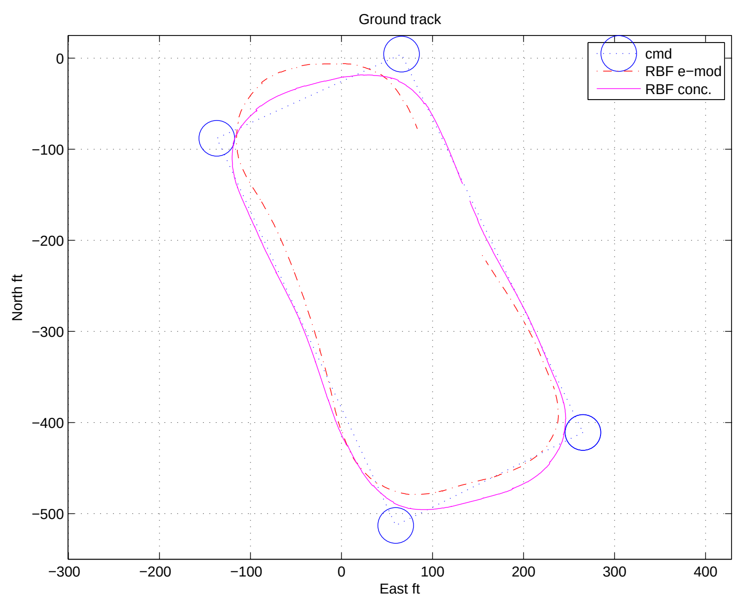


Figure 4. Comparison of ground track for baseline adaptive controller with concurrent learning adaptive controller. Note that the concurrent learning controller has better cross-tracking performance than the baseline adaptive controller

VII. Application of Least Squares Modification Method to Control of Wing-Rock Dynamics

In this section we use the method of Theorem 3 for the control a wing rock dynamics model. We show that the least squares modification is able to demonstrate convergence of weights and adaptive errors for the case where the structure of the uncertainty is known. 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:[35]

$$\dot{\phi} = p \tag{49}$$

$$\dot{p} = \delta_a + \Delta(x). \tag{50}$$

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[43] and,[46] 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 7 are held constant.

Consider the case where the exact form of the uncertainty is known (case 1, in section II). Figure 8(a) shows the performance of the baseline adaptive control law of equation 15 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. It is seen that the performance of the controller in both cases is unsatisfactory. Figure 8(b) shows the phase portrait of the states when the adaptive law with least squares modification of Theorem 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 9(a) shows the evolution of the adaptive control weights when only the baseline adaptive law of equation 15 is used. It is seen that the weights do not converge to the ideal values (W)

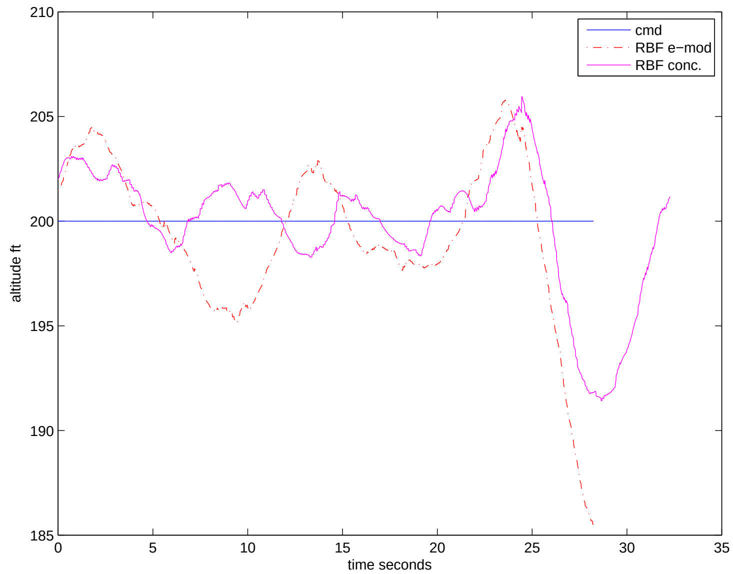


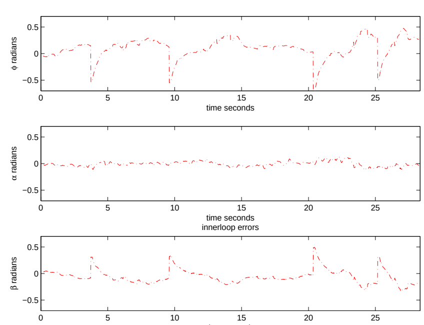
Figure 5. Comparison of altitude tracking for baseline adaptive controller with concurrent learning adaptive controller.

and evolve in an oscillatory manner. In contrast, figure 9(b) shows the convergence of the weights when the least squares modification based adaptive law of Theorem 1 used. Figure 10(a) compares the reference model states with the plant states for the baseline adaptive law, while 10(b) 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.

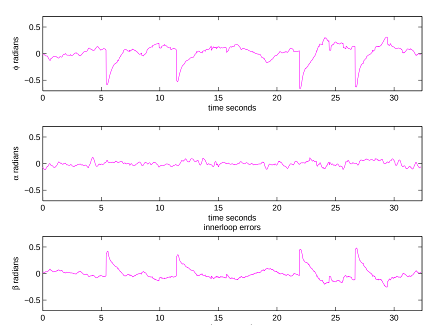
Finally, figure 11 shows that the tracking error converges asymptotically to the origin when least squares modification term is used.

VIII. Conclusion

We presented two methods that use past and current data concurrently to improve adaptive weight convergence. These methods were developed in the framework of Model Reference Adaptive Control and are applicable to adaptive control problems with structured or unstructured plant uncertainty. The first

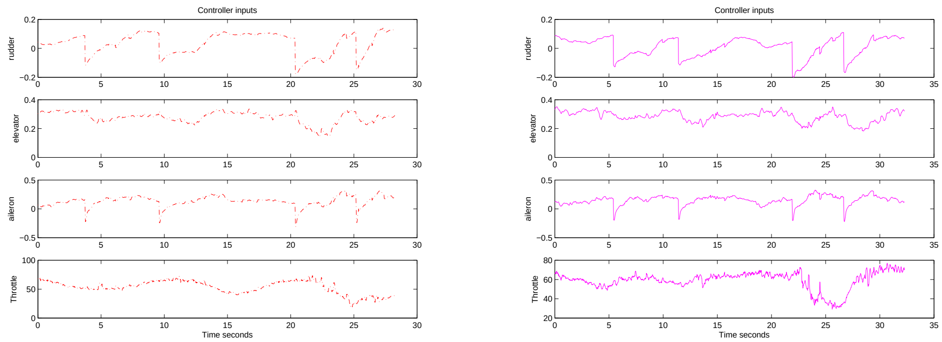


(a) Inner loop tracking errors for baseline adaptive controller



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

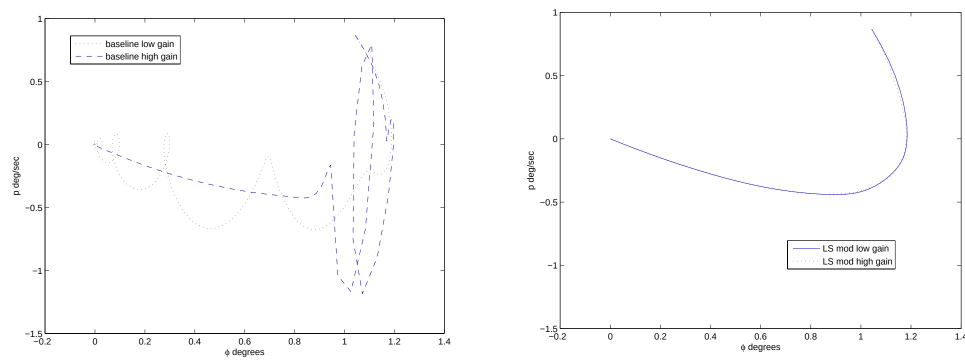
Figure 6. Comparison of inner loop tracking errors. Although the transient performance is similar, the concurrent learning adaptive controller was found to have better trim estimation



(a) Actuator inputs for baseline adaptive controller

(b) Actuator inputs for concurrent learning adaptive controller

Figure 7. 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.



(a) Phase portrait of system states with only baseline adaptive control

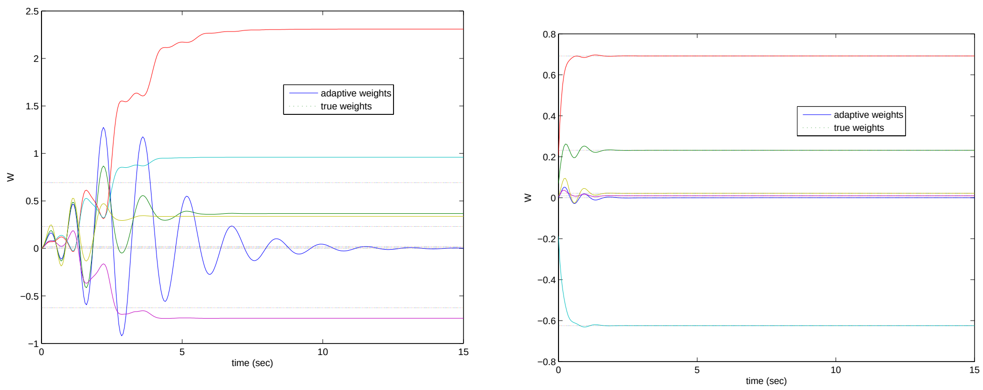
(b) Phase portrait of system states with least squares modification

Figure 8. Performance of adaptive controller with and without least squares modification for control of wing rock dynamics. Note that the control performance with least squares modification is better, and is comparable for both the low and high gain case.

method, termed as concurrent learning adaptive control, uses both current and recorded data concurrently for adaptation. One novel aspect of this method is that the weight updates based on past data can be restricted to the null space of the weight updates based on current data. This increases the rank of the adaptive law, while ensuring that inclusion of past data does not affect the responsiveness of the adaptive law. Using Lyapunov like analysis, we demonstrated that this method ensures that all systems stay uniformly ultimately bounded. Furthermore, we mentioned that if the recorded data contains as many linearly independent elements as the number of radial basis functions, then the weights are guaranteed to be bounded within a compact neighborhood of the true weights. We demonstrated the effectiveness of this method for adaptive flight control of a fixed wing unmanned aircraft.

The second method, termed as Least Squares Modification, presents a modification to baseline gradient based adaptive laws that drive the adaptive weights to an estimate of the ideal weights. We used Lyapunov analysis to show that the least squares based adaptive law guarantees convergence of adaptive weights and tracking error if the stored data meets a condition on linear independence. An adaptive controller with the least squares modification was found to perform significantly better than the baseline adaptive controller in simulation study of control of wing rock dynamics.

For both these methods, we showed that verifiable conditions on linear independence of the recorded data are sufficient to guarantee improved convergence properties. Therefore, these methods can be used



(a) Evolution of adaptive weights with only baseline adaptive control

(b) Evolution of adaptive weights with least squares modification

Figure 9. Convergence of adaptive weights with and without least squares modification for control of wing rock dynamics. Note that the weights converge to their true values rapidly when least squares modification is used.

to improve the performance of adaptive controllers without requiring the system states to be persistently exciting.

Acknowledgments

This work was supported in part by NSF ECS-0238993 and NASA Cooperative Agreement NNX08AD06A.

References

[1]Stephan Boyd and Sankar Sastry. Necessary and sufficient conditions for parameter convergence in adaptive control. *Automatica*, 22(6):629–639, 1986.

[2]Otto Bretscher. *Linear Algebra with Applications*. Prentice Hall, 2001.

[3]A. Calise, N. Hovakimyan, and M. Idan. Adaptive output feedback control of nonlinear systems using neural networks. *Automatica*, 37(8):1201–1211, 2001. Special Issue on Neural Networks for Feedback Control.

[4]Chengyu Cao and Naira Hovakimyan. L1 adaptive output feedback controller for systems with time-varying unknown parameters and bounded disturbances. In *Proceedings of American Control Conference*, New York, 2007.

[5]Girish Chowdhary. *Concurrent Learning for Convergence in Adaptive Control Without Persistentcy of Excitation*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2010. to be submitted in August 2010.

[6]Girish Chowdhary, Wesley Debusk, and Eric Johnson. Real-time system identification of a small multi-engine aircraft with structural damage. In *AIAA Infotech@Aerospace*, 2010.

[7]Girish Chowdhary and Eric Johnson. Recursively updated least squares based modification term for adaptive control. In *American Control Conference 2010*, 2010.

[8]Girish Chowdhary and Eric N. Johnson. A method for convergence in adaptive control without persistency of excitation. In *49th IEEE Conference on Decision and Control*, 2010. submitted.

[9]Girish V. Chowdhary and Eric N. Johnson. 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.

[10]Girish V. Chowdhary and Eric N. Johnson. 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.

[11]Girish V. Chowdhary and Eric N. Johnson. Flight test validation of long term learning adaptive flight controller. In *Proceedings of the AIAA Guidance Navigation and Control Conference, held at Chicago, IL*, 2009.

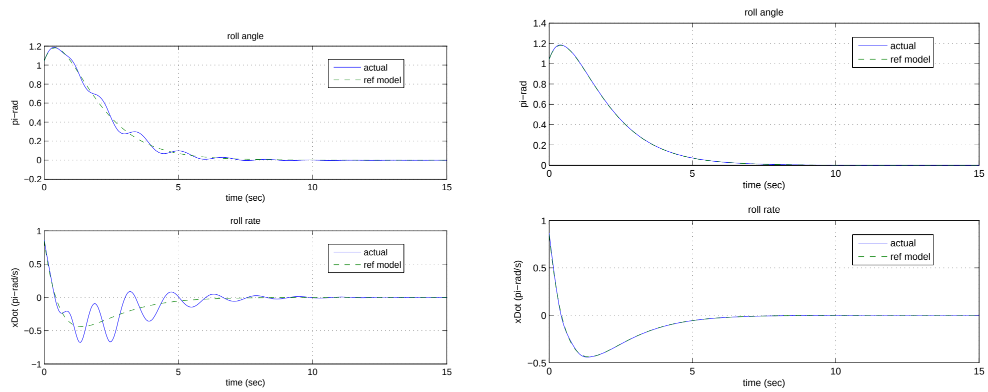
[12]Henrik B. Christophersen, Wayne R. Pickell, James C. Neidoefer, Adrian A. Koller, Suresh K. Kannan, and Eric N. Johnson. A compact guidance, navigation, and control system for unmanned aerial vehicles. *AIAA Journal of Aerospace Computing, Information, and Communication*, 3, May 2006.

[13]Girish Chwodhary and Eric Johnson. A method for convergence in adaptive control without persistency of excitation. *IEEE Transactions on Automatic Control*, 2010. in final stage for submission.

[14]Wesely Debusk, Girish Chowdhary, and Johnson Eric. Real-time system identification of a small multi-engine aircraft. In *Proceedings of AIAA AFM*, 2009.

[15]Arthur Gelb. *Applied Optimal Estimation*. MIT Press, Cambridge, 1974.

[16]Wassim M. Haddad and VijaySekhar Chellaboina. *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*. Princeton University Press, Princeton, 2008.



(a) Performance of adaptive controller with only baseline adaptive law
(b) Performance of adaptive controller with least squares modification

Figure 10. Performance of adaptive controller with and without least squares modification for control of wing rock dynamics. Note that the states track the reference model with great accuracy when using least squares modification.

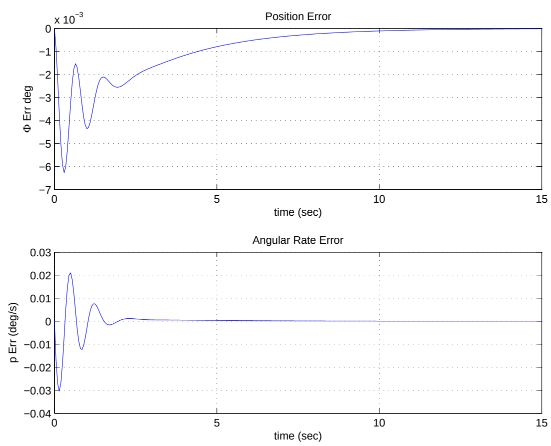


Figure 11. Evolution of tracking error with least squares modification. Note that $e(t) \to 0$

[17] Simon Haykin. *Neural Networks a Comprehensive Foundation*. Prentice Hall, USA, Upper Saddle River, 2 edition, 1998.

[18] N. Hovakimyan, B. J. Yang, and A. Calise. Robust adaptive output feedback control methodology for multivariable non-minimum phase nonlinear systems. *Automatica*, 42(4):513–522, 2006.

[19] Naira Hovakimyan. *Robust Adaptive Control*. Unpublished, 2008.

[20] Petros A. Ioannou and Jung Sun. *Robust Adaptive Control*. Prentice-Hall, Upper Saddle River, 1996.

[21] Ravindra V. Jategaonkar. *Flight Vehicle System Identification A Time Domain Approach*, volume 216 of *Progress in Astronautics and Aeronautics*. American Institute of Aeronautics and Astronautics, Reston, 2006.

[22] E. Johnson and S. Kannan. Adaptive flight control for an autonomous unmanned helicopter. In *Proceedings of the AIAA Guidance Navigation and Control Conference, held at Monterrery CA*, 2002.

[23] E. Johnson, M. Turbe, A. Wu, and S. Kannan. Flight results of autonomous fixed-wing uav transitions to and from stationary hover. In *Proceedings of the AIAA Guidance Navigation and Control Conference*, August 2006.

[24] E. N. Johnson. *Limited Authority Adaptive Flight Control*. PhD thesis, Georgia Institute of Technology, Atlanta Ga, 2000.

[25] Eric Johnson and Girish Chowdhary. Guidance and control of an airplane under severe structural damage. In *AIAA Infotech@Aerospace*, 2010. *Invited*.

[26] Eric N. Johnson and Seung Min Oh. Adaptive control using combined online and background learning neural network. In *Proceedings of CDC*, 2004.

[27] Eric N. Johnson and Daniel P. Schrage. System integration and operation of a research unmanned aerial vehicle. *AIAA Journal of Aerospace Computing, Information and Communication*, 1(1):5–18, Jan 2004.

[28]Suresh Kannan, Adrian Koller, and Eric Johnson. Simulation and development environment for multiple heterogeneous uavs. In *Modeling and Simulation Technologies Conference and Exhibit*, Providence, Rhode Island, 2004.

[29]Suresh K. Kannan. *Adaptive Control of Systems in Cascade with Saturation*. PhD thesis, Georgia Institute of Technology, Atlanta Ga, 2005.

[30]Nakawan Kim. *Improved Methods in Neural Network Based Adaptive Output Feedback Control, with Applications to Flight Control*. PhD thesis, Georgia Institute of Technology, Atlanta Ga, 2003.

[31]Y. H. Kim and F.L. Lewis. *High-Level Feedback Control with Neural Networks*, volume 21 of *Robotics and Intelligent Systems*. World Scientific, Singapore, 1998.

[32]Ali Kutay, Girish V. Chowdhary, A. Calise, and Eric N. Johnson. A comparision of two novel direct adaptive control methods under actuator failure accommodation. In *Proceedings of the AIAA GNC Conference, held at Honolulu, HI*, 2008.

[33]Eugene Lavertsky and Kevin Wise. Flight control of manned/unmanned military aircraft. In *Proceedings of American Control Conference*, 2005.

[34]F. L. Lewis. Nonlinear network structures for feedback control. *Asian Journal of Control*, 1:205–228, 1999. Special Issue on Neural Networks for Feedback Control.

[35]Morgan M. Monahemi and Miroslav Krstic. Control of wingrock motion using adaptive feedback linearization. *Journal of Guidance Control and Dynamics*, 19(4):905–912, August 1996.

[36]E. A. Morelli. Real time parameter estimation in the frequency domain. *Journal of Guidance Control and Dynamics*, 23(5):812–818, 2000.

[37]Flavio Nardi. *Neural Network based Adaptive Algorithms for Nonlinear Control*. PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332, nov 2000.

[38]K. Narendra and A. Annaswamy. A new adaptive law for robust adaptation without persistent excitation. *IEEE Transactions on Automatic Control*, 32(2):134–145, February 1987.

[39]Lester S.H. Ngia and Jonas Sjoeberg. Efficient training of nerual nets for nonlinear adaptive filtering using a recursive levenberg-marquardt algorithm. *IEEE Transactions on Signal Processing*, 48(7), July 2000.

[40]Nhan Nguyen, Kalamanje Krishnakumar, John Kaneshige, and Pascal Nespeca. Dynamics and adaptive control for stability recovery of damaged asymmetric aircraft. In *AIAA Guidance Navigation and Control Conference*, Keystone, CO, 2006.

[41]J. Park and I.W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computatations*, 3:246–257, 1991.

[42]H.D. Patiño, R. Carelli, and B.R. Kuchen. Neural networks for advanced control of robot manipulators. *IEEE Transactions on Neural Networks*, 13(2):343–354, Mar 2002.

[43]S. N. Singh, W. Yim, and W. R. Wells. Direct adaptive control of wing rock motion of slender delta wings. *Journal of Guidance Control and Dynamics*, 18(1):25–30, Feb. 1995.

[44]Johan A.K. Suykens, Joos P.L. Vandewalle, and Bart L.R. De Moor. *Artificial Neural Networks for Modelling and Control of Non-Linear Systems*. Kluwer, Norwell, 1996.

[45]Gang Tao. *Adaptive Control Design and Analysis*. Wiley, New York, 2003.

[46]Konstantine Volyanskyy and Anthony Calise. An error minimization method in adaptive control. In *Proceedings of AIAA Guidance Navigation and Control conference*, 2006.

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:

$$\begin{pmatrix} \dot{x} \\ \ddot{x} \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} x \\ \dot{x} \end{pmatrix} \tag{51}$$

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.[15] 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. 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:

$$
\begin{aligned}
\Phi_k &= e^{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} dt}, && (52)\\
Z_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{pmatrix} x \\ \dot{x} \end{pmatrix}, && (53)\\
\hat{x}_k(-) &= \Phi_k \hat{x}_{k-1}, && (54)\\
P_k(-) &= \Phi_k P_{k-1} {\Phi_k}^T + Q_k, && (55)\\
K_k &= P_k(-) {H_k}^T [H_k P_k(-) {H_k}^T + R_k]^{-1}, && (56)\\
\hat{x}_k(+) &= \hat{x}_k(-) + K_k [Z_k - H_k \hat{x}_k(-)], && (57)\\
P_k(+) &= [I - K_k H_k] P_k(-). && (58)
\end{aligned}
$$

The smoothed state estimate can be given as:

$$
\hat{x}_{k|N} = \hat{x}_{k|N-1} + B_N [\hat{x}_N(+) - \hat{x}_N(-)], \qquad (59)
$$

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