Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

A Neuro-Adaptive Augmented Dynamic Inversion Design for Robust Auto-Landing

Pradeep R. Ambati * **Radhakant Padhi** **

* *Former ME student, Dept. of Aerospace Eng., Indian Institute of Science, Bangalore, India.*
** *Associate Professor, Dept. of Aerospace Eng., Indian Institute of Science, Bangalore, India (e-mail: padhi@aero.iisc.ernet.in).*

Abstract: A neuro-adaptive augmented nonlinear dynamic inversion approach is proposed in this paper for robust automatic landing of unmanned aerial vehicles. Following the philosophy of indirect adaptive control, a set of linear in weight neural networks are used to rapidly learn the unknown part of the system model online. This continuously updated model is simultaneously used in a dynamic inversion framework that results in an adaptive controller which is quite robust to the modeling inaccuracies (i.e. parametric uncertainties of the model) and external wind shear disturbance. The training rule of neural networks is obtained from Lyapunov stability theory, where a Sobolev norm based Lyapunov function is chosen. This leads to 'directional learning', resulting in fast learning without exciting too much of transient oscillations. Robust performance of this neuro-adaptive augmented nonlinear controller is successfully validated through six degree-of-freedom simulation studies.

1. INTRODUCTION

Unmanned aerial vehicles (UAVs) are useful for numerous applications such as reconnaissance and surveillance, battle damage assessment, traffic monitoring, crime prevention, detection and containment of hazardous leakages in industries, assessment and rehabilitation in case of natural calamities etc. Unfortunately, however, many UAVs get either destroyed or severely damaged during landing. Hence, for successful repeated deployment, it is obvious that UAVs should have good automatic landing capability.

A landing phase trajectory typically consists of approach, glideslope and flare [4]. Auto-landing demands careful design and closely tracking of a good desired landing path in all the three segments. Linearized model of the aircraft have been used in the literature for auto-landing using separate longitudinal and lateral dynamics. However, linear system based approaches have a strong limitation that they work within a small operating range. The philosophy of 'gain scheduling' can be used to overcome this limitation to a limited extent. However gain scheduling is a tedious process and there is no guarantee that the interpolated gains can assure stability of the closed loop system [5]. Nonlinear control design techniques have also been used in the literature. Among various nonlinear techniques, dynamic inversion technique has been proposed for automatic landing aircrafts and UAVs [2], [8], [4], [6].

Despite its reported success in simulation studies, it is well-known that the dynamic inversion approach suffers from the drawback of its sensitivity for modeling inaccuracies, which is inherently present in aerospace vehicles owing to aerodynamic force and moment modeling from wind tunnel and flight testing. Hence, this issue parametric uncertainty in the model needs to be addressed explicitly with sufficient confidence. In addition, another factor that influences the aircraft (especially UAV) is the issue of 'wind shear', which can lead to substantial deviation of flight path because of the light weight of vehicle and its low speed in landing phase. Presence of wind shear can occur from variety of sources, such as atmospheric factors like microbursts and geographical factors like wake effects of building near landing sites [1].

An indirect adaptive control philosophy is used in this paper to address above issues. In this approach, which is largely inspired from the approach followed in [3], a set of linear in weight neural networks are used to rapidly learn the unknown components of the system model and update the model online. This updated model is then used in a dynamic inversion framework that results in an adaptive controller which is quite robust to modeling inaccuracies and external wind shear disturbance. Note that the training rule of neural networks is obtained by following Lyapunov theory, where a Sobolev norm based Lyapunov function is chosen to learn both the unknown function as well as the gradient vector of each of its components. This leads to 'directional learning', resulting in fast learning without exciting too much of transient oscillations. Two sets of neural networks are used, one in the guidance loop using translational dynamics and another in the control loop using rotational dynamics. The outer loop generates the necessary body rates, whereas the inner loop generates the necessary fin deflection. The overall structure leads to a robust controller which is capable of addressing the modelling inaccuracy and wind shear issues simultaneously. Performance of this neuro-adaptive augmented nonlinear dynamic inversion controller is successfully validated through six degree-of-freedom (Six-DOF) simulation studies.

2. AIRCRAFT MODEL

2.1 Dynamics without wind shear

Under the assumptions of airplane to be a rigid body and earth to be flat, the set of Six-DOF equations of motion are given by following differential equations.

Translational dynamic equations

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw - g\sin\theta + \dfrac{X_a + X_t}{m} \\ pw - ru + g\sin\phi\cos\theta + \dfrac{Y_a}{m} \\ qu - pv + g\cos\phi\cos\theta + \dfrac{Z_a}{m} \end{bmatrix} \tag{1}$$

Rotational dynamic equations

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} c_1 rq + c_2 pq + c_3 L_a + c_4 N_a \\ c_5 pr + c_6 \left(r^2 - p^2\right) + c_7 \left(M_a + M_t\right) \\ c_8 pq - c_2 rq + c_4 L_a + c_9 N_a \end{bmatrix} \tag{2}$$

Translational kinematic equations

$$\begin{bmatrix} \dot{x} \\ \\ \\ \dot{y} \\ \\ \\ \dot{h} \end{bmatrix} = \begin{bmatrix} u\cos\theta\cos\psi \\ +v\left(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi\right) \\ +w\left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right) \\ u\cos\theta\sin\psi \\ +v\left(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi\right) \\ +w\left(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\right) \\ u\sin\theta - v\sin\phi\cos\theta - w\cos\phi\cos\theta \end{bmatrix} \tag{3}$$

Rotational kinematic equations

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \\ q\cos\phi - r\sin\phi \\ q\sin\phi\sec\theta + r\cos\phi\sec\theta \end{bmatrix} \tag{4}$$

In above equations $X_a,\ Y_a,\ Z_a$ are the aerodynamic forces and $L_a,\ M_a,\ N_a$ are the moments about the body axis. X_t is the thrust force in body axis X direction and M_t is the moment around the Y axis caused by thrust. Constants $c_1, c_2, \ldots, c_9$ in above equations are function of inertial properties of aircraft given as

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_8 \\ c_9 \end{bmatrix} = \frac{1}{I_{xx}I_{zz} - I_{xz}^2} \begin{bmatrix} I_{zz}\left(I_{yy} - I_{zz}\right) - I_{xz}^2 \\ I_{xz}\left(I_{xx} - I_{yy} + I_{zz}\right) \\ I_{zz} \\ I_{xz} \\ I_{xz}^2 + I_{xx}\left(I_{xx} - I_{yy}\right) \\ I_{xx} \end{bmatrix}$$

$$\begin{bmatrix} c_5 \\ c_6 \\ c_7 \end{bmatrix} = \frac{1}{I_{yy}} \begin{bmatrix} \left(I_{zz} - I_{xx}\right) \\ I_{xz} \\ 1 \end{bmatrix}$$

Inertia properties of the 6 kg vehicle are given in Table 1. Nominal aerodynamic model and associated parameter values can be found in [6].

Table 1. Moment of Inertias of AE-2

$I_{xx}\ (kgm^2)$	$I_{yy}\ (kgm^2)$	$I_{zz}\ (kgm^2)$	$I_{xz}\ (kgm^2)$
0.51	**0.89**	**0.91**	**0.0015**

2.2 Dynamics with wind shear

General nonlinear model of aircraft developed by Etkin is derived assuming zero or constant wind. Performance and control of aircraft in extreme wind variation pose serious threat of losing aircraft control. Frost [1] incorporated spatial and temporal variations of wind into Six-DOF model.

$$\begin{bmatrix} \dot{x} \\ \\ \\ \dot{y} \\ \\ \\ \dot{h} \end{bmatrix} = \begin{bmatrix} u\cos\theta\cos\psi \\ +v\left(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi\right) \\ +w\left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right) + W_{xE} \\ u\cos\theta\sin\psi \\ +v\left(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi\right) \\ +w\left(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\right) + W_{yE} \\ u\sin\theta - \cos\theta(v\sin\phi + w\cos\phi) - W_{zE} \end{bmatrix} \tag{5}$$

Translational kinematic equations are modified by adding the wind velocity vector $[W_{xE},\ W_{yE},\ -W_{zE}]$ with respect to inertial frame on right hand side. Wind velocity vector $[W_x, W_y, W_z]$ is with respect to body frame. There will not be any change in rotational kinematic equations. In rotational dynamic equations, moment terms depend on wind vector and its gradient. Translational dynamic equations are modified as:

$$\begin{bmatrix} \dot{u} \\ \\ \dot{v} \\ \\ \dot{w} \end{bmatrix} = \begin{bmatrix} r\left(v + W_y\right) - q\left(w + W_z\right) - \dot{W}_x \\ -g\sin\theta + \dfrac{X_a + X_t}{m} \\ p\left(w + W_z\right) - r\left(u + W_x\right) - \dot{W}_y \\ +g\sin\phi\cos\theta + \dfrac{Y_a}{m} \\ q\left(u + W_x\right) - p\left(v + W_y\right) - \dot{W}_z \\ +g\cos\phi\cos\theta + \dfrac{Z_a}{m} \end{bmatrix} \tag{6}$$

Transformation of wind vector and its derivative from inertial frame to body frame is as shown below.

$$\begin{bmatrix} W_x \\ W_y \\ W_z \end{bmatrix} = L_{BE} \begin{bmatrix} W_{xE} \\ W_{yE} \\ W_{zE} \end{bmatrix} \tag{7}$$

where

$$L_{BE} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ (\sin\phi\sin\theta\cos\psi) & (\sin\phi\sin\theta\sin\psi) & \sin\phi\cos\theta \\ -(\cos\phi\sin\psi) & +(\cos\phi\cos\psi) & \\ (\cos\phi\sin\theta\cos\psi) & (\cos\phi\sin\theta\sin\psi) & \cos\phi\cos\theta \\ +(\sin\phi\sin\psi) & -(\sin\phi\cos\psi) & \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} \dot{W}_x \\ \\ \dot{W}_y \\ \\ \dot{W}_z \end{bmatrix} = \begin{bmatrix} \left(\dfrac{\partial W_x}{\partial x}\right)_B (u + W_x) + \left(\dfrac{\partial W_x}{\partial y}\right)_B (v + W_y) \\ + \left(\dfrac{\partial W_x}{\partial z}\right)_B (w + W_z) + \left(\dfrac{\partial W_x}{\partial t}\right)_B \\ \left(\dfrac{\partial W_y}{\partial x}\right)_B (u + W_x) + \left(\dfrac{\partial W_y}{\partial y}\right)_B (v + W_y) \\ + \left(\dfrac{\partial W_y}{\partial z}\right)_B (u + W_x) + \left(\dfrac{\partial W_y}{\partial t}\right)_B \\ \left(\dfrac{\partial W_z}{\partial x}\right)_B (u + W_x) + \left(\dfrac{\partial W_z}{\partial y}\right)_B (v + W_y) \\ + \left(\dfrac{\partial W_z}{\partial z}\right)_B (w + W_z) + \left(\dfrac{\partial W_z}{\partial t}\right)_B \end{bmatrix} \tag{9}$$

$$\nabla_B \boldsymbol{W} = L_{BE}^T \nabla_E \boldsymbol{W} L_{BE} \tag{10}$$

where

$$\nabla \boldsymbol{W} = \begin{bmatrix} \frac{\partial W_x}{\partial x} & \frac{\partial W_y}{\partial x} & \frac{\partial W_z}{\partial x} \\ \frac{\partial W_x}{\partial y} & \frac{\partial W_y}{\partial y} & \frac{\partial W_z}{\partial y} \\ \frac{\partial W_x}{\partial z} & \frac{\partial W_y}{\partial z} & \frac{\partial W_z}{\partial z} \end{bmatrix}$$

3. NEURO ADAPTIVE CONTROL

3.1 Neural networks

Neural networks are used in system identification as they can be easily trained for curve fitting by a given set of data points defined by input signal and a desired response. With the criterion for best fit, Radial Basis Functions(RBF) can be viewed as universal approximators for most of nonlinear classes. The input output mapping performed by RBF is

$$y = \sum_{i=1}^{m} w_i \phi_i(x) \tag{11}$$

The term ϕ_i is the i^{th} radial basis function which represents nonlinear transformation and the weights w_i maps linear transformation. Gaussian function is selected as radial basis function which is shown below

$$\phi_i(x) = e^{\frac{-(x-x_i)^2}{\sigma_i^2}}, \qquad i = 1, 2, \dots, m \tag{12}$$

where x_i is the center, σ_i is the width of i^{th} radial basis function.

3.2 Nonlinear Dynamic Inversion

Consider a nonlinear dynamical system which is affine in control and given by

$$\dot{X} = f(X) + g(X)U, \quad X(0) = X_0 \tag{13}$$

$$Y = h(X) \tag{14}$$

where $X \in R^n, U \in R^m, Y \in R^p$ are the state, control and output vectors of nominal system respectively. We assume the system is point wise controllable. The objective is to design a control U so that $Y \longrightarrow Y^*$ as $t \longrightarrow \infty$, where Y^* is the commanded signal for the Y to track. We assume Y^* is bounded, smooth and slowly varying.

To achieve the above objective using the chain rule of derivative the expression for Y can be written as

$$\dot{Y} = f_Y(X) + g_Y(X)U \tag{15}$$

where $f_Y = \left[\frac{\partial h}{\partial X}\right] f(X)$ and $g_Y = \left[\frac{\partial h}{\partial X}\right] g(X)$. Next, defining $E = (Y - Y^*)$, the controller is synthesized such that a stable linear error dynamics is satisfied $\dot{E} + KE = 0$, where K is chosen to be positive definite matrix.

$$U = [g_Y(X)]^{-1}[-f_Y(X) - K(Y - Y^*) + \dot{Y}^*] \tag{16}$$

In order to overcome the potential performance degradations of a dynamic inversion controller due to imperfect inversion or non-accurate aerodynamics, the control loop is augmented with adaptive elements. Online neural networks are used to augment the dynamic inversion

3.3 Neuro adaptive control

Consider a nonlinear plant with known dynamics and parameters as

$$\dot{X}_d = f(X_d, U_d), \quad where\ X_d \in R^n \tag{17}$$

$$U_d \in R^m (m \leq n) \tag{18}$$

for which a nominal baseline control, U_d is designed for desired tracking of states. Under uncertainty and parameter variation, adaptive control has been widely used for nonlinear systems with modeling uncertainties. If the structure of uncertainty is not known neural networks can be used as adaptive element by its universal approximation property.

Consider an actual plant with unmodeled dynamics as given below

$$\dot{X} = f(X) + g(X)U + d(X), \quad X(0) = X_0 \tag{19}$$

Here $d(X) \in \Re^n$ is the unmodeled dynamics. Neuro adaptive controller augments the nominal controller with an adaptive element updated based on Lyapunov theory. Let a virtual plant is created with state X^a and whose dynamics is given as

$$\dot{X}^a = f(X) + g(X)U + \hat{d}(X) + K_\tau(X - X^a) \tag{20}$$

The $\hat{d}(X)$ is an approximation of the actual function $d(X)$ and K_τ is a Hurwitz matrix which contains the desired time constants with which we want the virtual plant to track the actual plant. This ensures the actual state to follow nominal state through $X \longrightarrow X_a \longrightarrow X_d$ as $t \rightarrow \infty$. We shall define the error as $E = X - X^a$ and

$$\dot{E} = d(X) - \hat{d}(X) - K_\tau E \tag{21}$$

For the purpose of analysis and identification the error is decomposed into individual channels as $e_i = x_i - x_i^a$. The i^{th} channel error dynamics is given as

$$\dot{e}_i = d_i(X) - \hat{d}_i(X) - k_{\tau_i} e_i, \ \ i = 1, 2, \dots, n \tag{22}$$

The actual implementation requires an approximator that approximates the unmodeled dynamics $d_i(X)$ in the i^{th} channel. For this purpose we choose a single layer neural network with nonlinear basis functions. The architecture is

$$\hat{d}_i(X) = \hat{W_i}^T \Phi_i(X), \ \ W_i \in \Re^p \tag{23}$$

where, $\hat{W}_i$ are the weights and $\Phi_i(X)$ are the basis. Next, we shall consider that there exists an ideal approximator for the unknown function which approximates $d_i(X)$ with an ideal approximation error ϵ_i for the chosen basis $\Phi_i(X)$.

$$d_i(X) = {W_i}^T \Phi_i(X) + \epsilon_i \tag{24}$$

The weights W_i are the ideal weights which are unknown. The error dynamics in (22) can be written as

$$\dot{e}_i = {W_i}^T \Phi_i(X) + \epsilon_i - \hat{W_i}^T \Phi_i(X) - k_{\tau_i} e_i \quad (25)$$

The error in weights of the i^{th} approximating network is defined as $\tilde{W}_i = W_i - \hat{W}_i$ and

$$\dot{\tilde{W}}_i = -\dot{\hat{W}}_i, \quad W_i = constant \quad (26)$$

As part of the online adaptation the weights of the approximating networks $\hat{W}_i$ should approach the ideal weights W_i asymptotically, i.e., $\tilde{W}_i \to 0 \ as \ t \to \infty$ A Lyapunov approach is discussed in the next section for updating $\hat{W}_i$ online.

3.4 Lyapunov Analysis and Weight Update Rule

In the current analysis there are three quantities whose asymptotic stability has to be guaranteed

(1) e_i, the i^{th} channel error
(2) $\tilde{W}_i$, the error in i^{th} network weights
(3) $\left[\frac{\partial d_i(X)}{\partial X} - \frac{\partial \hat{d}_i(X)}{\partial X}\right]$, the error in i^{th} unknown function partial derivative

The positive definite Lyapunov function candidate is

$$V_i(e_i, \tilde{W}_i) = \beta_i \frac{e_i^2}{2} + \frac{\tilde{W}_i^T \tilde{W}_i}{2\gamma_i} + \left[\frac{\partial d_i(X)}{\partial X} - \frac{\partial \hat{d}_i(X)}{\partial X}\right]^T \times \frac{\Theta_i}{2} \left[\frac{\partial d_i(X)}{\partial X} - \frac{\partial \hat{d}_i(X)}{\partial X}\right] \quad (27)$$

where β_i, γ_i Θ_i are positive definite quantities. The partial derivative of $d_i(X)$ and $\hat{d}_i(X)$ are replaced to get

$$V_i(e_i, \tilde{W}_i) = \beta_i \frac{e_i^2}{2} + \frac{\tilde{W_i}^T \tilde{W}_i}{2\gamma_i} + \tilde{W_i}^T \left[\frac{\partial \Phi_i}{\partial X}\right] \frac{\Theta_i}{2} \left[\frac{\partial \Phi_i}{\partial X}\right]^T \tilde{W}_i \quad (28)$$

The dimensions of various quantities are

$$\tilde{W}_i \in \Re^p \ , \ \Phi_i(X) \in \Re^p \ , \ \frac{\partial \Phi_i(X)}{\partial X} \in \Re^{p \times n} \ , \ \Theta_i \in \Re^{n \times n} \ , \ V_i \in \Re$$

The Lie derivative of the Lyapunov function candidate is given by,

$$\dot{V}_i = \beta_i e_i \dot{e}_i - \frac{\tilde{W}_i^T \dot{\hat{W}}_i}{\gamma_i} - \tilde{W}_i \left[\frac{\partial \Phi_i}{\partial X}\right] \Theta_i \left[\frac{\partial \Phi_i}{\partial X}\right]^T \dot{\hat{W}}_i \quad (29)$$

Substituting the error dynamics from (25) in above equation, we get

$$\dot{V}_i = \tilde{W_i^T} \left\{ \beta_i e_i \Phi_i(X) - \frac{\dot{\hat{W}}_i}{\gamma_i} - \left[\frac{\partial \Phi_i}{\partial X}\right] \Theta_i \left[\frac{\partial \Phi_i}{\partial X}\right]^T \dot{\hat{W}}_i \right\} + \beta_i e_i \epsilon_i - \beta_i k_{\tau_i} e_i^2 \quad (30)$$

Equating the coefficient of $\tilde{W_i^T}$ to zero lead to the weight update rule in continuous time.

$$\dot{\hat{W}}_i = \beta_i e_i \left(\frac{I_p}{\gamma_i} + \left[\frac{\partial \Phi_i}{\partial X}\right] \Theta_i \left[\frac{\partial \Phi_i}{\partial X}\right]^T \right)^{-1} \Phi_i(X) \quad (31)$$

The matrix $\left[\frac{\partial \Phi_i}{\partial X}\right] \Theta_i \left[\frac{\partial \Phi_i}{\partial X}\right]^T$ is singular for $n < p$. But the matrix is always positive definite, so adding $\frac{I_p}{\gamma_i}$ will make the matrix nonsingular $\forall (n, \ p) \in \mathcal{N}$. The left over terms in the $\dot{V}_i$ equation are

$$\dot{V}_i = \beta_i e_i \epsilon_i - k_{\tau_i} \beta_i e_i^2 \quad (32)$$

The condition for which above equation becomes negative definite is $k_{\tau_i} > \frac{\epsilon_i}{e_i}$. Because k_{τ_i} is the time constant of the i^{th} channel, it is always positive. Therefore, we consider the absolute value on both sides of the inequality leading to a condition as $|e_i| > \frac{|\epsilon_i|}{k_{\tau_i}}$, which means that if the absolute error in the i^{th} channel exceeds the value in RHS then the Lyapunov function becomes negative definite and positive definite otherwise. Therefore, if the network weights are updated based of the rule given in (31), then the identification happens as long as absolute error is greater than certain value. By increasing k_{τ_i} error bound can be theoretically reduced.

4. ADAPTIVE AUTO-LANDING OF UAV

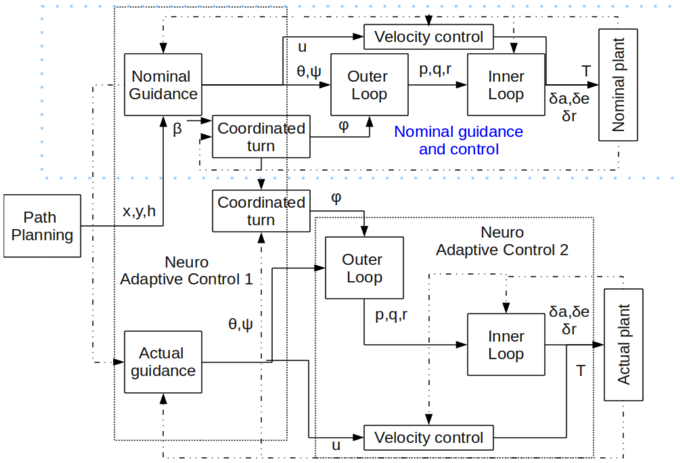

Fig. 1. Auto-landing guidance and control architecture

Nominal guidance together with dynamic inversion controller [7] is the reference model for neuro adaptive control. Fig: 1 shows the auto-landing architecture that is employed for AE-2. From Fig: 1, two neuro adaptive controllers are used, one in guidance loop and another in control. Neuro adaptive 1 adapts the unknown wind model with nominal guidance as reference model. This is detailed in adaptive guidance section and the simulation results for unknown wind shear provided good tracking of uncertain wind characteristics. Then the uncertainties that pops out in rotational dynamic equations are adapted by using the second neuro adaptive control. Combining these two stages of adaptation, UAV tracks the nominal desired path that is generated based on dynamic inversion as in the reference model. For training both the networks Sobolev norm based Lyapunov function are chosen. This resulted in fast learning without exciting too much of transient oscillations and also reduced the need for larger gains for fast learning. Moreover two networks as it used here can result loss of learning due to high gains. Besides these, UAV landing require quick adaptation for disturbances with less undesired transients. For this high gains are not apt choice as it may result for control saturation at crucial stage of landing.

4.1 Nominal guidance and control

Guidance These phases of landing is achieved by heading control for aligning the aircraft along the runway, altitude control to get desired glide slope angle and sink rate and having a coordinate turn constraint.

The desired sideward distance is achieved by having first order error dynamics in y and by proper substitution an expression for desired ψ is obtained as

$$\psi^* = \sin^{-1}\left(\frac{\dot{y}^* - k_y(y - y^*)}{\sqrt{a_y^2 + b_y^2}}\right) - \tan^{-1}\left(\frac{b_y}{a_y}\right) \quad (33)$$

where,

$$a_y \triangleq ucos\theta + vsin\phi sin\theta + wcos\phi sin\theta, b_y \triangleq vcos\phi - wsin\phi$$

Similarly by altitude control we can get an expression for desired pitch angle as,

$$\theta^* = \sin^{-1}\left(\frac{\dot{h}^* - k_h(h - h^*)}{\sqrt{a_h^2 + b_h^2}}\right) - \tan^{-1}\left(\frac{b_h}{a_h}\right) \quad (34)$$

where,

$$a_h \triangleq u, b_h \triangleq vsin\phi + wcos\phi$$

Coordinated turn constraint is required to make side slip angle zero. By using first order error in β and appropriate substitution we get an expression for desired roll angle.

$$\phi^* = \sin^{-1}\left(\frac{ru - pw - Y_a + \dot{V}_t \sin\beta - k_b \beta V_t \cos\beta}{g\cos\theta}\right) \quad (35)$$

Nominal Control

Outer loop Based on first order error dynamics in outer loop, desired rotation rates are achieved that are to be fed to inner loop

$$\begin{bmatrix} p^* \\ q^* \\ r^* \end{bmatrix} = \begin{bmatrix} \dot{\phi}^* - k_\phi(\phi - \phi^*) - (qsin\phi + rcos\phi)tan\theta \\ sec\phi(\dot{\theta}^* - k_\theta(\theta - \theta^*) + r\sin\phi) \\ sec\phi cos\theta(\dot{\psi}^* - k_\psi(\psi - \psi^*)) - qtan\phi \end{bmatrix} \quad (36)$$

Inner Loop Control Here aerodynamic controls are calculated by enforcing first order error dynamics in body rates. Proper substitution of equation in $\dot{p}$, $\dot{q}$, $\dot{r}$, we get as

$$f_r + g_r U = b_r \quad (37)$$

and the desired control from above equation is given by

$$U = g_r^{-1}(b_r - f_r) \quad (38)$$

where, $U = [\delta a \quad \delta e \quad \delta r]^T$ and other terms are defined as follows

$$f_r \triangleq \begin{bmatrix} c_1 rq + c_2 pq + c_3 L_{a_x} + c_4 N_{a_x} \\ c_5 pr + c_6(r^2 - p^2) + c_7(M_{a_x} - M_t) \\ c_8 pq - c_2 rq + c_4 L_{a_x} + c_9 N_{a_x} \end{bmatrix}$$

$$g_r \triangleq \begin{bmatrix} c_3 L_{a_u} & 0 & c_4 N_{a_u} \\ 0 & c_7 M_{a_u} & 0 \\ c_4 L_{a_u} & 0 & c_9 N_{a_u} \end{bmatrix} \quad b_r \triangleq \begin{bmatrix} \dot{p}^* - k_p(p - p^*) \\ \dot{q}^* - k_q(q - q^*) \\ \dot{r}^* - k_r(r - r^*) \end{bmatrix}$$

Velocity control The forward velocity can be controlled by varying the thrust through throttle control given by

$$\sigma_t = g_u^{-1}(b_u - rv + qw + gsin\theta - X_a/m) \quad (39)$$

where,

$$g_u \triangleq \frac{T_{max}}{m}, b_u \triangleq \dot{u}^* - k_u(u - u^*)$$

4.2 Adaptive guidance

Nominal guidance for auto-landing through all three phases provides required attitude and forward velocity. These equations use translational kinematic equation which have wind terms. Hence actual plant differs nominal plant for which adaptive modeling is done using neural networks. This section details adaptive guidance which is represented as neuro adaptive 1 in Fig. 1.

The desired sideward distance is achieved by controlling desired ψ for which an expression is obtained as

$$\psi^* = \sin^{-1}\left(\frac{\dot{y}_{nom} - k_y(y - y_{nom}) - \hat{d}_y}{\sqrt{a_y^2 + b_y^2}}\right) - \tan^{-1}\left(\frac{b_y}{a_y}\right) \quad (40)$$

Similarly for altitude control we can get an expression for desired pitch angle.

$$\theta^* = \sin^{-1}\left(\frac{\dot{h}_{nom} - k_h(h - h_{nom}) - \hat{d}_h}{\sqrt{a_h^2 + b_h^2}}\right) + \tan^{-1}\left(\frac{b_h}{a_h}\right) \quad (41)$$

Coordinated turn constraint is required to make side slip angle zero. By using first order error in β and appropriate substitution we get an expression for desired roll angle.

$$\phi^* = \sin^{-1}\left(\frac{a_\beta + \left(\dot{\beta}_{nom} - k_b(\beta - \beta_{nom}) - \hat{d}_\beta\right) V_t \cos\beta}{g\cos\theta}\right) \quad (42)$$

where,

$$a_\beta = ru - pw - Y_a + \dot{V}_t \sin\beta$$

$\hat{d}_y, \hat{d}_h$ and $\hat{d}_\beta$ are adaptive elements which are modeled using neural networks

4.3 Adaptive Control

Inner loop has the second neuro adaptive control which calculates required control deflections as

$$f_r + g_r U + \hat{D} = b_r \quad (43)$$

and the desired control from above equation is given by

$$U = g_r^{-1}(b_r - f_r - \hat{D}) \quad (44)$$

where, $U = [\delta a \quad \delta e \quad \delta r]^T$ and adaptive terms are defined as follows

$$\hat{D} = \left[\, \hat{d}_p \; \hat{d}_q \; \hat{d}_r \,\right]^T$$

For neuro adaptive based velocity control, we get

$$\dot{u} = \dot{u}^* + k_u(u - u^*) - \hat{d}_u \quad (45)$$

Substituting $\dot{u}$ in equation we get the control solution as

$$\sigma_t = g_u^{-1}(b_u - rv + qw + gsin\theta - X_a/m - \hat{d}_u) \quad (46)$$

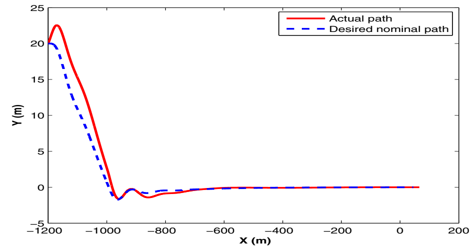

Fig. 2. Path of UAV in $x - y$ plane

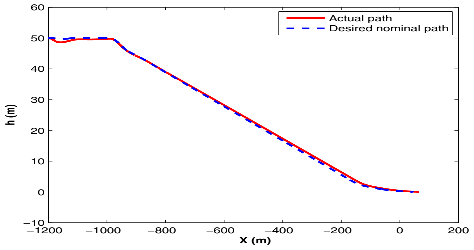

Fig. 3. Path of UAV in $x - h$ plane

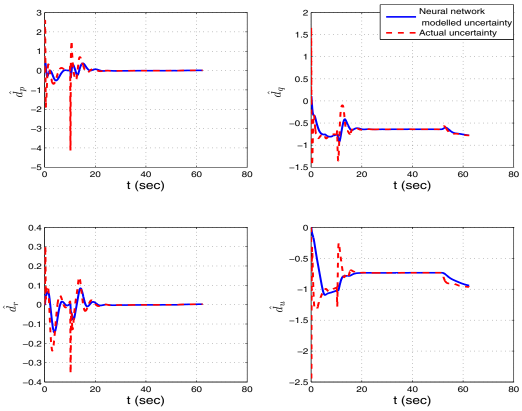

Fig. 4. Uncertainty due to parameter variation

5. SIMULATIONS

To check the adaptive architecture used for auto-landing, a simulation is shown with parameter uncertainty of about 50% along with wind shear disturbance. Uncertainty due to parameter variation in rotational dynamic equations is shown in Fig: 4 that match unknown parameter uncertainty during landing. Uncertainty due to wind in translational kinematic equations is shown in Fig: 5 which shows perfect tracking of unmodeled wind shear. Complete sequence of landing in all three phases follows desired path.

6. CONCLUSIONS

A neuro-adaptive augmented nonlinear dynamic inversion approach is proposed in this paper for robust automatic landing of UAVs. A set of neural networks are used to learn the unknown components of the system model and update the model online. This updated model is then used in a dynamic inversion framework leading to an adaptive controller which is quite robust to the modeling inaccuracies and external wind shear disturbance. Note

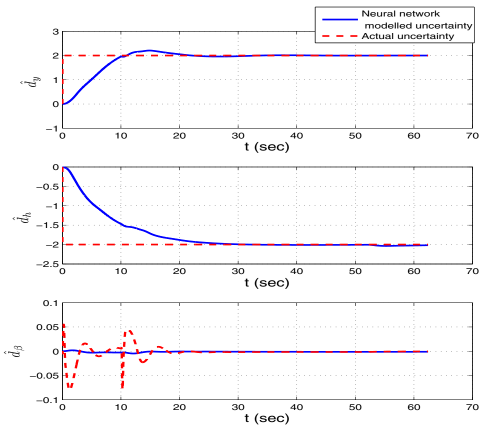

Fig. 5. Uncertainty due to wind shear disturbance

that the training rule of the neural networks is obtained by following the Lyapunov theory with a Sobolev norm based Lyapunov function, which leads to fast learning without exciting much of transient oscillations. The desired trajectory for landing has been made independent of time by scheduling the trajectory as a function of forward distance from runway. The problem of transition between the glide slope and flare is addressed by ensuring continuity and smoothness at transition. Performance of this neuro-adaptive augmented nonlinear controller has been successfully validated through six-DOF simulation studies.

REFERENCES

[1] W. Frost and R. L. Bowles. Wind shear terms in the equations of aircraft motion. *Journal of Aircraft*, volume 21, no. 11, pages 866–872, Nov 1984.

[2] C. Ha and H. Choi. Design of longitudinal autolanding guidance and control system in dynamic model inversion. *Proceedings of 6th Russain-Korean International Symposium on Science and Technology*, pages 7-10, 2002.

[3] R. Padhi, N. Unnikrishnan and S. N. Balakrishnan. Model following Neuro-Adaptive control design for non square, non affine nonlinear systems. *IET Control Theory and Applications*, volume 1, pages 1650–1661, Keystone, Colorado, Nov 2007.

[4] B. Prasad and S. Pradeep. Automatic landing system design using feedback linearization method. *Proceedings of AIAA Infotech Aerospace Conference and Exhibit*, Rohnert Park, California, May 2007.

[5] J. S. Shamma and M. Athans. Analysis of gain scheduled control for nonlinear plants. *IEEE Transactions on Automatic Control*, volume 35, no. 8, 1990.

[6] S. P. Singh and R. Padhi. Automatic path planning and control design for autonomous landing of UAVs using dynamic inversion. *Proceedings of American Control Conference*, pages 2409-2414, St. Louis, 2009.

[7] S.P. Singh. Autonomous landing of Unmanned Aerial Vehicle. Department of Aerospace Engineering, Indian Institute of Science, Bangalore, Feb 2009.

[8] M. J. Smith, S. S. M. Swei and M. B. Tischler. A study of the control of Micro Air Vehicles using the linear dynamic inverse approach. *AIAA 1st Intelligent Systems Technical Conference*, Chicago, Illinois, Sep 2004.