



Trajectory Tracking Controller Design Using Neural Networks For Tiltrotor UAV

Boo-Min Kim¹, Kwangchan Choi² and Byoung Soo Kim³
Gyeongsang National University, Gyeongnam, SOUTH KOREA

This paper presents a design of trajectory tracking controller for Smart UAV, a tiltrotor unmanned aerial vehicle. Its design objective is to control the tiltrotor aircraft with one fixed controller architecture for all flight modes, that is, helicopter mode, conversion mode, and airplane mode. The controller is implemented using twotime-scale separation and thus composed of inner-loop for attitude control and outer-loop for trajectory control. The dynamic model inversion technique is applied to both control loops such that the inner-loop DMI is driven using the angular equation of motion and the outer-loop DMI using the translational equation of motion. In addition on-line adaptive neural networks are employed to compensate the model inversion errors of the inner-loop and the outer-loop DMIs, due to the lack of full knowledge of tiltrotor aircraft dynamics. And the pseudo-control hedging algorithm is adopted to lessen instability problem caused by the time-delay of control loops and saturation and rate limit of actuators. The trajectory tracking controller is evaluated using a sophisticated 6DOF nonlinear simulation program with an approach and landing scenario including all flight modes of the Smart UAV.

Nomenclature

DMI	=	Dynamic model inversion
NN	=	Neural network
PCH	=	Pseudo control hedging
b	=	Control matrix of tracking error dynamics
e	=	Tracking error vector of state variables
F_x, F_y, F_z	=	Applied force vector components represented in body axis system
$F_{x_T}, F_{y_T}, F_{z_T}$	=	Applied force vector components represented in thrust axis system
K_p, K_d	=	Proportional gain, derivative gain
$L(\cdot)$	=	Coordinate transformation matrix
P, Q, R	=	Angular velocity
\hat{u}_{adi}	=	Adaptive control signal, Output of neural network
u_ϕ, u_θ, u_ψ	=	Pseudo control variables of angular acceleration represented in body axis system
U_x, U_y, U_z	=	Pseudo control variables of linear acceleration represented in inertia coordinate system
V_x, V_y, V_z	=	Pseudo control variables of linear acceleration represented in body-fixed coordinate system
W_x, W_y, W_z	=	Pseudo control variables of linear acceleration represented in rotor-fixed coordinate system
V_p	=	Total flight speed
\hat{w}_i	=	Weight vector of a neural network
X_e, Y_e, Z_e	=	Position vector components represented in inertia coordinate
U_e, V_e, W_e	=	Velocity vector components represented in inertia coordinate
$X_{\delta T}$	=	Thrust control derivative for airplane mode
$Z_{\delta c}$	=	Collective control derivative for helicopter mode
$\beta'_{ij}(\cdot)$	=	Basis function vector
β_M	=	Rotor tilting angle, i.e., 90° for rotating-wing mode
ϕ, θ, ψ	=	Euler angles

^{1,2} Graduate Students, School of Mechanical and Aerospace Eng., Gyeongsang National University

³ Professor, School of Mechanical and Aerospace Eng., Gyeongsang National University, Researcher of ReCAPT, bskim@gsnu.ac.kr

I. Introduction

The Smart Tiltrotor UAV(Unmanned Aerial Vehicle) has been developed in KARI(Korea Aerospace Research Institute) since 2002. The development of the Smart UAV and its demonstration program has been funded by the Korean government, as one of the 21st Century Frontier R&D Programs sponsored by the Ministry of Commerce, Industry and Energy of Korea. This tiltrotor has the unique capability to takeoff/land vertically from/to a small area, even in moving deck and to cruise with the high speed efficiency of a turboprop airplane. It has a high endurance loiter capability, as well. High speed and high maneuverability are essential when operating a moving platform(e.g., shipboard), so that no special platform concessions(i.e., ship's direction and speed) are required for launch, recovery, control. In the airplane mode, primary control is provided by conventional aerodynamic control surfaces, i.e. elevator, aileron, and rudder, added to propeller pitch; whereas in the helicopter mode, control is conducted by collective and cyclic pitch of the prop-rotors mounted on wing tip. Also, in the conversion mode, the rotor tilting angle is varying from 90 deg to 0 deg for forward flight. This type of aircraft shows various dynamic characteristics as the flight mode changes. Due to helicopter/airplane flight modes and conversion phase between them, it is very difficult to control with one fixed control structure. At the time of this writing, the 40% scaled Smart UAV is conducting flight tests for verification of the automatic flight control law designed using classical PID control algorithm.

When we design the flight control system using classical approach, the architecture has been performed in such a way of yielding linearized models about several trim flight conditions and designing linear controllers on these models, and then integrated the resulted controllers through gain scheduling scheme. However, the application of this approach to highly nonlinear and uncertain dynamics having multiple flight modes like a tiltrotor aircraft is much more tedious and time-consuming. Moreover, this approach may not globally satisfy the required performances and stability of the vehicle. In order to overcome these difficulties, various control techniques^{1,2} can be implemented on the flight control system. The most valuable candidate approach is adaptive nonlinear control technique using feedback linearization³⁻⁷. The main procedure of these approaches is to map a nonlinear system to a linear system through a coordinate transformation. However, this technique is quite limited, because it relies on exact knowledge of the plant dynamics. In other words, the dynamics covering the whole flight modes shows highly nonlinearities and contains uncertainties in control as well as state variables. Similar control strategies have been successfully applied to comprehensive simulations of an F/A-18 airplane¹ and an AH-64 helicopter⁴.

The main objective of this paper is to design the trajectory tracking controller with one fixed structure covering all flight mode of the Smart tiltrotor UAV. For this purpose, the inner-loop SCAS(Stability and Control Augmentation System) and the outer-loop trajectory tracking system are designed loop-by-loop based upon the time-scale separation and DMI(Dynamic Model Inversion) technique and adaptive control scheme using neural networks are used for both control-loop designs. The inner-loop DMI is derived using the angular equation of motion while the outer-loop DMI is derived using the translational equation of motion. Adaptive neural networks are applied to compensate for the inevitable model inversion error at both the inner-loop and the outer-loop DMIs, due to the lack of full knowledge of the tiltrotor UAV dynamics. The network update law is derived from a Lyapunov stability analysis based on tracking error and network performance. And the PCH algorithm is adopted to lessen the instability problem caused by the time-delay of the control loops, saturation and rate limit of actuators. In addition, the desired commands covering the conversion mode are derived with one relation as a function of rotor tile angle by mixing the desired commands for the helicopter and the airplane modes.

Section II contains a summary of the Smart UAV specifications and a description of its operational properties. Section III outlines the neural network based adaptive control methodology and section IV presents the implementation of the neural network augmented model inversion control architecture for the Smart UAV. Numerical results and evaluative remarks for the Smart UAV approach and landing scenario are included in section V, and the conclusion is in the final sections.

II. Smart UAV Dynamic Characteristics

The Smart UAV, tiltrotor research UAV is representative of those aircrafts which will employ the tiltrotor concept. The tiltrotor UAV has three flight modes, i.e. helicopter mode, conversion mode and airplane mode. The hover lift and cruise propulsive force are provided by disc loading rotors located at each wing tips. The rotor axes rotate from near vertical for helicopter flight to near horizontal for airplane flight. In the helicopter mode, control is provided by rotor-generated forces and moments while in the airplane flight mode, primary control is provided by conventional aerodynamic control surfaces.

The Smart UAV is capable of hovering for approximately one hour at sea-level starting at its design gross weight of 1,000 Kg. In the airplane mode, it can fly at speeds greater than 500 km/h. The detailed specification of the Smart UAV is given in Table 1 and Fig. 1 shows its drawing.

Table 1. Specification of the full scale Smart UAV

Design Gross Weight (Kg)	1000	Rotor Speed (hel /conve. mode)	1605 rpm
Empty Weight (Kg)	720	Rotor Speed (airplane mode)	1284 rpm
Wing Area (m ²)	3.2	Conversion Range (deg)	0~93
Wing Span (m)	4.0	Conversion Min. Time (sec)	< 60 sec
Length (m)	4.96	Max. Level Flight Speed (km/h)	500
Rotor Diameter (m)	2.866	Max Speed in Hel mode (km/h)	120
Disc Area per Rotor (m ²)	6.451	Range (km)	200

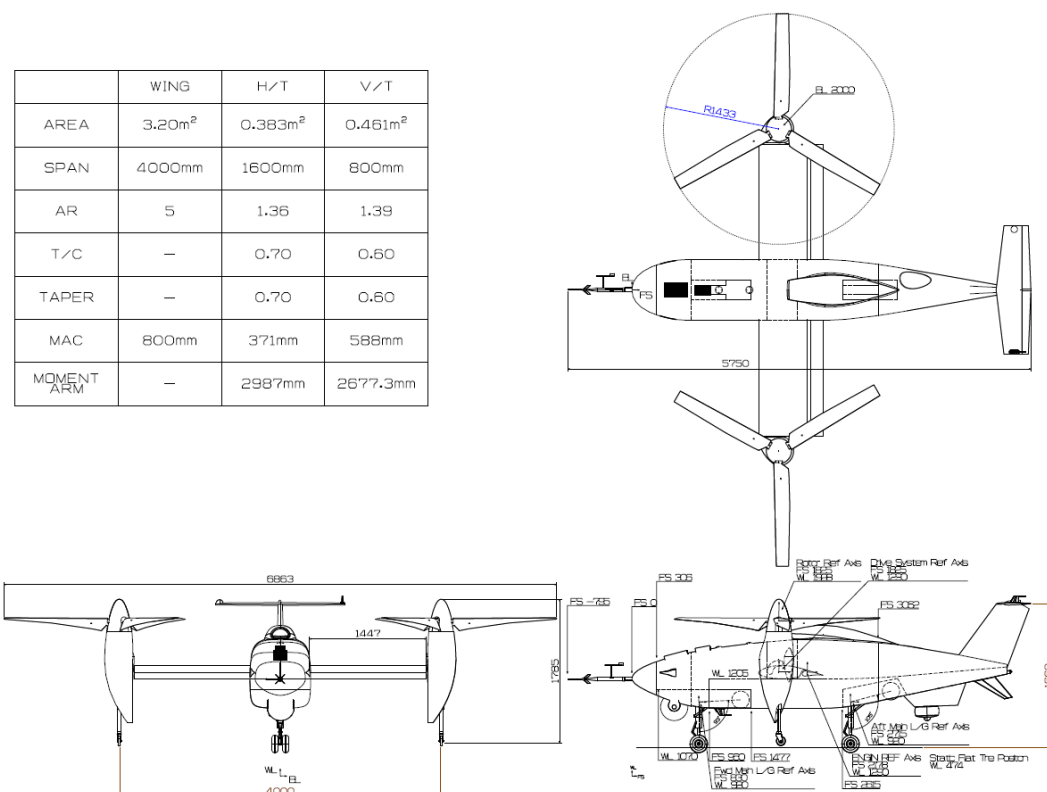
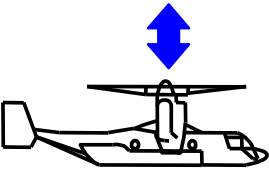
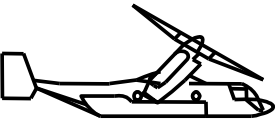
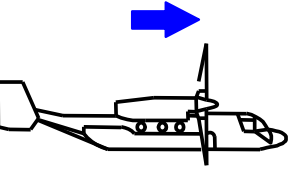


Figure 1. Geometric Configuration of the Smart UAV

As given in Table 2, in helicopter mode, pitching moment is generated by applying longitudinal cyclic pitch change to the rotor blades. This is accomplished by the pitch control input. Yawing moment is generated by the application of differential longitudinal cyclic pitch change to the rotors. This is accomplished by yaw control input. Rolling moment is generated by applying differential collective pitch change to the rotors. This is accomplished by the roll control input. Motion of the collective/power stick changes engine power and rotor collective pitch to provide vertical thrust control. During conversion from helicopter mode to airplane mode, the desired control response is achieved by phasing out the rotor controls as control surfaces become effective. Elevator and flaperons are activated respectively by the pitch and roll control inputs. The yaw control input in the smart UAV is connected

to differential collective pitch of rotors for yaw control instead of rudder. The power control consists of a pair of throttles and a collective pitch lever. In the helicopter and conversion modes, the collective/power lever motion simultaneously changes the engine throttles and the rotor collective pitch. In the airplane mode, the collective/power lever controls only the engine throttles as the collective pitch input is phased out as a function of rotor tilt angle. In addition, power management is simplified by the automatic inputs of a governor which adjusts collective pitch to maintain the pre-selected rotor rpm. The Smart UAV has a 'mixer system' taking care of all above functions of mixing control effectors for each flight mode. In addition, the mixer will adjust its airspeed and conversion angle simultaneously to stay within the conversion corridor limits as shown in Fig. 2.

Table 2. Control effectors of the Smart UAV at various flight modes

Mode	Helicopter mode	Conversion mode	Airplane mode
Rotor Position			
Tilting Angle	90 deg	$0 < \text{Nacelle Angle} < 90$	0 deg
Power Input	Throttle and collective pitch with beta governing	Mixed	Throttle with beta governing
Pitch Control	Longitudinal cyclic	Mixed	Elevator
Roll Control	Differential collective pitch	Mixed	Flaperon
Yaw Control	Differential longitudinal cyclic	Mixed	Differential collective pitch
Side Force Control	Lateral cyclic	-	-

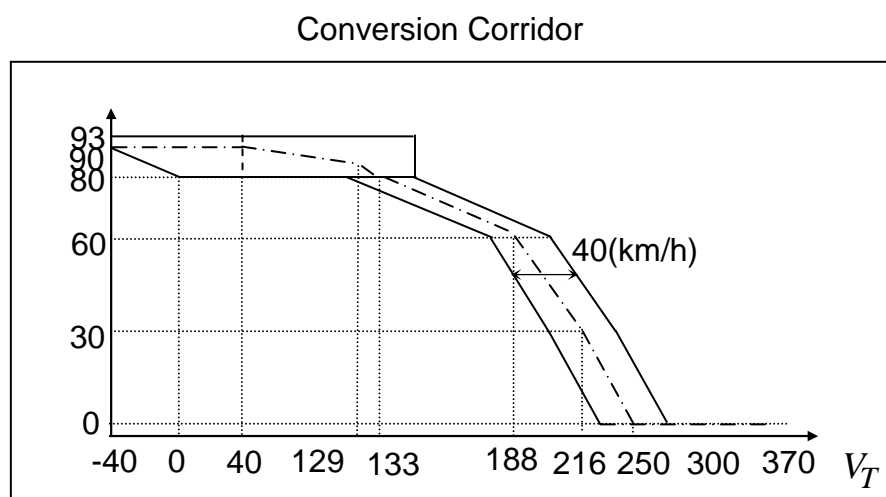


Figure 2. Smart UAV Conversion Corridor

III. Neural Network based Adaptive Control Methodology

The following discussion constitutes a brief overview of the dynamic inversion based adaptive scheme proposed in Ref. 3. It is included in order to introduce some necessary terminology and notation as well as to provide insight in the nature of the stability control law. For details pertaining to the stability analysis of the closed-loop system, the reader may consult of Ref. 3.

A. Dynamic Model Inversion

A nonlinear system of n degrees of freedom are given as

$$\ddot{x} = f(x, \dot{x}, \delta) \quad (1)$$

where $x(t) \in R^n$ and $\dot{x}(t) \in R^n$ are the state variables and $\delta(t) \in R^m$ is the control variable. The function $f(\cdot)$ is a mapping from a domain $R^n \times R^n \times R^m$ into R^n . If $n=m$, it is called a square system. If $f(x, \dot{x}, \cdot)$ is invertible and all states are measurable, Eq.(1) can be transformed to the form

$$\begin{aligned} \ddot{x} &= u \\ u &= f(x, \dot{x}, \delta) \end{aligned} \quad (2)$$

where $u(t) \in R^n$ is the pseudo-control variable. The inverse transformation of Eq.(2) is expressed by

$$\delta = f^{-1}(x, \dot{x}, u) \quad (3)$$

If the function f is perfectly known and its inverse is computed exactly, one may obtain the exact linearized system. In practice, however, the exact linearization can not be achievable due to uncertainties and unmodelled dynamics in mathematical modeling. Thus, the system which results from applying dynamic inversion may be expressed as follows:

$$\ddot{x} = u + \Delta'(x, \dot{x}, u) \quad (4)$$

where

$$\Delta'(x, \dot{x}, u) = f(x, \dot{x}, \hat{\delta}) - \hat{f}(x, \dot{x}, \hat{\delta}) \quad (5)$$

$\Delta' : R^n \times R^n \times R^n \rightarrow R^n$ is a mapping representation of the inversion error. In Eq.(5), $\hat{\delta} = \hat{f}^{-1}(x, \dot{x}, u)$ represents an approximate inversion mapping for the control input. Fig. 3 illustrates the approximate dynamic inversion process.

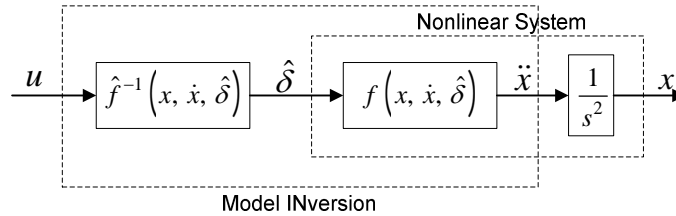


Figure 3. Approximate Dynamic Model Inversion

B. Adaptive Control Using Neural Network

In the ideal case, any linear controller which satisfies stability and performance objectives would be sufficient to control the linearized system. For simplicity, the linear controller considered here will be of the proportional-plus-derivative (PD) variety. When there is nonzero inversion error, however, the linear controller output is augmented by an adaptive control which is used to compensate for this error. These two components combine to yield the following expression for the pseudo-control

$$u_i(t) = u_{pd_i}(t) + \ddot{x}_{c_i}(t) - \hat{u}_{ad_i}(t), \quad i = 1, 2, \dots, n \quad (6)$$

where $i = 1, 2, \dots, n$ indicates each individual pseudo-control input channel, and u_{pd_i} and \hat{u}_{ad_i} represent a PD control and an adaptive control signals, respectively. Fig. 4 depicts the neural network based adaptive control architecture of the i^{th} degree of freedom. Note that each degree of freedom may have a separate neural network and that in general all the state variables and the all the pseudo-controls are inputs to each network.

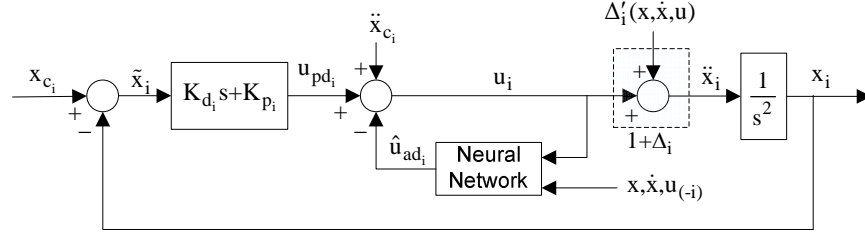


Figure 4. Neural Network based Adaptive Control Architecture

The error dynamics can be obtained from Eq.(6) and Eq.(4)

$$\ddot{\tilde{x}}_i + K_{d_i} \dot{\tilde{x}}_i + K_{p_i} \tilde{x}_i = \hat{u}_{ad_i} - \Delta'_i(x, \dot{x}, u), \quad i = 1, 2, \dots, n$$

In state space form, we have

$$\dot{e}_i = A_i e_i + b(\hat{u}_{ad_i} - \Delta'_i(x, \dot{x}, u)) \quad (7)$$

where $\tilde{x}_i = x_{c_i} - x_i$, and $e_i^T = [\tilde{x}_i \quad \dot{\tilde{x}}_i]$, and

$$A_i = \begin{bmatrix} 0 & 1 \\ -K_{p_i} & -K_{d_i} \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (8)$$

The adaptive term \hat{u}_{ad_i} in Eq.(7) is used to compensate for the inversion error Δ'_i , so that the steady state error of $\tilde{x}_i(t)$ becomes zero. And it is noted that the PD-control gains are chosen to satisfy the flying quality requirements for the corresponding aircraft. In Ref. 3, the adaptive control signal of neural network is proposed as

$$\hat{u}_{ad_i}(t) = \sum_{j=1}^N \hat{w}_{ij}(t) \beta'_{ij}(x, \dot{x}, u) = \hat{w}_i^T(t) \beta'_i(x, \dot{x}, u) \quad (9)$$

In Eq.(9), \hat{w}_i indicates the vector of the weight $\hat{w}_{ij}(t)$ that is updated online, and $\beta'_{ij}(\cdot)$ is a basis function of the neural network. This approximation to the inversion error may, in general, be realized by an adaptive neural network. It is further shown in Ref. 3 that the adaptive control achieves global asymptotic stability, that is, the error dynamics expressed in Eq. (7) is bounded in the sense of Lyapunov stability when the weight update rule is chosen as follows:

$$\dot{\hat{w}}_i(t) = \begin{cases} -\gamma_i e_i^T P_i b \beta'_i(x, \dot{x}, u) & \text{for } \|e_i\|_{P_i} > e_{0_i} \\ 0 & \text{for } \|e_i\|_{P_i} \leq e_{0_i} \end{cases}, \quad i = 1, 2, \dots, n \quad (10)$$

where $\|e_i\|_{P_i} = \sqrt{e_i^T P_i e_i}$. The positive constant γ_i is an adaptation gain and P_i is a symmetric, positive definite matrix satisfying the Lyapunov equation of $P_i A_i + A_i^T P_i = -I$. The existence of a unique $P_i = P_i^T > 0$ is assured by the Hurwitz matrix A_i of Eq.(8). The description of P_i is given by

$$P_i = \begin{bmatrix} \frac{K_{d_i}}{2K_{p_i}} + \frac{K_{p_i}}{2K_{d_i}} \left(1 + \frac{1}{K_{p_i}}\right) & \frac{1}{2K_{p_i}} \\ \frac{1}{2K_{p_i}} & \frac{1}{2K_{d_i}} \left(1 + \frac{1}{K_{p_i}}\right) \end{bmatrix} \quad (11)$$

Note from Fig. 2 that the network input, \hat{u}_{ad_i} , depends on the current network output. Thus, a critical assumption in the stability proof involves the existence of a fixed-point solution for the output \hat{u}_{ad_i} . A simple iterative scheme was employed here to compute the network output. Also, the deadzone is required to account for the fact that the network is incapable of exactly representing the inversion error (Δ') by employing a finite set of basis functions. A deadzone was not used in this design, however, in order to emphasize the effect that the choice of basis functions has on performance.

C. Pseudo Control Hedging

Pseudo-Control Hedging (PCH) is introduced to protect the adaptive law from effects due to actuator rate and position limits, unmodeled actuator dynamics and to protect the adaptive process when it is not in control of the plant⁸. The main idea behind the PCH methodology is to modify the reference command in order to prevent the adaptive element from adapting to these actuator characteristics. This is commonly done by generating the command using a reference model for the desired response. The reference model is 'hedged' by an amount equal to the difference between the commanded and an estimate for the achieved pseudo-control. PCH has been proven to be the most effective and powerful method in resolving such problems⁸.

That is, this dynamic inversion element is designed without consideration of actuator saturation. This command (δ_{cmd}) will not necessarily equal the actual control (δ) due to actuator limitation. To get the PCH signal (u_h), an estimated actuator position ($\hat{\delta}$) is determined based on a model or measurement. This estimate is then used to get the difference between commanded pseudo-control and the estimated-actual pseudo-control

$$u_h = \hat{f}(x, \dot{x}, \delta_{cmd}) - \hat{f}(x, \dot{x}, \hat{\delta}) = u - \hat{u} \quad (12)$$

where u_h is also simply described as a difference of the pseudo inputs and the achieved signal through actuators. Fig.5 depicts the schematic diagram of the DMI with PCH compensation.

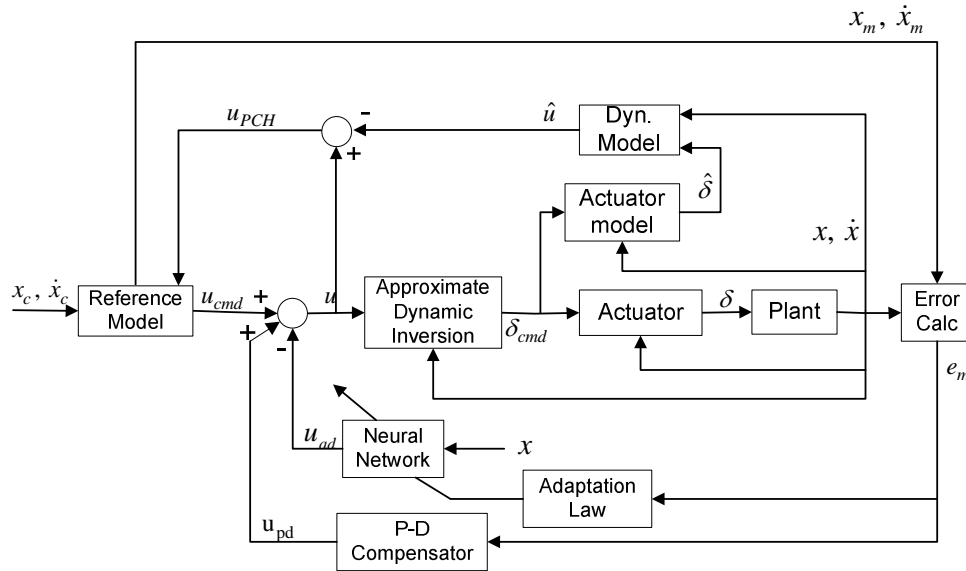


Figure 5. Dynamic model inversion with PCH compensation

The reference model update is

$$\ddot{x}_{rm} = u_{crm} (x_{rm}, \dot{x}_{rm}, x_c, \dot{x}_c) - u_h \quad (13)$$

where x_c, \dot{x}_c are the external commands. This choice of reference model update will remove the actuator characteristics from reference model tracking error. The instantaneous pseudo-control output of the reference model in the feedback path is u_{crm} . The detailed discussion about the theory associated with this application of PCH can be referred to Ref. 8.

IV. Flight Control Application for Smart UAV

The flight control system for the Smart UAV has two time-scale separation architecture. It is composed of inner-loop SCAS(Stability and Command Augmentation System) as faster dynamics and outer-loop trajectory tracking controller as slower dynamics. Both controllers are implemented using dynamic model inversion based adaptive control technique with neural networks, mentioned in section III. Bandwidth separation of the command filter and the adaptation dynamics allows for implementation of time scale separation assumption and design requirements.

A. Stability and Command Augmentation System Design

The implemented response type of the SCAS is an ACAH(Attitude Command Attitude Hold) type. The inner-loop variables to be controlled are the roll, pitch and yaw attitude angles. And the output variables of the controller are the control inputs of the Smart UAV, that is, longitudinal stick input (δ_{LN}), lateral stick input (δ_{LT}), pedal input (δ_{PD}). In order to achieve input-output linearization for the angular motion, the second time derivatives of the Euler angles to be controlled are equated to three pseudo-control variables. These results in

$$u_\phi = \ddot{\Phi}, \quad u_\theta = \ddot{\Theta}, \quad u_\psi = \ddot{\Psi} \quad x = [\Phi \ \Theta \ \Psi] \quad (13)$$

The actual controls required for achieving approximated tracking of the command variables are computed using a angular moment equation as shown as

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \hat{A} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} + \hat{B} \begin{bmatrix} \delta_{LN} \\ \delta_{LT} \\ \delta_{PD} \end{bmatrix} \quad (14)$$

where matrices \hat{A} and \hat{B} are dynamic matrix and control matrix, respectively. The inversion for calculating is performed by solving the moment equation of Eq.(14) as follows.

$$\begin{bmatrix} \delta_{LN} \\ \delta_{LT} \\ \delta_{PD} \end{bmatrix} = \hat{B}^{-1} \left(\begin{bmatrix} \dot{p}_c \\ \dot{q}_c \\ \dot{r}_c \end{bmatrix} - \hat{A} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \right) \quad (15)$$

where

$$\begin{aligned} \dot{p}_c &= u_\phi - u_\psi \sin \Theta - \dot{\Psi} \dot{\Theta} \cos \Theta \\ \dot{q}_c &= u_\theta \cos \Phi - \dot{\Theta} \dot{\Phi} \sin \Phi + u_\psi \sin \Phi \cos \Theta + \dot{\Psi} \dot{\Phi} \cos \Phi \cos \Theta - \dot{\Psi} \dot{\Theta} \sin \Phi \sin \Theta \\ \dot{r}_c &= -u_\theta \sin \Phi - \dot{\Theta} \dot{\Phi} \cos \Phi + u_\psi \cos \Phi \cos \Theta + \dot{\Psi} \dot{\Phi} \sin \Phi \cos \Theta - \dot{\Psi} \dot{\Theta} \cos \Phi \sin \Theta \end{aligned} \quad (16)$$

In Eqs.(15) and (16) the little letters among Euler angle symbols mean perturbed variables around a trim condition and the capital ones mean original nonlinear variables.

In this design, single model linearized on a nominal operating point is used in the inner-loop DMI for simplicity of the controller design process. Adaptive neural network makes it possible to use a linear model in DMI process covering the whole flight envelope. For example, one candidate of the linear models can be the linear model at the condition of velocity of 20 km/h with helicopter configuration (tilting angle of 90 deg).

$$\hat{A} = \begin{bmatrix} 0.0129 & -0.0001 & -0.0243 & -0.0000 & -0.1941 & -0.0000 \\ 0.0000 & -0.0140 & -0.0002 & -2.5912 & 0.0000 & 0.3851 \\ -0.0000 & 0.0201 & 0.0002 & 0.3835 & 0.0000 & -0.0267 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} -2.8766 & -0.0000 & -0.0000 \\ -0.0000 & 6.0824 & -0.0133 \\ -0.0000 & -1.0727 & 1.8394 \end{bmatrix}$$

The Proportional plus derivative control laws is used to shape the attitude response of each channel as follows:

$$\begin{aligned} u_\phi &= K_{p_\phi} (\Phi_c - \Phi) + K_{d_\phi} (\dot{\Phi}_c - \dot{\Phi}) + \ddot{\Phi}_c \\ u_\theta &= K_{p_\theta} (\Theta_c - \Theta) + K_{d_\theta} (\dot{\Theta}_c - \dot{\Theta}) + \ddot{\Theta}_c \\ u_\psi &= K_{p_\psi} (\Psi_c - \Psi) + K_{d_\psi} (\dot{\Psi}_c - \dot{\Psi}) + \ddot{\Psi}_c \end{aligned} \quad (17)$$

where, $K_{p_\phi}, K_{d_\phi}; K_{p_\theta}, K_{d_\theta}; K_{p_\psi}, K_{d_\psi}$ are proportional and derivative control gains for each channel. The PD gains were specified to satisfy the system specifications. The values of the proportional and derivative gains were chosen for roll and pitch channels as 25 and 10, respectively. These values correspond to inner-loop settling times of 0.9 sec and damping ratios of 1. For yaw channel 9 and 6 were chosen, resulting in 1% settling time of 1.5 sec and damping ratio of 1. In the outer-loop, the proportional gains were chosen for Ue and Ve channels as 2. For altitude channel proportional gain, 3 and derivative gain, 3.46 were chosen.

The inversion process is not perfect because the inversion is based on the linear model at an arbitrary single flight condition in Eq.(15). Furthermore, modeling uncertainties and nonlinearities due to various flight modes cause the inversion performance to be further degraded. Thus, a neural network is introduced in series with the DMI to compensate on-line for the inversion error during flight, according to the direct adaptive tracking architecture mentioned in Section III. The adaptive control signal of the neural network is chosen as

$$\hat{u}_{adi}(t) = \sum_{j=1}^N \hat{w}_{ij}(t) \beta'_{ij}(x, \dot{x}, u) = \hat{w}_i^T(t) \beta'_i(V_p, U, V, W, P, Q, R, \Phi, \Theta, \Psi, u_i), \quad i = \Phi, \Theta, \Psi \quad (18)$$

In this design, Sigma-Pi Neural Network(SPNN) is used for adaptation and the structure of the neural network is shown in Fig. 4. In each neural network, 42 sigma-pi units were employed to build the basis functions, β_{ij} 's. Thus, the total number of weights for each channel is 42 in inner-loop and 45 in outer-loop. The adaptation gain, γ_i was selected as 1.4 in all channels of inner-loop and 0.3, 0.15, 0.1 in each channel of outer-loop. The initial values of the weights for integration of Eq.(10) were set to zeros.

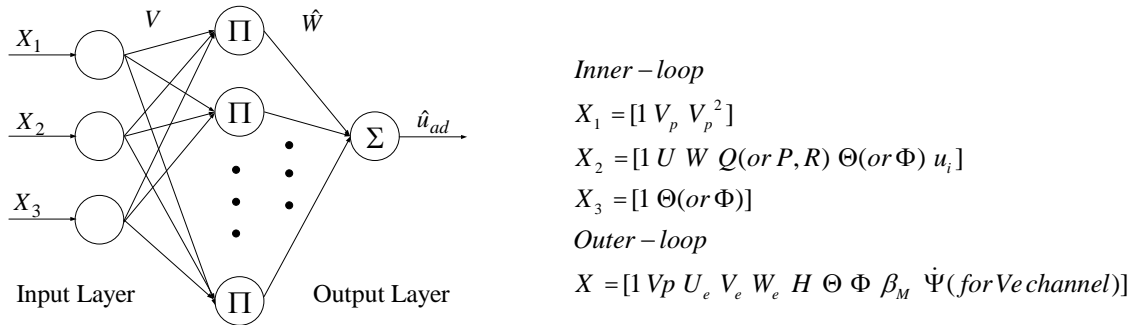


Figure 6. Structure of the on-line adaptive Sigma-Pi neural network

B. Trajectory Tracking Controller Design

The dynamic model inversion for the outer-loop trajectory tracking control is derived so as to cover all flight modes of the tiltrotor UAV. The input variables to be controlled in the outer-loop controller are the inertial positions, X (North), Y (East), Z (Down) and the output variables of the controller are the roll(Φ_c)and pitch(Θ_c) attitude command angles, and the collective control input to the main rotor δ_{COL} . We introduce the second time derivatives of X, Y, Z which can be equated to pseudo-control variables as follows:

$$\begin{bmatrix} \ddot{X}_e \\ \ddot{Y}_e \\ \ddot{Z}_e \end{bmatrix} = L_{IB}(\Phi, \Theta, \Psi) \begin{bmatrix} F_x / m \\ F_y / m \\ F_z / m \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (19)$$

$$U_x = \ddot{X}_e, U_y = \ddot{Y}_e, U_z = \ddot{Z}_e \quad (20)$$

where F_x, F_y, F_z are applied force components along X, Y, Z axis of the aircraft body-fixed frame, respectively. And L_{IB} is the coordinate transformation matrix from body-fixed to inertial coordinate system. Transforming Eq.(20) into the body-fixed frame results in

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = L_{IB}^{-1} \begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} = \begin{bmatrix} F_x / m \\ F_y / m \\ F_z / m \end{bmatrix} + \begin{bmatrix} -g \sin \Theta \\ g \sin \Phi \cos \Theta \\ g \cos \Phi \cos \Theta \end{bmatrix} \quad (21)$$

where V_x, V_y, V_z are pseudo-control variables expressed in the body-fixed frame. Once more, transforming Eq.(21) into the rotor-fixed frame results in

$$\begin{bmatrix} W_x \\ W_y \\ W_z \end{bmatrix} = L_2(\beta_M) \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = L_2(\beta_M) \left(\begin{bmatrix} F_x / m \\ F_y / m \\ F_z / m \end{bmatrix} + \begin{bmatrix} -g \sin \Theta \\ g \sin \Phi \cos \Theta \\ g \cos \Phi \cos \Theta \end{bmatrix} \right) \quad (22)$$

where

$$\begin{bmatrix} F_{x_R} \\ F_{y_R} \\ F_{z_R} \end{bmatrix} = \begin{bmatrix} F_x \cos \beta_M - F_z \sin \beta_M \\ F_y \\ F_x \sin \beta_M + F_z \cos \beta_M \end{bmatrix} \quad (23)$$

and $L_2(\beta_M)$ is a coordinate transformation matrix from body-fixed to rotor-fixed coordinate system by tilting angle, β_M . Eq.(22) can be rewritten using Eq.(23).

$$W_x = \left(\frac{F_x}{m} \right) \cos \beta_M - \left(\frac{F_z}{m} \right) \sin \beta_M - g \sin \Theta \cos \beta_M - g \cos \Phi \cos \Theta \sin \beta_M \quad (24)$$

$$W_y = \frac{F_y}{m} + g \sin \Phi \cos \Theta \quad (25)$$

$$W_z = \left(\frac{F_x}{m} \right) \sin \beta_M + \left(\frac{F_z}{m} \right) \cos \beta_M - g \sin \Theta \sin \beta_M + g \cos \Phi \cos \Theta \cos \beta_M \quad (26)$$

The specific force(aerodynamic and thrust forces) component along the X_B -axis can be expressed as

$$\frac{F_x}{m} = \frac{F_{x_0}}{m} + \frac{\Delta F_x}{m} \approx \frac{F_{x_0}}{m} + X_\alpha \alpha + X_{\delta c} \delta c = \frac{F_{x_0}}{m} + X_\alpha (\theta - \gamma) + X_{\delta c} \delta c \quad (27)$$

The specific force component along the Z_B -axis is expressed as

$$\frac{F_z}{m} = \frac{F_{z_0}}{m} + \frac{\Delta F_z}{m} \approx \frac{F_{z_0}}{m} + Z_\alpha \alpha + Z_{\delta_c} \delta_c = \frac{F_{z_0}}{m} + Z_\alpha (\theta - \gamma) + Z_{\delta_c} \delta_c \quad (28)$$

where X_{δ_c} and Z_{δ_c} are the control derivatives due to rotor thrust along the X_B -axis and Z_B -axis, respectively. After substituting Eq.(27) and Eq.(28) into Eq.(26), we can linearize Eq.(26) to obtain the desired pitch command.

$$\Delta W_z (= W_z) = \{X_\alpha (\theta - \gamma) + X_{\delta_c} \Delta \delta_c\} \sin \beta_M + \{Z_\alpha (\theta - \gamma) + Z_{\delta_c} \Delta \delta_c\} \cos \beta_M - g \theta \sin \beta_M \cos \Theta_0 - g \theta \cos \beta_M \cos \phi \sin \Theta_0 \quad (29)$$

Terms containing the flight path angle can be considered as uncertainties. We neglected in this design. With this assumption, from Eq.(29) we can calculate the desired pitch angle as follows:

$$\theta_{des} = \frac{W_z - (X_{\delta_c} \sin \beta_M + Z_{\delta_c} \cos \beta_M) \Delta \delta_{c_{des}}}{X_\alpha \sin \beta_M + Z_\alpha \cos \beta_M - g \sin \beta_M \cos \Theta_0 - g \cos \beta_M \cos \phi_{des} \sin \Theta_0} \quad (30)$$

In Eq.(24), forces can be considered as the function of thrust and aerodynamic. However, in the inversion process, the aerodynamic forces are neglected such that these are considered as inversion errors. With this assumption the desired thrust command covering for all flight modes can be derived. We can linearize Eq.(24) to obtain the desired collective command

$$\left(\frac{\Delta F_x}{m} \right) \cos \beta_M - \left(\frac{\Delta F_z}{m} \right) \sin \beta_M = \Delta W_x (= W_x) + g \theta \cos \Theta_0 \cos \beta_M - g \theta \sin \Theta_0 \sin \beta_M \quad (31)$$

With a previous assumption, the right terms of Eq.(31) can be expressed as

$$\left(\frac{\Delta F_x}{m} \right) \cos \beta_M - \left(\frac{\Delta F_z}{m} \right) \sin \beta_M \approx \{X_\alpha (\theta - \gamma) + X_{\delta_c} \Delta \delta_c\} \cos \beta_M - \{Z_\alpha (\theta - \gamma) + Z_{\delta_c} \Delta \delta_c\} \sin \beta_M \quad (32)$$

In this design, the control derivative due to rotor thrust calculated in the airplane mode, X_{δ_r} was used as X_{δ_c} . And $Z_{\delta_{col}}$ in the helicopter mode was used as Z_{δ_c} . After substituting Eq.(32) into Eq.(31), we can calculate the desired thrust control input as follows:

$$\Delta \delta_{c_{des}} = \frac{W_x + \{(g \cos \Theta_0 - X_\alpha) \cos \beta_M - (g \cos \phi_{des} \sin \Theta_0 - Z_\alpha) \sin \beta_M\} \theta_{des}}{X_{\delta_r} \cos \beta_M - Z_{\delta_{col}} \sin \beta_M} \quad (33)$$

Using Eq.(25), the desired roll angle command becomes

$$\Phi_{des} = \sin^{-1} \left(\frac{W_y}{g \cos \Theta_{des}} \right) \quad (34)$$

The desired heading angle command is generated using turn coordination logic.

$$\dot{\Psi}_{des} = \frac{g}{V_p} \tan \Phi_{des} \quad (35)$$

For helicopter mode ($\beta_M = 90$ deg), θ and $\Delta \delta_c$ become

$$\theta_{des} = -\frac{W_z (= V_x) - X_{\delta_c} \Delta \delta_{c_{des}}}{X_\alpha - g \cos \phi_{des} \cos \Theta_0}, \quad \Delta \delta_{c_{des}} = \frac{W_x (= -V_z) - (g \cos \phi_{des} \sin \Theta_0 - Z_\alpha) \theta_{des}}{-Z_{\delta_{col}}} \quad (36)$$

Similarly, for airplane mode ($\beta_M = 0$ deg), θ and $\Delta\delta_c$ become

$$\theta_{des} = \frac{W_z (=V_z) - Z_{\delta_c} \Delta\delta_{c_{des}}}{Z_\alpha - g \cos \phi_{des} \sin \Theta_0}, \quad \Delta\delta_{c_{des}} = \frac{W_x (=V_x) + (g \cos \Theta_0 - X_\alpha) \theta_{des}}{X_{\delta_T}} \quad (37)$$

Fig. 7 shows a block diagram of the adaptive trajectory tracking controller for the Smart UAV.

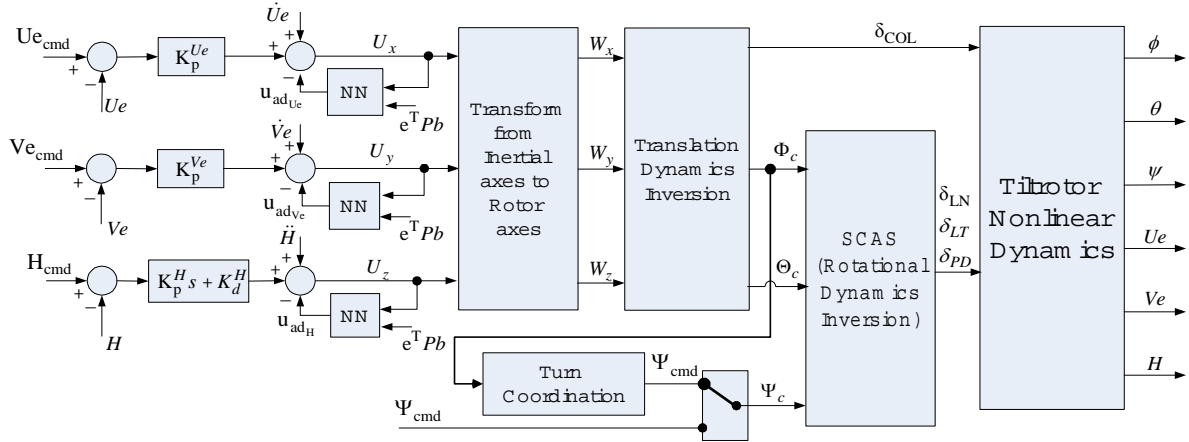


Figure 7. Adaptive Trajectory Tracking Controller for Smart UAV

V. Simulation Study

The trajectory tracking controller including inner-loop SCAS is evaluated with approach and landing scenario covering all flight modes of the Smart UAV, using a sophisticated 6DOF nonlinear simulation program.

A. Performance Evaluation of Inner-loop SCAS

Table 3 shows performance of SCAS with various combinations of DMI models and simulation initial conditions. Three linearized model used in DMI process and three initial conditions for simulation were selected for evaluation such as (a) 0 km/h with 90 deg tilt angle (helicopter mode), (b) 115 km/h with 30 deg tilt angle (transition mode), and (c) 250 km/h with 0 deg tilt angle (airplane mode). All three channels of roll, pitch, and yaw were evaluated. A sinusoidal command input with amplitude of 1 deg and frequency of 1 rad/sec is given to each channel and amplitude of 5 deg and frequency of 1 rad/sec is given to roll and yaw channel. The reason why we chose small amplitude of command input is to see the efficiency of the adaptive neural network only on model inversion error without nonlinearity effect of the dynamics. The symbols of O and □ mean 'perfect tracking' and 'existence of slight tracking error', respectively. From the results of Table 3, we could choose the linear model perturbed at hovering condition with helicopter configuration as a unique DMI model for all flight modes. The time response shown in Fig. 8 corresponds to the cases of shaded boxes in Table 3. The case of Fig. 8(a) does not have inversion error, while the case of Fig. 8(b) and (c) contains significant inversion. However, all cases in Fig. 8 show the perfect command tracking after 2 or 3 cycles in the presence of significant inversion error due to inconsistent flight configuration mode with the DMI model.

B. Reference Trajectory for Automatic Landing of the Smart UAV

The approach and landing scenario of the Smart UAV is one of the challenging cases to verify the trajectory tracking controller design since it contains multiple flight modes of tiltrotor, i.e., airplane, conversion, and helicopter modes. Thus, the trajectory tracking controller proposed in section IV, was evaluated with the approach and landing scenario of the Smart UAV. Fig. 10 depicts the approach and landing scenario of the 40% scaled Smart UAV. The reference trajectory of the scenario is according to the Smart UAV conversion corridor given in Fig. 2. This scenario starts at the forward speed of 164 fps and altitude of 984 ft with airplane configuration and steadily decreases the speed and altitude to 137 fps speed and 656 m altitude at the same time. From this point, the Smart UAV enters into the conversion mode. And then, the nacelle angle is tilted gradually from 0 deg to 80 deg, scheduled with forward speed mode while descending to speed of 80 fps. In the helicopter mode, the forward speed is decreasing to 0 fps with altitude of 16 ft. Finally, the Smart UAV bobs down to ground for touch down.

Table 3. Simulation results of the inner-loop SCAS

Linear Model used in DMI	Flight Condition	Input Command		
		θ (deg)	φ (deg)	ψ (deg)
V=0, $\beta_M=90^\circ$ (Helicopter Mode)	V=0, $\beta_M=90^\circ$	O	O	O
	V=115, $\beta_M=30^\circ$	O	O	O
	V=250, $\beta_M=0^\circ$	O	O	O
V=115, $\beta_M=30^\circ$ (Conversion Mode)	V=0, $\beta_M=90^\circ$	O	O	O
	V=115, $\beta_M=30^\circ$	O	O	O
	V=250, $\beta_M=0^\circ$	O	O	O
V=250, $\beta_M=0^\circ$ (Airplane mode)	V=0, $\beta_M=90^\circ$	□	□	O
	V=115, $\beta_M=30^\circ$	O	O	□
	V=250, $\beta_M=0^\circ$	O	O	O

V : Airspeed(km/h), β_M : Tilting angle, Command Tracking Results O : Excellent, Δ : good, \times : Bad

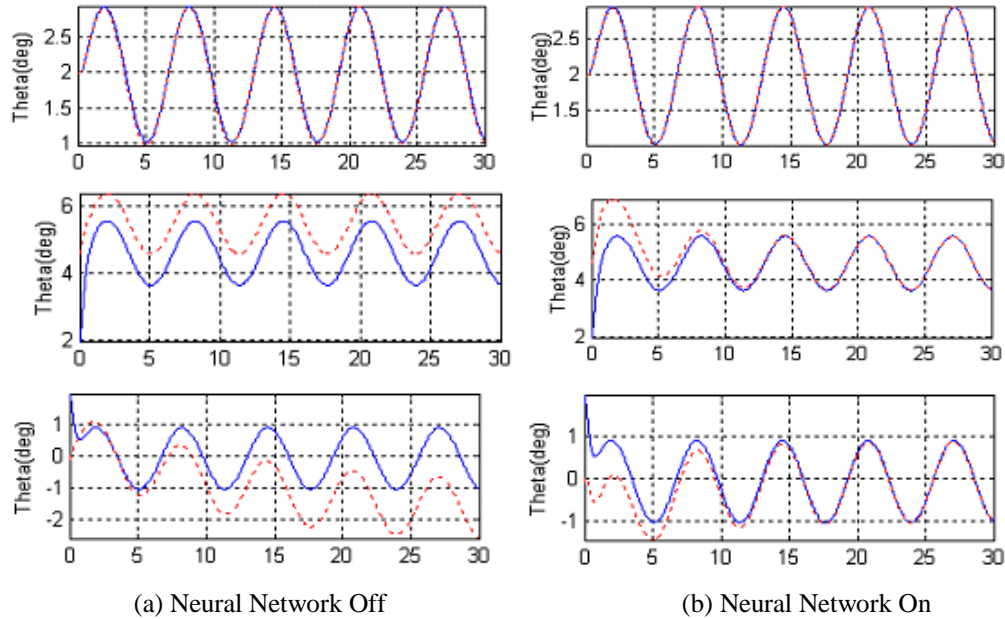


Figure 8. Simulation results of the inner-loop SCAS for pitch angle command

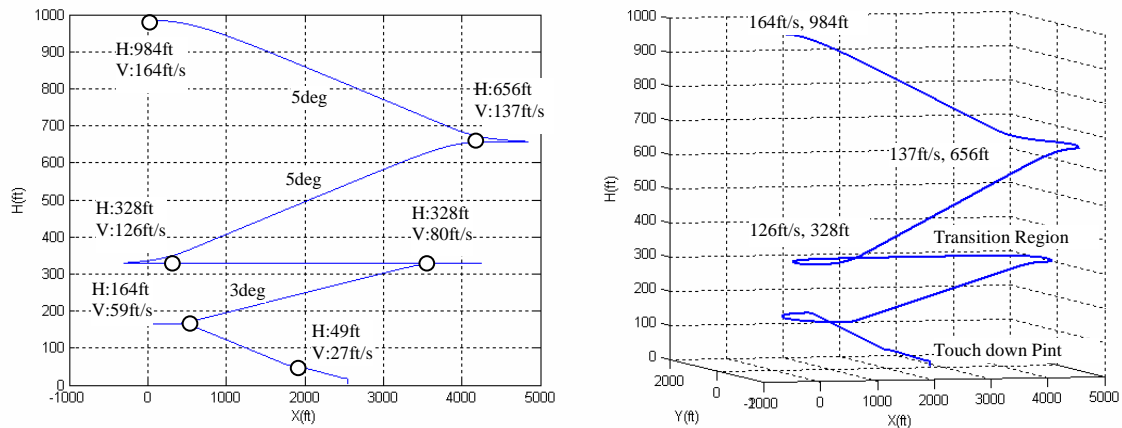


Figure 9. Approach and landing scenario for Smart UAV

C. Performance Evaluation of the trajectory tracking controller during approach and landing

For this simulation, a linear model at hover condition with helicopter configuration is chosen as the dynamic model used in inner-loop DMI inversion for inner-loop SCAS. The adaptation of the neural networks in inner-loop system was always turned on. The predetermined approach and landing reference trajectory command as shown in Figs. 9 was used for evaluation of the outer-loop performance. Figs. 10 and Fig. 11 show the tracking performance of the outer-loop controller and inner-loop SCAS, respectively. Fig. 12 and Fig. 13 represent the time responses of state variables and control variables, respectively. Fig. 14 shows the three dimensional flight trajectories when NN on and off. In the time span, the airplane mode is maintained from start to 100 sec, conversion mode from 100 sec to 125 sec, helicopter mode from 125sec to final time. The first and second rows of Fig. 10 represent the reference command and corresponding responses of the inertial velocity components (U_e , V_e) on the horizontal plane while the third row represents altitude command and time history. The outer-loop NN 'on' case shows better tracking performance, compared with NN 'off' case, but not much difference except during the conversion mode. However, from Fig. 14 we can see a significant difference in the position tracking performance. The maximum tracking errors of U_e during conversion mode are about 40 fps and 10 fps with NN off and on, respectively. The tracking performance of the inner-loop SCAS is almost perfect as shown in Fig. 11. From Fig. 13, we can see the control inputs staying in a reasonable range during this approach and landing. There is something to be investigated. The first one is the high angle of attack response occurred during level turn decreasing forward speed in airplane mode. The second one is the increasing pattern of the angle of attack together with pitch attitude during level flight changing nacelle angle from 0 deg to 90 deg. The third one is the oscillation in roll channel at entering and exiting point at around 125 sec and 170 sec in helicopter mode.

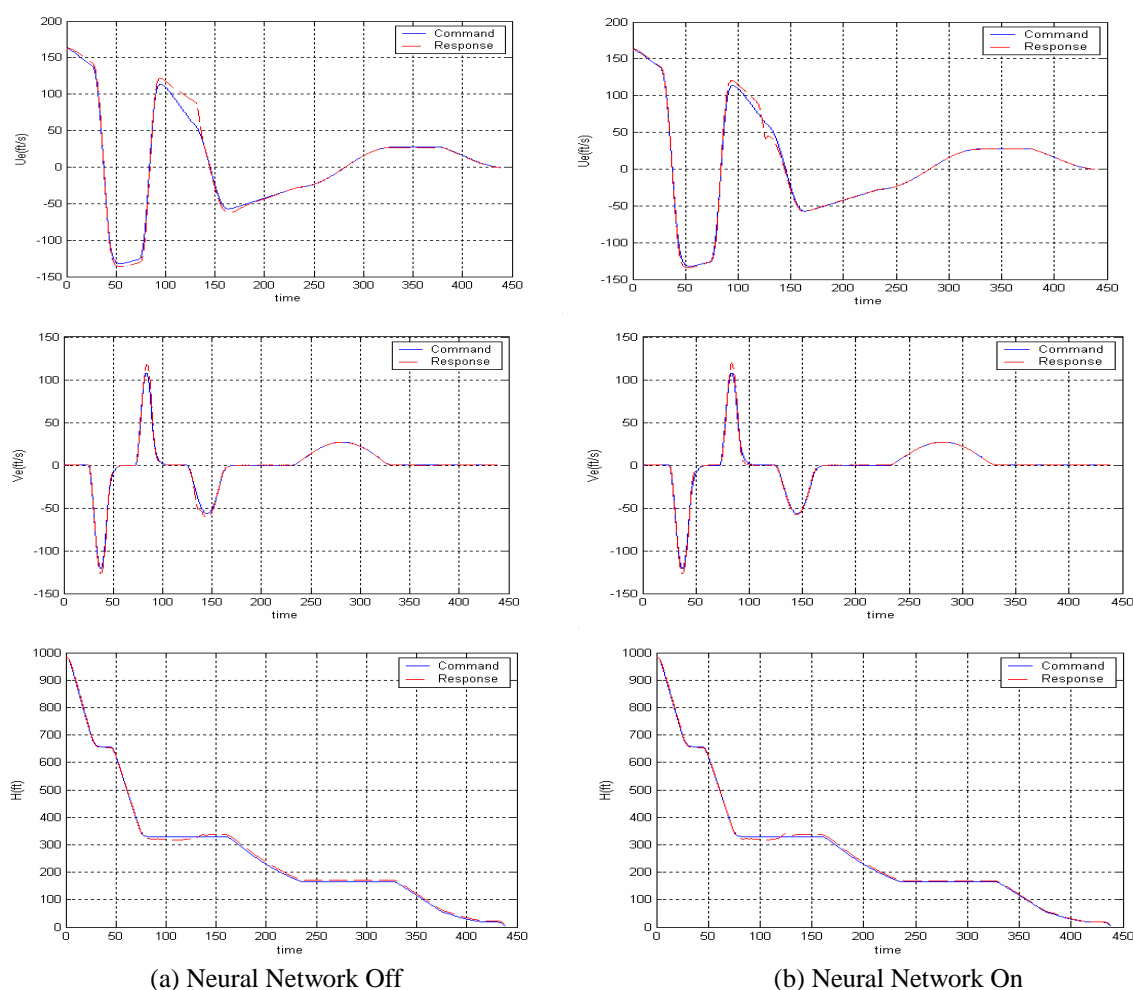


Figure 10. Command tracking performance of outer-loop controller during approach and landing

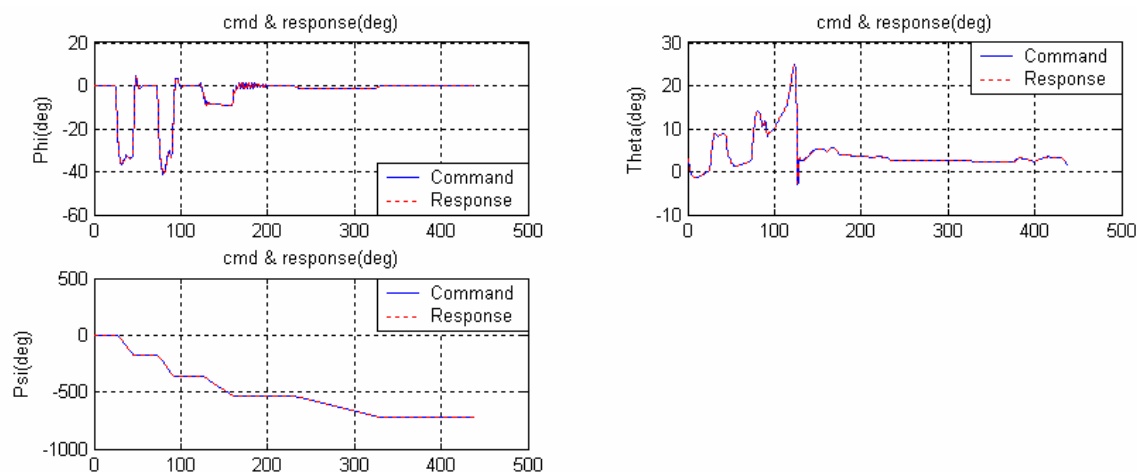


Figure 11. Command tracking performance of inner-loop controller with NN on

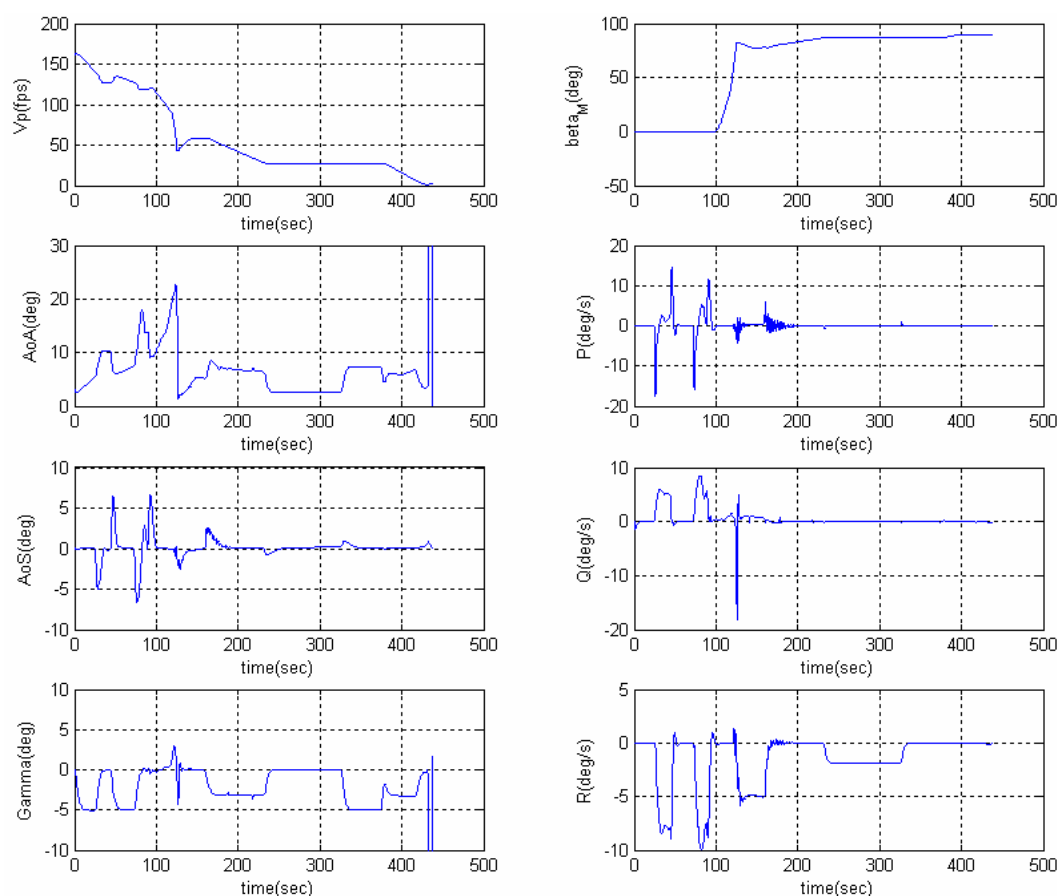


Figure 12. Response of state variables during landing of Smart UAV

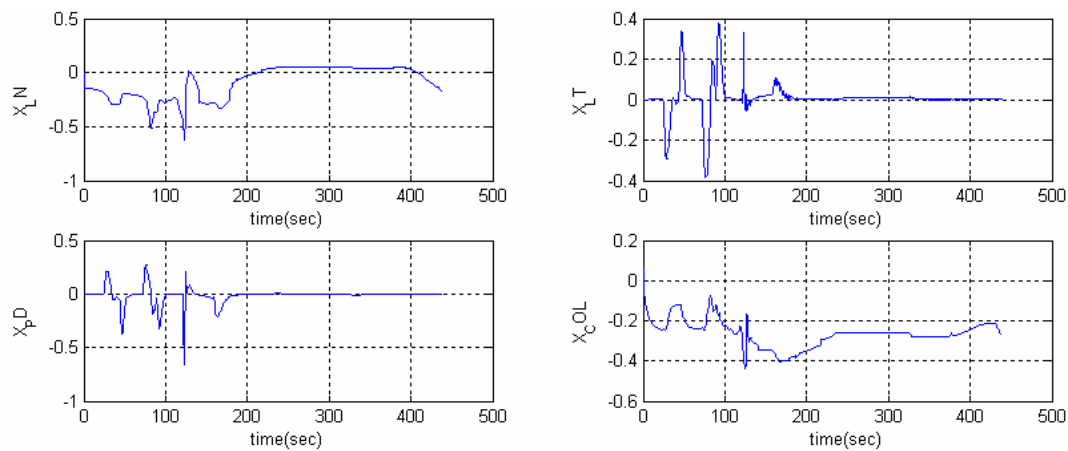
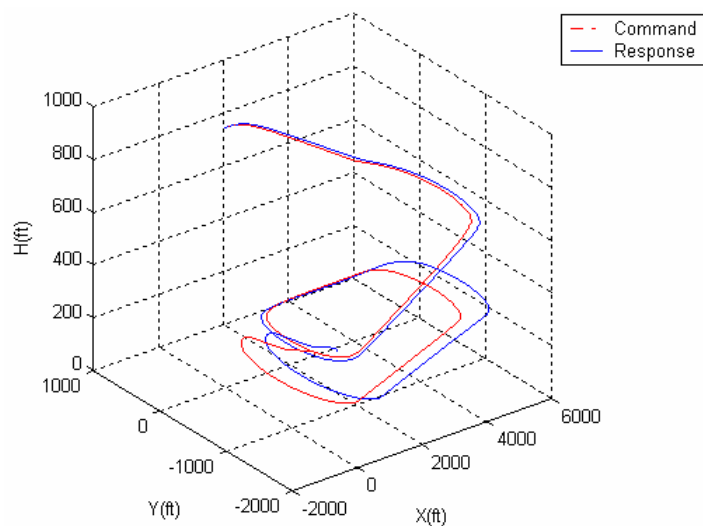
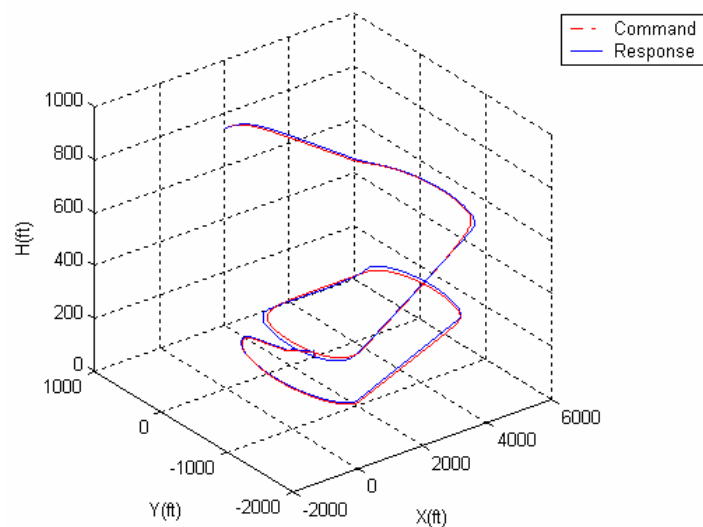


Figure 13. Control input responses during landing of Smart UAV



(a) Neural Network Off



(b) Neural Network On

Figure 14. 3-D flight trajectory during approach and landing of Smart UAV

VI. Conclusion

A full envelope nonlinear trajectory tracking controller covering all flight modes of a tiltrotor UAV has been designed using a dynamic model inversion technique integrated with adaptation of neural networks. An inner-loop/outer-loop architecture is employed based on two time-scale separation. An approximate dynamic model inversion for the outer-loop covering all flight modes was formulated within the framework of neural network based adaptive control. This paper illustrates a practical application with approach and landing scenario of Smart UAV within a realistic environment. First, inner-loop SCAS tests were executed with unmodeled unknown dynamics due to configuration change. And then the controller was applied to the 40% scaled Smart UAV to follow a three dimensional automatic approach and landing mission. Simulation results show that the controller especially displays good command following performance over the 3-D trajectory. However, a degraded tracking performance was shown in conversion mode, which needs to be investigated to refine the outer-loop DMI relating to conversion mode. The designed nonlinear controller demonstrates significant robustness and excellent performance characteristics within a highly nonlinear and uncertain environment. From these results, it is concluded that the formulation of outer-loop DMI integrated with online adaptive neural networks suggested in this paper is very effective in dealing with performance degradation problem of the trajectory following caused by insufficient information of system dynamics like a tiltrotor aircraft.

Acknowledgments

This paper was performed for the Smart UAV Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea and this work was partly supported by the Brain Korea 21(BK21) Project and the Program of New University for Regional Innovation (NURI).

References

- ¹. R.C. Baker and B. Charlie, "Nonlinear Unstable Systems," *International Journal of Control*, Vol. 24, No. 4, 1989.
- ². K.S. Hong and C.S. Kim, "Linear Stable Systems," *IEEE Transaction on Automatic Control*, Vol. 33, No. 3, 1993.
- ³. B.S. Kim and A.J. Calise, "Nonlinear Flight Control Using Neural Networks," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997.
- ⁴. J. Leitner, A.J. Calise, and J.V.R. Prasad, "Analysis of Adaptive Neural Networks for Helicopter Flight Control," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 5, 1997.
- ⁵. M.B. McFarland and C.N. D'Souza, "Missile Flight Control With Dynamic Inversion and Structured Singular Value Synthesis," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Scottsdale, AZ, August 1994.
- ⁶. R.T. Rysdyk and A.J. Calise, "Nonlinear Adaptive Control of Tiltrotor Aircraft using Neural Networks," *Proceedings of SAE/AIAA World Aviation Congress*, Anaheim, CA, 1997.
- ⁷. A.J. Calise, S. Lee, and M. Sharma, "Development of a Reconfigurable Flight Control Law for the X-36 Tailless Fighter Aircraft," *AIAA-2000-3940 Guidance, Navigation and Control Conference*, Denver, CO, August 2000.
- ⁸. E. Johnson and A.J. Calise, "Feedback Linearization with Neural Network Augmentation Applied to X-33 Attitude Control," *AIAA-2000-4157 Guidance, Navigation and Control Conference*, Denver, CO, August 2000.
- ⁹. B. Min, H. Shin, M. Thak, B.M. Kim, and B.S. Kim, "Auto-Landing Guidance System Design for Smart UAV," *The Korean Society for Aeronautical and Space Sciences*, Vol. 7, No. 1, May 2006.