AIAA Guidance, Navigation, and Control Conference and Exhibit
15 - 18 August 2005, San Francisco, California

AIAA 2005-6002

6-DOF Nonlinear Simulation of Vision-based Formation Flight

Ramachandra Sattigeri[1] and Anthony J. Calise[2]
School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0150

Byoung Soo Kim[3]
School of Mechanical and Aerospace Engineering, Gyeongsang National University, Gyeongnam, SOUTH KOREA

Konstantin Volyanskyy[4]
School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0150

and

Nakwan Kim[5]
School of Aerospace Engineering, Chungnam National University, Daejon, SOUTH KOREA

This paper presents an adaptive guidance and control law algorithm for implementation on a pair of Unmanned Aerial Vehicles (UAVs) in a 6 DOF leader-follower formation flight simulation. The objective of the simulation study is to prepare for a flight test involving a pair of UAVs in formation flight where the follower aircraft will be equipped with an onboard camera to estimate the relative distance and orientation to the leader aircraft. The follower guidance law is an adaptive acceleration based guidance law designed for the purpose of tracking a maneuvering leader aircraft. We also discuss the limitations of a preceding version of the guidance algorithm shown in a previous paper. Finally, we discuss the design of an adaptive controller (autopilot) to track the commands from the guidance algorithm. Simulation results for different leader maneuvers are presented and analyzed.

Nomenclature

X	=	coordinate along inertial X-axis
Y	=	coordinate along inertial Y-axis
Z	=	coordinate along inertial Z-axis
$\dot{X}$	=	velocity component along inertial X-axis
$\dot{Y}$	=	velocity component along inertial Y-axis
$\dot{Z}$	=	velocity component along inertial Z-axis
a	=	body-axes acceleration vector
f	=	body-axes specific force vector
g	=	acceleration due to gravity
p	=	roll-rate
q	=	pitch-rate
r	=	yaw-rate

[1] Graduate Research Assistant, School of Aerospace Engineering, gte334x@prism.gatech.edu, AIAA Member.
[2] Professor, School of Aerospace Engineering, anthony.calise@ae.gatech.edu, Fellow AIAA.
[3] Associate Professor, School of Mechanical and Aerospace Engineering, bskim@gsnu.ac.kr, AIAA Member. Visiting Scholar at Georgia Tech.
[4] Graduate Research Assistant, School of Aerospace Engineering, gtg891s@mail.gatech.edu, AIAA Member.
[5] Assistant Professor, School of Aerospace Engineering, luck1kim@cnu.ac.kr, AIAA Member.

ϕ = roll attitude
θ = pitch attitude
ψ = heading angle
α = angle of attack, subtended angle
β = sideslip angle
δ = actuator signal
τ_a = actuator model time constant
τ_d = actuator model time delay
τ_{wo} = washout filter time constant
K_p = proportional control gain
K_I = integral control gain
K_d = derivative control gain
R = range between leader and follower
λ_X = relative azimuth angle between leader and follower
λ_{XY} = relative azimuth angle between leader and follower
P = inertial position
V = speed
γ = flight path angle
$\hat{u}_{los}$ = unit vector from the follower to the leader aircraft
ν = pseudo-control signal
ν_{ad} = neural network output
ν_h = pseudo-control hedging signal
Δ = inversion error, or modeling error due to approximate dynamic inversion
ω_n = natural frequency
ζ = damping ratio
$sat(\cdot)$ = linear saturation operator
$\langle \bullet \rangle$ = dot-product operator
$\text{Proj}(\cdot,\cdot)$ = projection operator

Subscripts

e = elevator
a = aileron
r = rudder
T = throttle
cmd = command
com = command
x = projection along body X-axis
y = projection along body Y-axis
z = projection along body Z-axis
L = Leader
F = Follower
wo = washout
i = waypoint index
min = minimum
c = command filter, reference model

Abbreviations

DOF = Degrees of Freedom
UAV = Unmanned Aerial Vehicle
MAV = Micro Unmanned Aerial Vehicle
GPS = Global Positioning System
LOS = Line of Sight
EKF = Extended Kalman Filter
NN = Neural Network
MRAC = Model Reference Adaptive Control
PCH = Pseudo-Control Hedging
FOV = Field of View
MIMO = Multi-Input Multi-Output
c.g. = center of gravity

I. Introduction

As demonstrated in recent conflicts, UAVs are becoming an important component of our military force structure. UAVs, operating in close proximity to enemy forces, provide real-time information difficult to obtain from other sources, without risk to human pilots. Among the strategies employed by these UAVs will be flocks of cooperative MAVs operating in close proximity to terrain or structures that will gather information and, under human supervision, seek out, identify, and engage targets. They will be expected to maintain a formation while at the same time executing searches in a congested environment. Stealth like operations will also be important, implying the need to maintain autonomy and to minimize communication. Maintaining a formation is also important from this perspective so that passive (vision based) sensing can be used to ascertain the locations and behaviors of cooperating MAVs/UAVs.

The motivation for this paper comes from the need to flight test UAVs in formation flight using only onboard cameras for relative LOS information. In our previous paper [1], we presented an approach wherein formation flight between multiple UAVs could be maintained by implementing an adaptive guidance scheme. In this approach, it was assumed that an onboard camera on a follower aircraft could provide estimates of range and relative orientation to a leader aircraft. An estimate of the range-rate that is required by the guidance law is generated as the output of an adaptive NN that is trained online using information available from the navigation and vision sensors. However, this approach was found to be deficient when applied in a 6 DOF aircraft model setting. The reasons are the assumptions involved in the adaptive guidance law design. The approach commanded a velocity vector (speed and heading) along the LOS to the leader aircraft. The velocity vector dynamics was assumed to be of 1^{st} order, subject to maximum and minimum acceleration limits. Secondly, the velocity vector dynamics were assumed to be significantly faster than the desired range dynamics. These assumptions break down in a 6 DOF setting. The consequences of the assumptions showed up as significantly degraded range tracking performance in the presence of leader aircraft maneuvers. In order to take into account the velocity dynamics, we have shifted to an acceleration based guidance law design. The basic approach to guidance law design however remains the same. The range and LOS angles are output feedback variables to be controlled. An inverting controller inverts the approximate range and LOS angle dynamics. The inverting controller is augmented by the output of an adaptive NN that compensates for the modeling error due to inversion. The outputs of this inverting controller are acceleration commands along and perpendicular to the LOS direction.

Flight test results on autonomous formation flight have been reported in recent literature. Ref. [2] provided flight test results on 2 F/A-18 aircraft in close-coupled formation flight. Ref. [3] discusses flight test techniques to evaluate performance benefits in close-coupled formation flight. Ref. [4] contained flight test results of a pair of scaled YF-22 research aircraft in formation flight. In this case, the objective was to provide a formation flight demonstration of a low-cost "off the shelf" hardware system. The leader aircraft's inertial position and velocity vector information were continuously communicated to the follower aircraft. To the best of our knowledge, there haven't been flight test results reported on autonomous formation flight with a vision-based onboard sensor for estimating relative position and velocity from a leader aircraft. Ref. [5] describes the design of formation control laws, a simulation setup and provides some results for close formation flight based on an onboard vision sensing system.

The contributions of this paper can be summarized as follows. The paper includes an adaptive guidance and control law design algorithm for a pair of UAVs in a 6 DOF leader-follower formation flight configuration. No communication between the UAVs is assumed. The follower UAV is equipped with just one fixed camera for passive sensing of the LOS information. The adaptive nature of the follower's guidance law prevents significant degradation of the range tracking performance in the presence of leader aircraft maneuvers. We also present an adaptive controller (autopilot) design to track the commands coming from the guidance. The adaptive autopilot is common to both the leader and follower aircraft. The adaptive nature of the autopilot design allows us to use the same autopilot under different operating conditions and subject to parametric uncertainty in the aerodynamic data. This greatly reduces the dependence of the design on the aircraft aerodynamic data.

Section II describes the overall closed-loop UAV system and briefly discusses the Image Processing and Estimation blocks. Section III discusses the guidance law design for the leader and follower aircraft. Section IV presents the adaptive autopilot design. Section V summarizes the 6 DOF rigid-body aircraft model and actuator model used in the simulation. Section VI presents and analyzes the simulation results, and conclusions are presented in Section VII.

II. Image Processing and Estimation

The complete closed-loop system is summarized in the block diagram in Figure 1. The Image Processing Computer Vision block and Estimation block are summarized in this section.

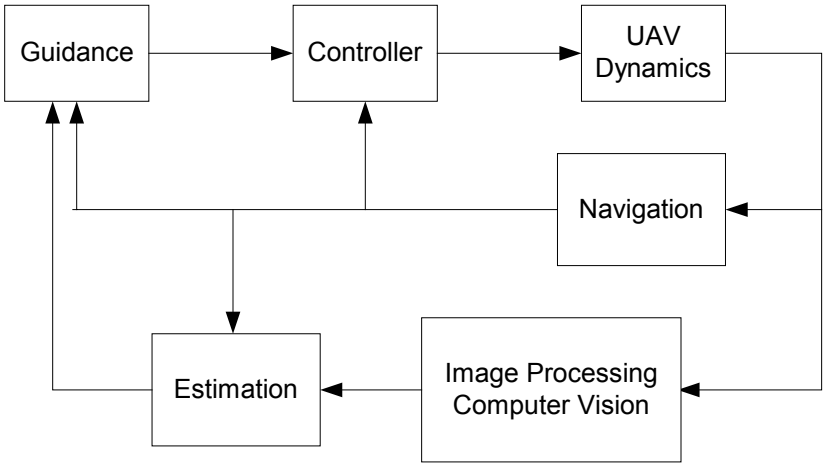

Figure 1. Closed-loop UAV system

The Image Processing and Computer Vision block takes as input the image frames from an onboard camera and processes them in real-time for visual tracking of a target (leader) aircraft. This block utilizes the method of *geometric active contours* [6,7] to track various features of interest in the image frames over a period of time. Active contours have the ability to conform to various object shapes and motions, making them ideal for segmentation, edge detection, shape modeling and visual tracking. Level set methods allow for fast, robust implementations of the active contours algorithms [6].

For the purposes of formation flight, the guidance block needs estimates of the range, range-rate, LOS angles and LOS rates between the leader and follower to compute acceleration and velocity commands to the controller. While the LOS angles are available from the images, the range and range-rate are not. The estimates of range, range-rate and LOS rates have to be generated by an estimator, usually an EKF. Various implementations of the EKF are possible, employing different states [6,8], and could be augmented with an adaptive element for robust estimation in the presence of unmodeled dynamics and disturbances [9]. The Image Processing block provides the position and size of the leader aircraft in the camera images. From the position, a unit-vector $\hat{u}_{los}$ to the leader aircraft is computed, and the maximum angle subtended by the leader aircraft on the image plane α is computed from the size of the leader aircraft (Figure 2). The EKF model then includes the states $\left[\hat{u}_{los}, \dot{\hat{u}}_{los}, \frac{1}{R}, \frac{\dot{R}}{R}, b\right]^T$ driven by the

measurements $[\hat{u}_{los}, \alpha]^T$ where $R, \dot{R}$ are the range and range-rate respectively, and b is the wing-span of the leader aircraft [8].

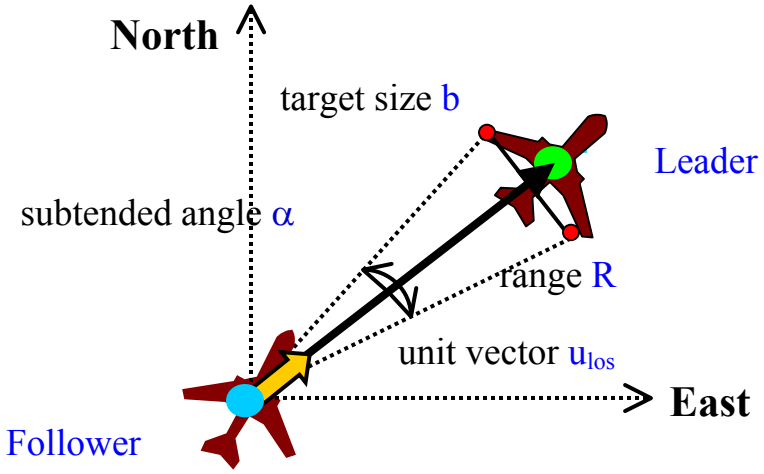

Figure 2. Leader-Follower Configuration

At this stage we have not integrated our adaptive guidance and control algorithms with the image processing and estimation. Presently, we are using the true values of the range (R), LOS elevation (λ_{XY}), and LOS azimuth (λ_X) of the leader with respect to the follower. An approximate differentiator operates on λ_{XY} and λ_X to provide estimates of $\dot{\lambda}_{XY}$ and $\dot{\lambda}_X$ respectively, and an estimate of range-rate is obtained using an error observer[10]. The relationships between range and the LOS angles with the inertial coordinates of the leader and follower are:

$$R = \sqrt{(X_L - X_F)^2 + (Y_L - Y_F)^2 + (Z_L - Z_F)^2} \tag{1}$$

$$\lambda_X = a\tan 2(Y_L - Y_F, X_L - X_F) \tag{2}$$

$$\lambda_{XY} = \tan^{-1}\left(- \frac{(Z_L - Z_F)}{\sqrt{(X_L - X_F)^2 + (Y_L - Y_F)^2}} \right) \tag{3}$$

III. Guidance Law Design for Formation Flight

The objective of the formation flight experiment will be for the follower aircraft to maintain a prescribed relative position from the leader aircraft in the presence of leader maneuvers and other unmodeled disturbances. The only information about the leader aircraft is the LOS data available directly from the image processing and estimation.

The control law design for the formation flight problem, like in other conventional flight trajectory control problems, exhibits a two time-scale feature because the trajectory dynamics (relative position and velocity) are slower than the attitude dynamics (orientation and angular velocity). Hence, the controller consists of a guidance algorithm whose inputs are the commanded positions in inertial space and outputs are the body-axes specific forces (or accelerations) and attitude angles that form the inputs to the inner-loop controller. The inner-loop controller is an autopilot that produces actuator command signals. In the following sections, we will discuss the guidance algorithms for the leader and follower aircraft.

A. Leader Aircraft Guidance

The leader aircraft is assigned to track waypoints in inertial space. The commanded waypoint coordinates are $\vec{P}_{com,i} = [X_{com}, Y_{com}, Z_{com}]_i^T$, $i = 1,2,\ldots,N$, N is the number of waypoints, and i is the waypoint index. The guidance logic constructs a command inertial velocity vector $\vec{V}_{com}$ of constant magnitude oriented along the LOS from the current position coordinates of the vehicle $\vec{P}$ to the assigned waypoint, i.e.,

$$\vec{V}_{com} = V_{com}\left(\frac{\vec{P}_{com,i} - \vec{P}}{\left|\vec{P}_{com,i} - \vec{P}\right|}\right) \tag{4}$$

To transition to the $(i+1)^{th}$ waypoint from the i^{th} waypoint, the following rule has to be satisfied

$$d = \frac{\left|\vec{A}\mathrm{x}\vec{B}\right|}{\left|\vec{B}\right|} \le d_{\min} \tag{5}$$

where d is the perpendicular distance from the current aircraft position to the line joining the i^{th} waypoint to the $(i+1)^{th}$ waypoint as shown in figure 3, and $d_{\min}$ is a constant.

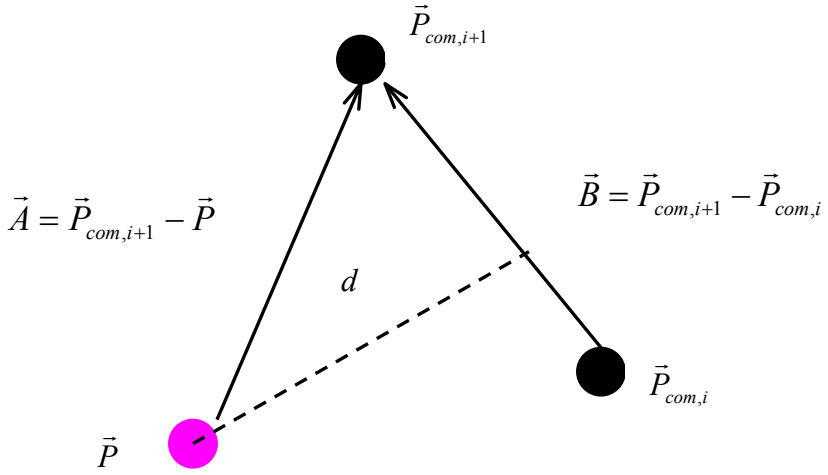

Figure 3. Waypoint Transitioning Logic

To implement $\vec{V}_{com}$, we first compute inertial acceleration commands. Let $\vec{V}$ be the velocity vector, and V, ψ, γ be the speed, heading angle and flight path angle respectively. Thus,

$$\vec{V} = (V\cos\psi\cos\gamma)\hat{I} + (V\sin\psi\cos\gamma)\hat{J} - (V\sin\gamma)\hat{K} \tag{6}$$

where $\hat{I}, \hat{J}, \hat{K}$ are the unit vectors of the Earth-surface fixed reference frame. Then $\vec{V}_{com}$ can be resolved into speed, heading and flight path angle commands as,

$$\vec{V}_{com} = (V_{com}\cos\psi_{com}\cos\gamma_{com})\hat{I} + (V_{com}\sin\psi_{com}\cos\gamma_{com})\hat{J} - (V_{com}\sin\gamma_{com})\hat{K} \tag{7}$$

Differentiating eq. (6) w.r.t. time and treating the Earth-fixed frame as an inertial frame, we have,

$$\dot{\vec{V}} = \left(\dot{V}\cos\psi\cos\gamma - V\dot{\psi}\sin\psi\cos\gamma - V\dot{\gamma}\cos\psi\sin\gamma\right)\hat{I} + (\dot{V}\sin\psi\cos\gamma + V\dot{\psi}\cos\psi\cos\gamma - V\dot{\gamma}\sin\psi\sin\gamma)\hat{J} - \left(\dot{V}\sin\gamma + V\dot{\gamma}\cos\gamma\right)\hat{K} \tag{8}$$

The inertial acceleration command is obtained by replacing $\dot{\vec{V}}$ with $\vec{A}_{com}$ on the left hand side of eq. (8) and the true velocity dynamics $\dot{V}, \dot{\psi}, \dot{\gamma}$ by the commanded velocity dynamics $\dot{V}_{com}, \dot{\psi}_{com}, \dot{\gamma}_{com}$ on the right hand side.

$$\vec{A}_{com} = \left(\dot{V}_{com}\cos\psi_{com}\cos\gamma_{com} - V_{com}\dot{\psi}_{com}\sin\psi_{com}\cos\gamma_{com} - V_{com}\dot{\gamma}_{com}\cos\psi_{com}\sin\gamma_{com}\right)\hat{I} + (\dot{V}_{com}\sin\psi_{com}\cos\gamma_{com} + V_{com}\dot{\psi}_{com}\cos\psi_{com}\cos\gamma_{com} - V_{com}\dot{\gamma}_{com}\sin\psi_{com}\sin\gamma_{com})\hat{J} - \left(\dot{V}_{com}\sin\gamma_{com} + V_{com}\dot{\gamma}_{com}\cos\gamma_{com}\right)\hat{K} \tag{9}$$

The commanded velocity dynamics are computed as

$$\dot{V}_{com} = \left(K_{p_V} + \frac{K_{I_V}}{s}\right)\left(\text{sat}\left[V_{com}\right] - V\right) \tag{10}$$

$$\dot{\psi}_{com} = K_{p_\psi}\left(\psi_{com} - \psi\right) \tag{11}$$

$$\dot{\gamma}_{com} = K_{p_\gamma}\left(\gamma_{com} - \gamma\right) \tag{12}$$

Eq. (10) shows that $\dot{V}_{com}$ is the output of a PI controller acting on the commanded speed error. Integral control was included because a steady-state error was noticed in the tracking of the speed command. Also, note that the speed command is limited by a linear saturation element in eq. (10). This is to prevent large speed commands that can saturate the controls. Eq. (11) and (12) show that $\dot{\psi}_{com}$ and $\dot{\gamma}_{com}$ are the outputs of a proportional controller acting on the commanded heading and flight path angle error.

B. Follower Aircraft Guidance

In a previous paper [1], we presented an adaptive formation guidance approach in which a follower aircraft could closely maintain a commanded range from a leader aircraft, even in the presence of leader maneuvers. This approach considered range as a relative degree 1 variable and designed an output feedback based inverting controller, augmented with a NN for compensating the leader aircraft's velocity along the LOS. The output of the adaptive guidance approach was a velocity vector command oriented along the LOS to the leader aircraft. This approach was found to be deficient when applied in a 6 DOF setting. The approach assumed first-order velocity dynamics that were sufficiently faster than the desired range dynamics, so that the velocity vector command $\left(V_{com}, \psi_{com}, \gamma_{com}\right)$ of the formation guidance was achieved with only a small, first-order lag subject to acceleration limits. In the 6 DOF setting, the velocity dynamics are definitely not first order and are not much faster than the desired range dynamics. Secondly, commanding the velocity vector along the LOS direction resulted in a tail-chase situation for the follower aircraft, causing it to maneuver a lot just to keep the commanded range from the leader. The consequence of the assumptions of the formation guidance approach resulted in significantly degraded range tracking performance in the 6 DOF setting. The overshoots in the range variable were large during leader maneuvers and range commands of the order of 1-2 wing-span lengths resulted in destabilization of the follower closed-loop system.

The guidance law design in this paper takes into account the velocity dynamics by considering the range and LOS angles as relative degree 2 variables to be controlled. This means that the guidance law generates acceleration commands instead of velocity commands. If we try to control the range by controlling motion only along the LOS axis, i.e., by commanding acceleration only along the LOS axis, we lose controllability of the LOS dynamics. This can be checked by linearizing eq. (13)-(15), zeroing out the acceleration terms in eq. (14) and (15), and doing a controllability analysis of the resulting linear system. Hence, one possibility is to control both range and the LOS angles. However, since it is not desirable to restrict the follower aircraft to a particular orientation with respect to the

leader aircraft, particularly in the presence of leader maneuvers, we control the LOS rates instead of the LOS angles. In the present problem formulation, we command zero LOS rates.

Differentiating eq. (1)-(3) twice, we have,

$$\ddot{R} = R\left(\dot{\lambda}_{XY}^2 + \dot{\lambda}_X^2 \cos^2 \lambda_{XY}\right) + \left[a_{L_R} - a_{F_R}\right] \tag{13}$$

$$\ddot{\lambda}_X = \frac{1}{\cos \lambda_{XY}} \left\{ -2\dot{\lambda}_X \left[\left(\frac{\dot{R}}{R} \right) \cos \lambda_{XY} - \dot{\lambda}_{XY} \sin \lambda_{XY} \right] + \left(\frac{1}{R} \right) \left[a_{L_{\lambda x}} - a_{F_{\lambda x}} \right] \right\} \tag{14}$$

$$\ddot{\lambda}_{XY} = -2\left(\frac{\dot{R}}{R} \right) \dot{\lambda}_{XY} - \dot{\lambda}_X^2 \sin \lambda_{XY} \cos \lambda_{XY} - \left(\frac{1}{R} \right) \left[a_{L_{\lambda xy}} - a_{F_{\lambda xy}} \right] \tag{15}$$

where the terms a_{L_R}, $a_{L_{\lambda_X}}$, $a_{L_{\lambda_{XY}}}$ represent leader aircraft accelerations along the LOS and perpendicular to the LOS respectively, and likewise for the follower aircraft. The dynamics in eq. (13)-(15) are inverted to construct the acceleration commands for the follower aircraft. The first step in the process of dynamic inversion is to re-write the right hand sides of eq. (13)-(15) as

$$\ddot{R} = \nu_R + \left\{ R\left(\dot{\lambda}_{XY}^2 + \dot{\lambda}_X^2 \cos^2 \lambda_{XY}\right) + a_{L_R} \right\} = \nu_R + \Delta_R \tag{16}$$

$$\ddot{\lambda}_X = \nu_{\lambda_X} + \frac{1}{\cos \lambda_{XY}} \left\{ -2\dot{\lambda}_X \left[\left(\frac{\dot{R}}{R} \right) \cos \lambda_{XY} - \dot{\lambda}_{XY} \sin \lambda_{XY} \right] + \left(\frac{a_{L_{\lambda x}}}{R} \right) \right\} = \nu_{\lambda_X} + \Delta_{\lambda_X} \tag{17}$$

$$\ddot{\lambda}_{XY} = \nu_{\lambda_{XY}} + \left[-2\left(\frac{\dot{R}}{R} \right) \dot{\lambda}_{XY} - \dot{\lambda}_X^2 \sin \lambda_{XY} \cos \lambda_{XY} - \left(\frac{a_{L_{\lambda xy}}}{R} \right) \right] = \nu_{\lambda_{XY}} + \Delta_{\lambda_{XY}} \tag{18}$$

where ν_R, ν_{λ_X} and $\nu_{\lambda_{XY}}$ are the so-called 'pseudo-controls'. These represent an approximation of the $\ddot{R}, \ddot{\lambda}_X$ and $\ddot{\lambda}_{XY}$ dynamics. The terms $\Delta_R, \Delta_{\lambda_X}$ and $\Delta_{\lambda_{XY}}$ represent the inversion error due to the terms that are ignored in the approximation.

$$\nu_R = -a_{F_R} \tag{19}$$

$$\nu_{\lambda_X} = \frac{-a_{F_{\lambda x}}}{R \cos \lambda_{XY}} \tag{20}$$

$$\nu_{\lambda_{XY}} = \frac{a_{F_{\lambda xy}}}{R} \tag{21}$$

Equations (19)-(21) are inverted to give the acceleration commands for the follower aircraft,

$$a_{F_{R,COM}} = -\nu_R \tag{22}$$

$$a_{F_{\lambda x,COM}} = -R \cos \lambda_{XY} \nu_{\lambda_X} \tag{23}$$

$$a_{F_{\lambda xy,COM}} = R \nu_{\lambda_{XY}} \tag{24}$$

The acceleration command for the follower aircraft can be written in vector form as:

$$\vec{A}_{F_{com}} = a_{F_{R,COM}}\hat{e}_R + a_{F_{\lambda x,COM}}\hat{e}_{\lambda_X} + a_{F_{\lambda xy,COM}}\hat{e}_{\lambda_{XY}} = A_{F_{X,COM}}\hat{I} + A_{F_{Y,COM}}\hat{J} + A_{F_{Z,COM}}\hat{K} \tag{25}$$

where $\hat{I},\hat{J},\hat{K}$ are inertial coordinate unit vectors and $\hat{e}_R,\hat{e}_{\lambda_X},\hat{e}_{\lambda_{XY}}$ are spherical coordinate unit vectors along and perpendicular to the LOS direction. The coordinate transformation between the inertial coordinate system and the spherical coordinate system is

$$\begin{bmatrix} A_{F_{X,COM}} \\ A_{F_{Y,COM}} \\ A_{F_{Z,COM}} \end{bmatrix} = \begin{bmatrix} \cos\lambda_X & -\sin\lambda_X & 0 \\ \sin\lambda_X & \cos\lambda_X & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\lambda_{XY} & 0 & \sin\lambda_{XY} \\ 0 & 1 & 0 \\ -\sin\lambda_{XY} & 0 & \cos\lambda_{XY} \end{bmatrix} \begin{bmatrix} a_{F_{R,COM}} \\ a_{F_{\lambda x,COM}} \\ a_{F_{\lambda xy,COM}} \end{bmatrix} \tag{26}$$

It is seen that the process of dynamic inversion ignores the nonlinearities and leader acceleration terms in eq. (13)-(15). To compensate for these missing terms, we augment the pseudo-controls with the outputs of an adaptive NN that is trained online with inputs from the image processing and navigation blocks. We thus have a MIMO Adaptive Output Feedback Control problem. The theory behind our approach to this problem is described in detail in [10]. Refer to figure 4 for a block diagram of the follower's guidance logic.

The pseudo-control vector $\vec{\nu} = \left[\nu_{\lambda_X}, \nu_{\lambda_{XY}}, \nu_{\lambda_R}\right]^T$ is constructed as,

$$\vec{\nu} = \vec{\nu}_{crm} + \vec{\nu}_{dc} - \vec{\nu}_{ad} \tag{27}$$

where $\vec{\nu}_{crm}$, $\vec{\nu}_{dc}$ and $\vec{\nu}_{ad}$ represent vectors of outputs from reference models, linear controllers and the NN respectively. The range command R_{com} is filtered through a 2nd order reference model, while the LOS rate commands are filtered through 1st order reference models. Thus the outputs of the reference models are defined by:

$$\begin{aligned} \nu_{crm_R} &= \omega_{n_R}^2\left(R_{com} - R_c\right) - 2\varsigma_R\omega_{n_R}\dot{R}_c \\ \nu_{crm_{\lambda x}} &= \frac{1}{\tau_{\lambda_X}}\left(\dot{\lambda}_{X_{com}} - \dot{\lambda}_{X_c}\right) \\ \nu_{crm_{\lambda xy}} &= \frac{1}{\tau_{\lambda_{XY}}}\left(\dot{\lambda}_{XY_{com}} - \dot{\lambda}_{XY_c}\right) \end{aligned} \tag{28}$$

where $R_c, \dot{R}_c, \dot{\lambda}_{X_c}, \dot{\lambda}_{XY_c}$ are states of the respective reference models, and $\omega_{n_R}^2, \varsigma_R, \tau_{\lambda_X}, \tau_{\lambda_{XY}}$ are tunable parameters set by choosing a desired form of response based on rise time, settling time, and maximum overshoot.

A first order dynamic compensator is used for controlling range, and proportional controllers are used to control the LOS rates. For controlling the range variable, the linear compensator chosen is a 1st order dynamic compensator, and for controlling the LOS rates, the linear compensators are just proportional error controllers.

$$\begin{aligned} \dot{\eta} &= a_c\eta + b_c\left(R_c - R\right) \\ \nu_{dc_R} &= c_c\eta + d_c\left(R_c - R\right) \\ \nu_{dc_{\lambda x}} &= K_{\lambda_X}\left(\dot{\lambda}_{X_c} - \hat{\dot{\lambda}}_X\right) \\ \nu_{dc_{\lambda xy}} &= K_{\lambda_{XY}}\left(\dot{\lambda}_{XY_c} - \hat{\dot{\lambda}}_{XY}\right) \end{aligned} \tag{29}$$

where , a low-pass differentiator (derivative filter in figure 4) is used to estimate the LOS rates, η is the state of

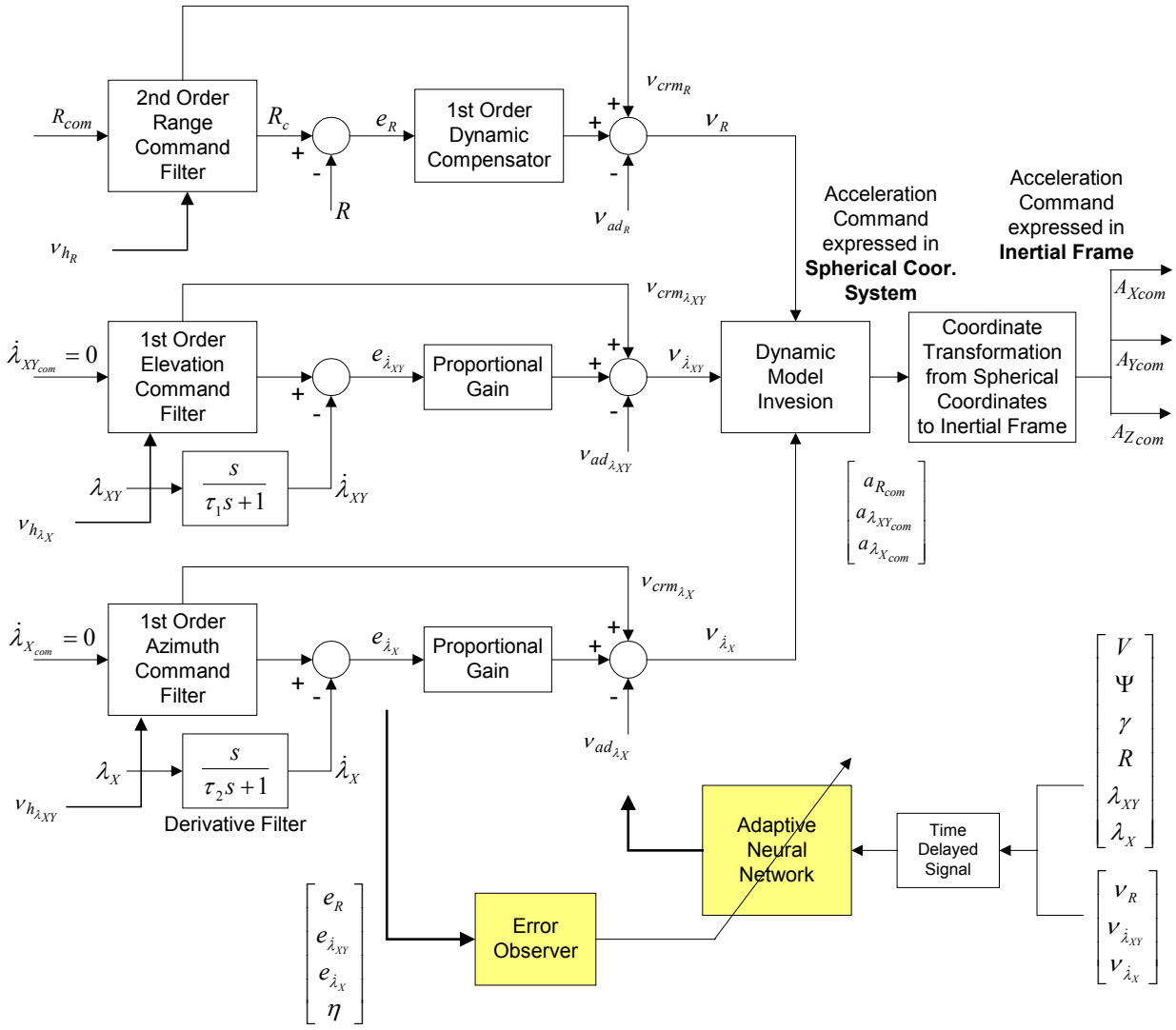

Figure 4. Follower Guidance Logic Block Diagram

the 1st order dynamic compensator, (a_c, b_c, c_c, d_c) are the parameters of this compensator, $K_{\lambda_X}, K_{\lambda_{XY}}$ are the proportional controller gains. The linear compensator parameters are chosen such that the closed-loop tracking error and η dynamics are asymptotically stable in the absence of any modeling uncertainty [10].

The tracking error vector is defined to be

$$e = \begin{bmatrix} e_R \\ \dot{e}_R \\ e_{\lambda_X} \\ e_{\lambda_{XY}} \end{bmatrix} = \begin{bmatrix} R_c - R \\ \dot{R}_c - \dot{R} \\ \dot{\lambda}_{X_c} - \hat{\dot{\lambda}}_X \\ \dot{\lambda}_{XY_c} - \hat{\dot{\lambda}}_{XY} \end{bmatrix} \tag{30}$$

Defining $E = \left[e^T, \eta\right]^T$ the tracking error dynamics is given by,

$$\dot{E} = \begin{bmatrix} \dot{e}_R \\ \ddot{e}_R \\ \dot{e}_{\lambda_X} \\ \dot{e}_{\lambda_{XY}} \\ \dot{\eta} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -d_c & 0 & 0 & 0 & -c_c \\ 0 & 0 & -K_{\lambda_X} & 0 & 0 \\ 0 & 0 & 0 & -K_{\lambda_{XY}} & 0 \\ b_c & 0 & 0 & 0 & a_c \end{bmatrix}}_{\overline{A}} \begin{bmatrix} e_R \\ \dot{e}_R \\ e_{\lambda_X} \\ e_{\lambda_{XY}} \\ \eta \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{\overline{B}} \begin{bmatrix} \nu_{ad_R} - \Delta_R \\ \nu_{ad_{\lambda_X}} - \Delta_{\lambda_X} \\ \nu_{ad_{\lambda_{XY}}} - \Delta_{\lambda_{XY}} \end{bmatrix}$$

$$z = \begin{bmatrix} e_R \\ e_{\lambda_X} \\ e_{\lambda_{XY}} \\ \eta \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\overline{C}} \begin{bmatrix} e_R \\ \dot{e}_R \\ e_{\lambda_X} \\ e_{\lambda_{XY}} \\ \eta \end{bmatrix} = \overline{C}E \tag{31}$$

For full state-feedback applications, the NN weight update laws are in terms of the error vector e [16,17]. However, $\dot{R}$, $\dot{\lambda}_X$ and $\dot{\lambda}_{XY}$ are not directly available. As noted earlier, low-pass differentiators are used to estimate the LOS rates. A similar approach could be used for the range-rate. However we employ an error observer to estimate the entire error vector E by defining $\hat{E} = \left[\hat{e}_R,\ \hat{\dot{e}}_R,\ \hat{e}_{\lambda_X},\ \hat{e}_{\lambda_{XY}},\ \hat{\eta}\right]^T$. The error observer dynamics is given by,

$$\dot{\hat{E}} = \left(\overline{A} - K_{obs}\overline{C}\right)\hat{E} + K_{obs}z \tag{32}$$

where $\widetilde{A} = \overline{A} - K_{obs}\overline{C}$ is an asymptotically stable matrix and K_{obs} is the error observer gain matrix [10]. The error observer gain matrix is designed by pole placement techniques by specifying the eigenvalues of $\widetilde{A}$ to be 4 times faster than that of $\overline{A}$.

The NN is a SHL NN having the input vector,

$$\overline{\mu}(t) = \left[1\ \ \overline{\nu}_{\mathrm{d}}^{\mathrm{T}}(t)\ \ \overline{R}_d^T(t)\ \ \overline{\lambda}_{XY\,d}^T(t)\ \ \overline{\lambda}_{X\,d}^T(t)\ \ \overline{V}_d^T(t)\ \ \overline{\psi}_d^T(t)\ \ \overline{\gamma}_d^T(t)\right]^T \tag{33}$$

where the bar and subscript d signify vectors of delayed values of the signals [10]. The input vector represents the time histories of input and output data required by the NN to reconstruct the states of the unknown dynamics in eq. (16)-(18). The input-output map of the SHL NN is given by

$$\vec{\nu}_{ad} = M^T\sigma\left(N^T\overline{\mu}\right) \tag{34}$$

where σ is the so-called squashing function. The NN is trained online with the adaptive law [11]

$$\begin{aligned} \dot{M} &= -\Gamma_M\,\mathrm{Proj}\left(M, \sigma\hat{E}^T P\overline{B}\right) \\ \dot{N} &= -\Gamma_N\,\mathrm{Proj}\left(N, \overline{\mu}\hat{E}^T P\overline{B}M^T\sigma'\right) \end{aligned} \tag{35}$$

where $\mathrm{Proj}(\cdot,\cdot)$ is a projection operator [15], the term $\hat{E}^T P\overline{B}$ is the training signal for the NN, $\Gamma_M > 0$ and $\Gamma_N > 0$ are the adaptation gains. The adaptive law in eq. (35) guarantees (subject to upper and lower bounds on the adaptation gains) that all error signals and the NN weights are uniformly ultimately bounded [11].

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 plant [12]. 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 [12]. Note that since we are commanding acceleration (eq. (26)), the acceleration dynamics are treated as actuator dynamics and hedged.

The estimate of the achieved pseudo-control is given by

$$\hat{\nu}_{\lambda_X} = \frac{-a_{F_{\lambda x}}}{R\cos\lambda_{XY}} \tag{36}$$

$$\hat{\nu}_{\lambda_{XY}} = \frac{a_{F_{\lambda xy}}}{R} \tag{37}$$

$$\hat{\nu}_R = -a_{F_R} \tag{38}$$

where the acceleration signals $a_{F_{\lambda x}}$, $a_{F_{\lambda xy}}$ and a_{F_R} are computed from the achieved inertial acceleration and by inverse transforming eq. (26). The hedge signals are given by

$$\begin{bmatrix} \nu_{h_R} \\ \nu_{h_{\lambda_X}} \\ \nu_{h_{\lambda_{XY}}} \end{bmatrix} = \begin{bmatrix} \nu_R - \hat{\nu}_R \\ \nu_{\lambda_X} - \hat{\nu}_{\lambda_X} \\ \nu_{\lambda_{XY}} - \hat{\nu}_{\lambda_{XY}} \end{bmatrix} \tag{39}$$

The reference models are then updated with the hedge signals as shown below,

$$\begin{aligned} \ddot{R}_c &= \nu_{crm_R} - \nu_{h_R} \\ \ddot{\lambda}_{X\,c} &= \nu_{crm_{\lambda x}} - \nu_{h_{\lambda_X}} \\ \ddot{\lambda}_{XY\,c} &= \nu_{crm_{\lambda x}} - \nu_{h_{\lambda_X}} \end{aligned} \tag{40}$$

C. Converting Acceleration Commands into Autopilot Commands

In this section, we describe the process of converting the inertial acceleration commands $\vec{A}_{com}$ from the guidance algorithms of the follower (eq. 26) and leader (eq. 9) aircraft into commands for the autopilot. The gravity vector is subtracted from $\vec{A}_{com}$ to form the commanded specific force vector, which is basically the desired aerodynamic plus thrust force vector divided by mass.

$$\vec{F}_{com} = \vec{A}_{com} - [0,\ 0,\ g]^T \tag{41}$$

The commands to the inner-loop controller are the body frame x-axis and z-axis specific force commands ${}_B f_{x_{com}}$, ${}_B f_{z_{com}}$, and the bank angle command ϕ_{com}. The commanded specific force vector $\vec{F}_{com}$ is rotated into the body frame axes to form the above commands. This is accomplished by rotating through the Euler angles ψ (heading), θ (pitch), and ϕ_{com} (roll). The intermediate frame formed after the ψ and θ rotations is called the 2-frame for convenience. The bank angle command ϕ_{com} rotates the $\vec{F}_{com}$ components in the 2-frame ${}_2\vec{f}_{com}$, to the body frame components ${}_B\vec{f}_{com}$.

$${}_2\vec{f}_{com} = [L_{2V}(\psi,\theta)]\vec{F}_{com} \tag{42}$$

$$\left[L_{2V}(\psi,\theta)\right]=\begin{bmatrix}\cos\theta & 0 & -\sin\theta\\ 0 & 1 & 0\\ \sin\theta & 0 & \cos\theta\end{bmatrix}\begin{bmatrix}\cos\psi & \sin\psi & 0\\ -\sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix} \tag{43}$$

$$_B\vec{f}_{com}=\left[L_{B2}(\phi_{com})\right]_2\vec{f}_{com} \tag{44}$$

The components of the commanded vectors in eq. (42) and (44) are given by

$$_2\vec{f}_{com}=\begin{bmatrix}_2f_{x_{com}}\\ _2f_{y_{com}}\\ _2f_{z_{com}}\end{bmatrix},\ _B\vec{f}_{com}=\begin{bmatrix}_Bf_{x_{com}}\\ _Bf_{y_{com}}\\ _Bf_{z_{com}}\end{bmatrix}$$

The situation after rotation into the 2-frame is shown in figure 5. Figure 5 depicts the view from the front of the aircraft, with the X-axis coming out of the page.

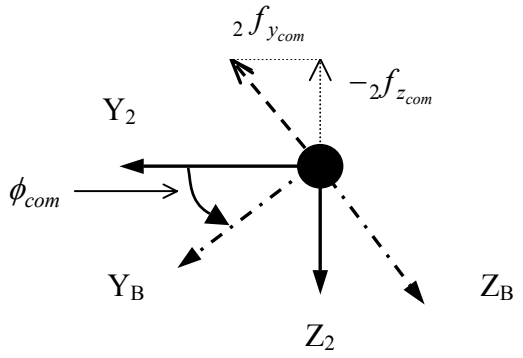

Figure 5. Front view of the 2-frame and body frame

From figure 5, the bank angle command is constructed as

$$\phi_{com}=a\tan 2\left(_2f_{y_{com}},-_2f_{z_{com}}\right) \tag{45}$$

From figure 5, the body frame z-axis specific force command is given as

$$_Bf_{z_{com}}=-\sqrt{_2f_{y_{com}}^2+_2f_{z_{com}}^2} \tag{46}$$

The body frame x-axis specific force command is

$$_Bf_{x_{com}}=_2f_{x_{com}} \tag{47}$$

IV. Adaptive Autopilot

We design adaptive controllers for tracking the normal acceleration command $_Bf_{z_{com}}$, lateral acceleration $_Bf_{y_{com}}=0$ and bank angle command ϕ_{com}. The throttle controller is a PI controller with an anti-windup feature[13] for tracking speed command V_{com} or a command formed by a combination of the longitudinal acceleration

command ${}_B f_{x_{com}}$ and V_{com}. The reason why we don't use a pure acceleration command as input to the throttle controller is explained below in the section on throttle autopilot.

A. Adaptive Normal Acceleration (f_z) Control

The control design is based on the JDAM approach to adaptive autopilot design in [14]. This approach uses an inverting controller augmented by the output of an adaptive NN for pitch rate q control in the inner loop, and a PI controller for f_z control in the outer loop (figure 6).

The inverting controller in the inner loop is based on the short period approximation of the longitudinal aircraft dynamics,

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \hat{Z}_\alpha / u_0 & 1 \\ \hat{M}_\alpha & \hat{M}_{\mathrm{q}} \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} \hat{Z}_\alpha / u_0 \\ \hat{M}_\delta \end{bmatrix} \delta_e \tag{48}$$

where u_0 is the equilibrium flight speed and "^" represents an estimated quantity in the model. The true q dynamics are given by

$$\dot{q} = f_q(x, \delta_e) \tag{49}$$

where x is the state of the plant. Using the linear approximation to the q dynamics in eq. (48), the true q dynamics can be represented as

$$\dot{q} = \hat{M}_\alpha \alpha + \hat{M}_q q + \hat{M}_\delta \delta e + \Delta_q(x, \delta_e) = \nu_q + \Delta_q(x, \delta_e) \tag{50}$$

where $\nu_q = \hat{M}_\alpha \alpha + \hat{M}_q q + \hat{M}_\delta \delta e$, is the pseudo-control signal, and $\Delta_q(x, \delta_e) = f_q(x, \delta_e) - \nu_q$ is the modeling error. Then ν_q can be solved for the actuator command input $\delta_{e_{com}}$ as

$$\delta_{e_{com}} = \frac{1}{\hat{M}_\delta}\left(\nu_q - \hat{M}_\alpha \alpha - \hat{M}_q q\right) \tag{51}$$

The pseudo-control ν_q is constructed as

$$\nu_q = \nu_{crm_q} + K_{p_q}(q_c - q) - \nu_{ad\,q} \tag{52}$$

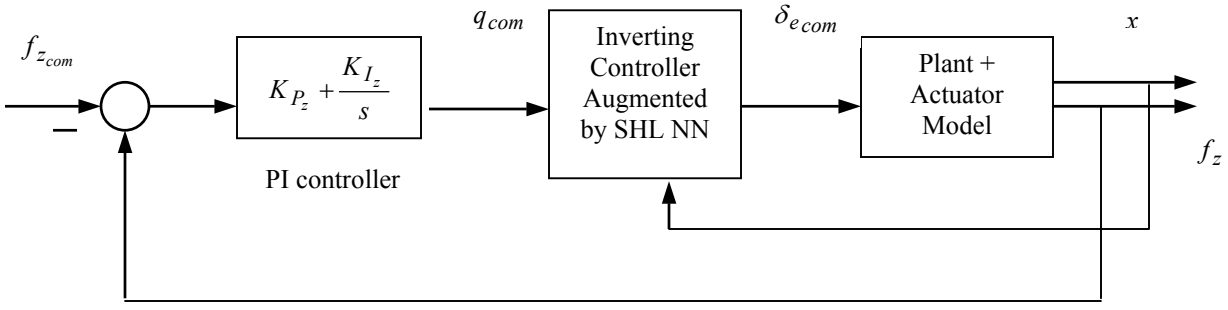

Figure 6. Adaptive Normal Acceleration Controller

$$\nu_{crm_q} = \frac{1}{\tau_q}\left(q_{com} - q_c\right) \tag{53}$$

where $\nu_q = \hat{M}_\alpha \alpha + \hat{M}_q q + \hat{M}_\delta \delta e$, is the pseudo-control signal, and $\Delta_q\left(x, \delta_e\right) = f_q\left(x, \delta_e\right) - \nu_q$ is the modeling error. Then ν_q can be solved for the actuator command input $\delta_{e_{com}}$ as

$$\delta_{e_{com}} = \frac{1}{\hat{M}_\delta}\left(\nu_q - \hat{M}_\alpha \alpha - \hat{M}_q q\right) \tag{51}$$

The pseudo-control ν_q is constructed as

$$\nu_q = \nu_{crm_q} + K_{p_q}\left(q_c - q\right) - \nu_{ad\,q} \tag{52}$$

$$\nu_{crm_q} = \frac{1}{\tau_q}\left(q_{com} - q_c\right) \tag{53}$$

where q_c is the reference pitch-rate command, obtained by filtering the raw pitch-rate command q_{com} through a first-order reference model with time constant τ_q, ν_{crm_q} is a feed-forward signal from the reference model, $K_{p_q} > 0$ is a proportional control gain, and ν_{ad_q} is the output of a SHL NN. The reader is referred to [14] for further details.

The reference model is hedged to prevent the adaptive law from adapting to the nonlinear actuator characteristics.

$$\begin{aligned} \nu_{h_q} &= \nu_q - \hat{\nu}_q = \left[\hat{M}_\alpha \alpha + \hat{M}_q q + \hat{M}_\delta \delta e_{com}\right] - \left[\hat{M}_\alpha \alpha + \hat{M}_q q + \hat{M}_\delta \hat{\delta}_e\right] = \hat{M}_\delta\left(\delta_{e_{com}} - \hat{\delta}_e\right) \\ \dot{q}_c &= \nu_{crm_q} - \nu_{h_q} \end{aligned} \tag{54}$$

where $\hat{\delta}_e$ is an estimate of the actuator deflection obtained by means of an actuator model. Figure 7 shows a block diagram representation of the adaptive pitch-rate control system.

The assumption of perfect inner-loop tracking of q_c enables the analysis and design of the PI controller in the outer-loop. Consider the following short-period approximation of the longitudinal dynamics with normal acceleration at the c.g. as the measured output,

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} Z_\alpha / u_0 & 1 \\ M_\alpha & M_q \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} Z_\alpha / u_0 \\ M_\delta \end{bmatrix} \delta_e \tag{55}$$

$$f_z = Z_\alpha \alpha + Z_q q \tag{56}$$

The transfer function from δ_e to q is given by

$$G_{q,\delta_e}(s) = \frac{M_\delta s - \left(Z_\alpha M_\delta - M_\alpha Z_\delta\right)/u_0}{s^2 - \left(Z_\alpha / u_0 + M_q\right)s + \dfrac{Z_\alpha M_q}{u_0} - M_\alpha} \tag{57}$$

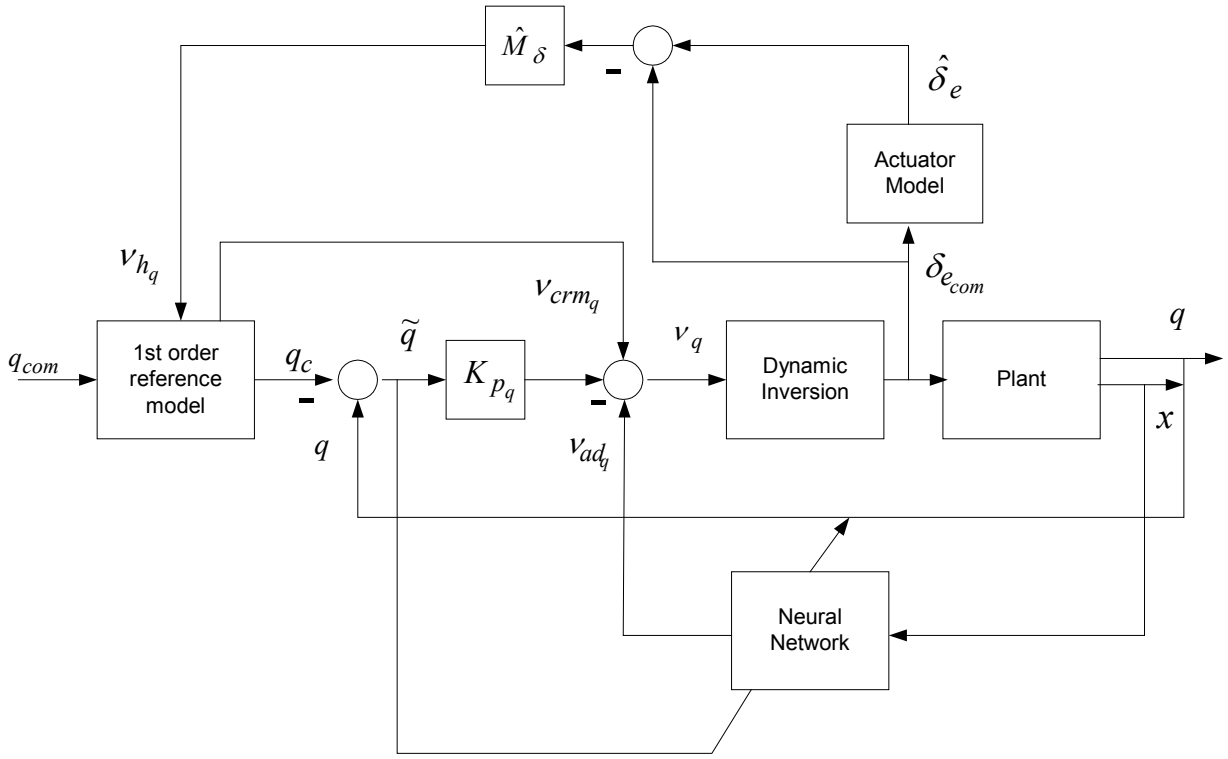

Figure 7. Block diagram representation of Adaptive pitch-rate control system

and the transfer function from δ_e to f_z is given by

$$G_{f_z,\delta_e}(s) = \frac{Z_\delta s^2 - Z_\delta M_q s + (Z_\alpha M_\delta - M_\alpha Z_\delta)}{s^2 - \left(Z_\alpha / u_0 + M_q\right)s + \frac{Z_\alpha M_q}{u_0} - M_\alpha} \tag{58}$$

Making the assumption $|Z_\delta| << |Z_\alpha M_\delta|$, the transfer function from q to f_z is given by

$$G_{f_z,q}(s) = \frac{G_{f_z,\delta_e}(s)}{G_{q,\delta_e}(s)} = \frac{Z_\alpha}{s - Z_\alpha / u_0} \tag{59}$$

Assuming perfect inner-loop tracking, i.e, $q_c = q$, allows us to represent the transfer function from q_{com} to q as

$$\frac{q(s)}{q_{com}(s)} = \frac{q_c(s)}{q_{com}(s)} = \frac{1}{\tau_q s + 1} \tag{60}$$

The block diagram of figure 6 can be idealized to the block diagram of figure 8. The analysis of the transfer function in figure 8 gives the loop transfer function

$$G(s) = \frac{Z_\alpha \left(K_{p_z} s + K_{I_z}\right)}{s(\tau_q s + 1)(s - Z_\alpha / u_0)} \tag{61}$$

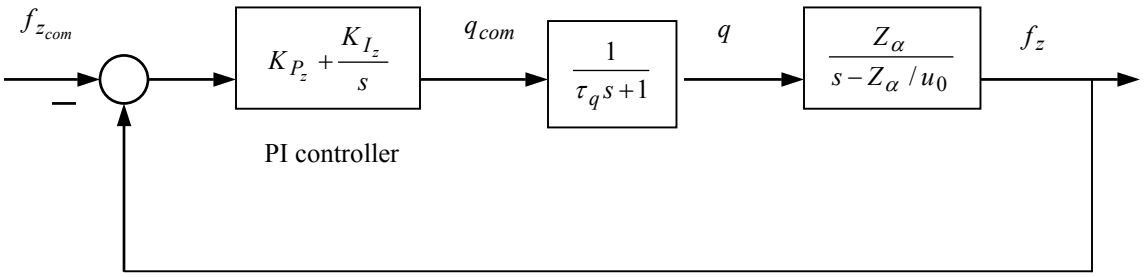

Figure 8. Idealized block diagram representation of Normal Acceleration Controller

The choice $\frac{K_{I_z}}{K_{p_z}} = -\frac{Z_\alpha}{u_0}$ results in a pole-zero cancellation, which leads to the following closed-loop transfer function

$$\frac{f_z(s)}{f_{z_{cmd}}(s)} = \frac{K_{p_z} Z_\alpha / \tau_q}{s^2 + s/\tau_q + K_{p_z} Z_\alpha / \tau_q} \tag{62}$$

Note that since $Z_\alpha < 0$ and $\left|\frac{Z_\alpha}{u_0}\right| >> 1$, the pole-zero cancellation in eq. (61) does not cause problems despite parametric uncertainty in Z_α. The PI controller gains can be obtained by choosing a desired natural frequency ω_n and damping ratio ς for the closed-loop transfer function in eq. (62).

$$\tau_q = \frac{1}{2\zeta\omega_n},\ K_{p_z} = \frac{\omega_n}{2\zeta Z_\alpha},\ K_{I_z} = -\frac{\omega_n}{2\zeta u_0} \tag{63}$$

B. Adaptive Lateral Acceleration (f_y) Control

Lateral acceleration control is almost identical in structure to normal acceleration control. The difference from the normal acceleration controller lies in regulating a blended output variable $y_{lat} = \beta + C_r r$ in the inverting inner-loop controller instead of just the yaw-rate r. The reason lies in the fact that the transfer function from $\delta_r(s)$ to $r(s)$ has a zero close to the origin, the effect of which is to produce a very slow mode when the dynamic inversion is inexact. By redefining the output to be controlled in the inverting inner-loop as y_{lat}, the zero of the associated transfer-function can be placed at a desirable location. The reader is referred to [14] for details.

C. Adaptive Bank Angle (ϕ) Control

Compared to f_z and f_y control, adaptive bank angle control is straightforward. This is because the transfer function from the aileron δ_a to the bank angle ϕ is minimum phase. The true bank angle dynamics can be given as

$$\ddot{\phi} = A_{roll} x_{lat} + B_{roll} \delta_a + \Delta_{roll}(x_{lat}, \delta_a) \tag{64}$$

where $x_{lat} = [v, p, r, \phi]^T$ represents the state vector for the lateral-directional dynamics, A_{roll} and B_{roll} are row vectors that come from the linear model obtained at the trim flight condition, and $\Delta_{roll}(x, \delta_a)$ is the modeling error. Inversion-based control is designed as

$$\delta_{a\,com} = \frac{1}{B_{roll}}\left(\nu_{roll} - A_{roll}\, x_{lat}\right) \tag{65}$$

where the pseudo-control ν_{roll} is given as

$$\nu_{roll} = \nu_{crm_\phi} + K_{p_\phi}\left(\phi_c - \phi\right) + K_{d_\phi}\left(\dot{\phi}_c - \dot{\phi}\right) - \nu_{ad\,\phi} \tag{66}$$

where $\phi_c, \dot{\phi}_c$ and ν_{crm_ϕ} refer to the outputs of a second-order, linear reference model. The reference model is hedged to prevent the adaptive law from adapting to the nonlinear characteristics of the actuator.

$$\nu_{crm_\phi} = \omega_{n_\phi}^2 \phi_{com} - \omega_{n_\phi}^2 \phi_c - 2\zeta_\phi \omega_{n_\phi} \dot{\phi}_c \tag{67}$$

$$\begin{aligned} \nu_{h_\phi} &= \nu_{roll} - \hat{\nu}_{roll} = \left[A_{roll}\, x_{lat} + B_{roll}\, \delta_{a_{com}}\right] - \left[A_{roll}\, x_{lat} + B_{roll}\, \hat{\delta}_a\right] = B_{roll}\left(\delta_{a_{com}} - \hat{\delta}_a\right) \\ \ddot{\phi}_c &= \nu_{crm_\phi} - \nu_{h_\phi} \end{aligned} \tag{68}$$

where $\hat{\delta}_a$ is an estimate of the aileron deflection.

D. Throttle Control

The throttle controller is a PI controller with anti-windup feature for commanding throttle position $\delta_{T_{com}}$. There is a stability issue when we use a pure acceleration command, $e_{x_{cmd}} = f_{x_{cmd}} - f_x$, as input to the throttle controller. This stability issue is particular to our aircraft model and occurs when sharp heading turns are commanded. This is due to the fact that the guidance logic generates an excessive negative acceleration command along the X-axis of the body frame when starting a heading turn and this causes saturation into the lower bound of the throttle, ultimately leading to instability of the entire closed-loop system. So, we modified the command to the throttle controller as,

$$e_{Lx_{cmd}} = V_{com} - V_L \text{, for the leader aircraft, and} \tag{69}$$

$$e_{Fx_{cmd}} = V_{com} - V_F + K_x f_{x_{cmd}},\ K_x > 0 \text{ for the follower aircraft} \tag{70}$$

where V_{com} is the commanded speed of the leader, and V_L, V_F refer to the leader and follower speeds. There is a trade-off with the range tracking performance as a result of the command modification to the throttle controller. For the follower aircraft, using $e_{Fx_{cmd}}$ as throttle controller command, reduces the transient speed of response of the range variable and the desired steady-state with respect to range is not exactly achieved, even when the leader stops maneuvering. The gain K_x has to be chosen carefully, to get acceptable range tracking performance while not saturating the throttle controller. In case there is no danger of saturating the throttle by commanding negative acceleration along the body X-axis, for e.g., if spoilers can be deployed for additional drag, it should be possible to use $e_{x_{cmd}}$ as input to the throttle controller.

V. Nonlinear Simulation Model

The aircraft to be employed for the formation flight experiment is an unmanned, fixed-wing aircraft called the Sig Rascal 110. Figure 9 shows a picture of this aircraft.

Figure 9. Sig Rascal 110

Data pertaining to this aircraft was obtained by applying methods of parameter identification to data obtained from a flight test. A nonlinear 6-DOF simulation with linearized aerodynamics is used for the testing of the control and guidance algorithms for formation flight. A minimum number of sensors were assumed in developing the simulation and a nominal autopilot. Quaternion attitude angles are obtained by integrating the rate gyros. The simulation model is a rigid body aircraft model with 13 states, 3 for position with respect to the Earth-fixed frame, 3 for translational velocity expressed in the body frame, 4 for the quaternions and 3 for the angular velocity expressed in the body frame. Engine thrust is obtained from a linear interpolation map of throttle position. The actuators are modeled as first-order, stable linear filters with rate and position limits and time delays.

$$\dot{\delta}(t) = \text{sat}\left(\frac{1}{\tau_a}\left[\delta_{cmd}(t-\tau_d) - \text{sat}(\delta(t))\right] \right) \tag{71}$$

where $\delta_{cmd}(t)$ is the actuator command signal at time t, $\delta(t)$ is the actuator output at time t, $\text{sat}(\cdot)$ is a linear saturation operator, τ_a is the actuator time constant, and τ_d is the time delay.

VI. Simulation Results

We demonstrate closed-loop formation flight results for three sets of leader aircraft maneuvers. The leader maneuvers by tracking waypoints in the inertial reference frame. Three sets of waypoints are prescribed: 1) waypoints in the inertial X-Z plane, requiring the leader to climb and descend, 2) waypoints at the corners of a square box in the inertial X-Y plane, requiring the leader to make sharp heading changes at the corners of the box while holding altitude, and 3) waypoints prescribed at the corners of a slanted box in 3D space, requiring sharp heading changes, climbing and descending motions of the leader. The three sets of waypoints require increasing levels of maneuvers from the leader. The objective is to investigate the range tracking performance of the follower aircraft in presence of leader maneuvers.

The range command is set to $R_{com} = 5$ meters, which is slightly less than 2 wing-span lengths. The wing-span length of the simulated aircraft is $b = 2.8$ meters. The commanded speed of the leader aircraft is $V_{com} = 25$ meters/sec. The initial conditions for the leader and follower is the trim condition of steady, level flight. The leader initial position is at (0,0,0) meters and the follower initial position is at (-10,0,5) meters. The gain K_x in eq. (70) is optimized for the 3 maneuvers and set to 0.1. The simulation is terminated when the leader aircraft approaches within $d_{\min} = 20$ meters (eq. 5) of the last waypoint. The plots shown are used to demonstrate the tracking performance of the guidance and control law over the range of maneuvers.

Case 1: Waypoints in inertial X-Z plane

Figure 10 shows the 3 D trajectory of the leader and follower. The red circle and blue triangle at the bottom right of the figure are the initial positions of the leader and follower respectively. Note that the Z-axis scale is magnified compared to the X axis. Figure 11 shows the range tracking performance. Note that the maximum overshoot from the commanded range is less than 1 meter in the presence of leader maneuvers. The performance is very much acceptable.

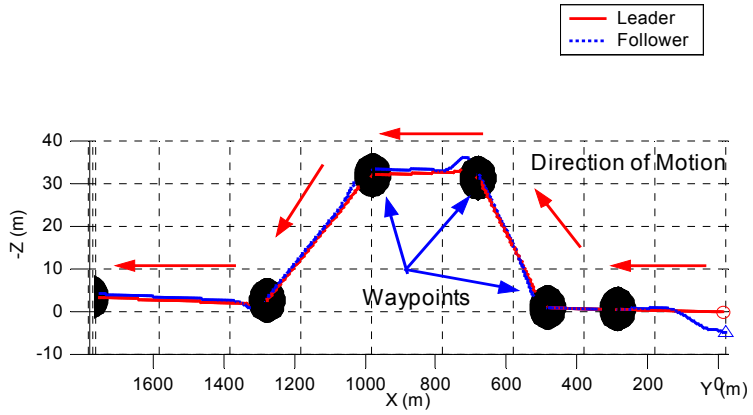

Figure 10. 3D Leader and Follower Trajectory, in meters (Case 1)

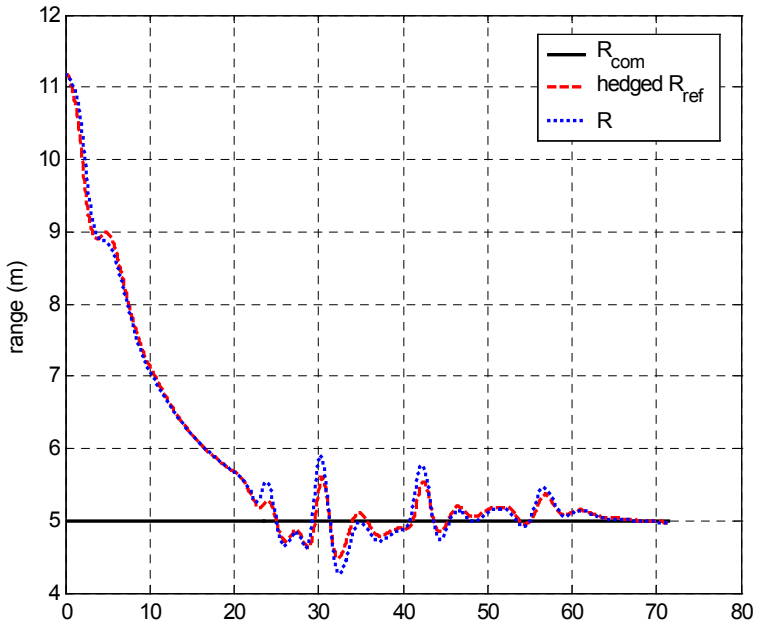

Figure 11. Range Tracking Performance, in meters (Case 1)

Figure 12 shows the specific force tracking performance for both leader and follower aircrafts. Notice that the tracking performance of $f_{x_{com}}$ for the leader and follower aircraft (top subplots) is not good as that in the z-axis. The reason is the choice of command to the throttle controller (eq. 69 –70). Figure 13 shows the Euler attitude angles of the leader and follower aircraft. Figure 14 shows the actuator histories for the leader and follower.

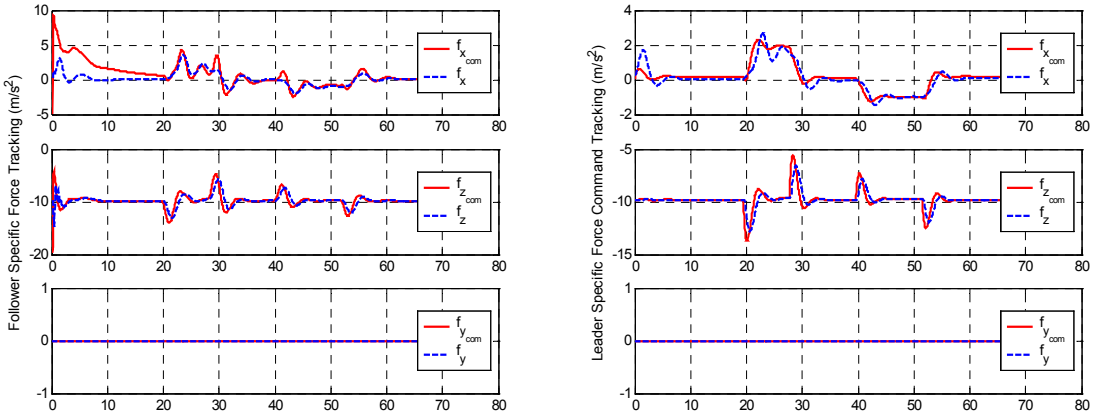

Figure 12. Specific Force Tracking (left = Follower, right = Leader), in m/s^2 (Case 1)

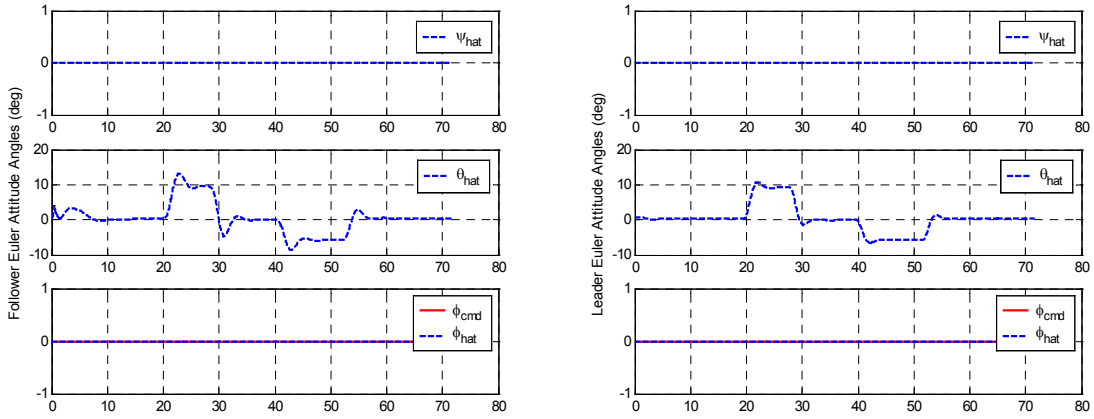

Figure 13. Euler Attitude Angles (left = Follower, right = Leader), in deg (Case 1)

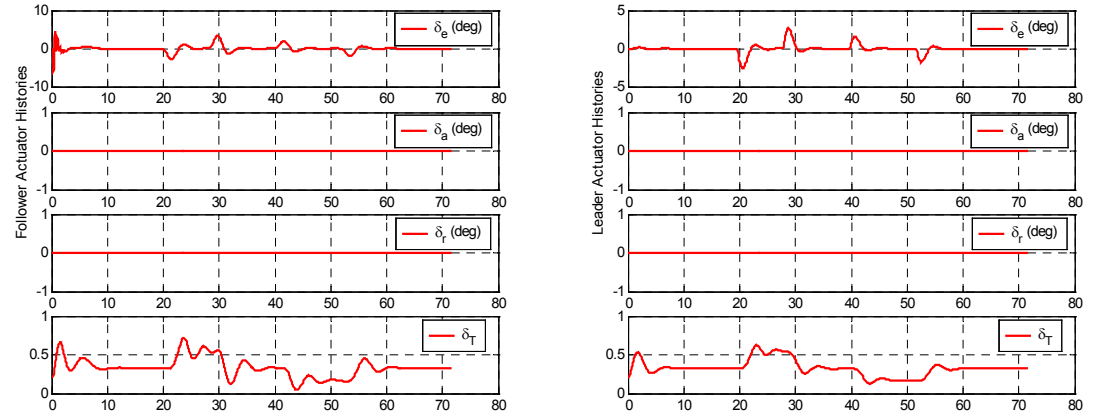

Figure 14. Actuator Histories (left = Follower, right = Leader) (Case 1)

Case 2: Waypoints at the corner of square box in inertial X-Y plane

Figure 15 shows the 3D trajectory of the leader and follower from the top view. Figure 16 shows the range tracking history. Note that the maximum overshoot from the commanded range is less than 2 meters in the presence of leader maneuvers. This implies a slight deterioration compared to the tracking performance in figure 11. The overshoots occur when the leader aircraft commands a change in heading after passing through a waypoint. Overall, the performance is still acceptable.

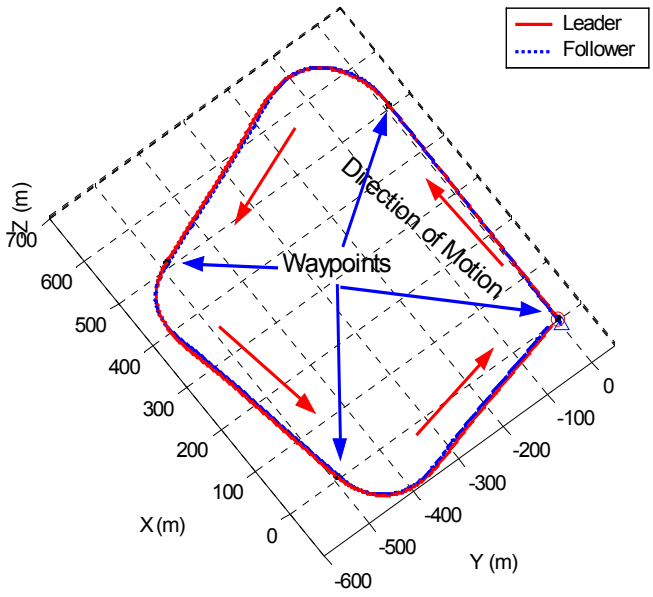

Figure 15. 3D Leader and Follower Trajectory, in meters (Case 2)

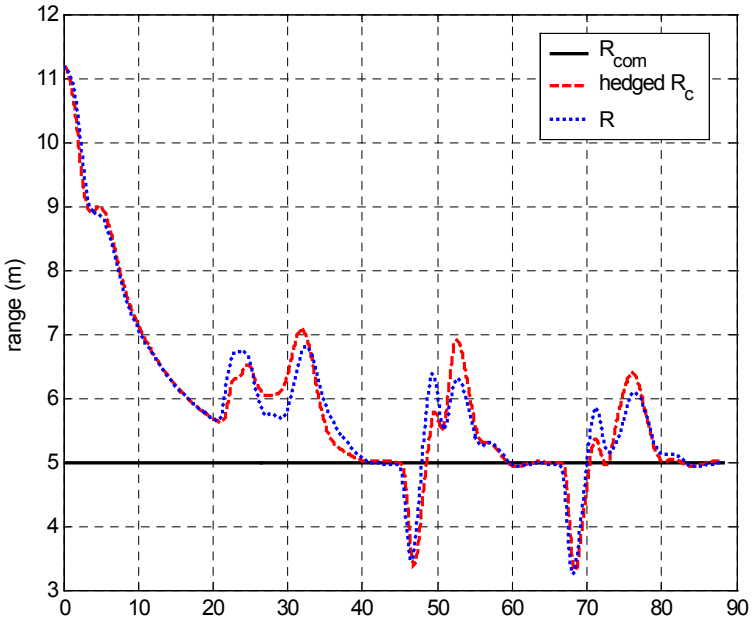

Figure 16. Range Tracking Performance, in meters (Case 2)

Figure 17 shows the specific force tracking performance for both leader and follower aircrafts. The tracking is fairly acceptable for the normal acceleration and lateral acceleration commands, for both the leader and follower. Figure 18 shows the Euler attitude angles histories. The bottom subplots show the bank angle command tracking for both the leader and follower. The tracking is very good. Figure 19 shows the actuator histories for the leader and follower.

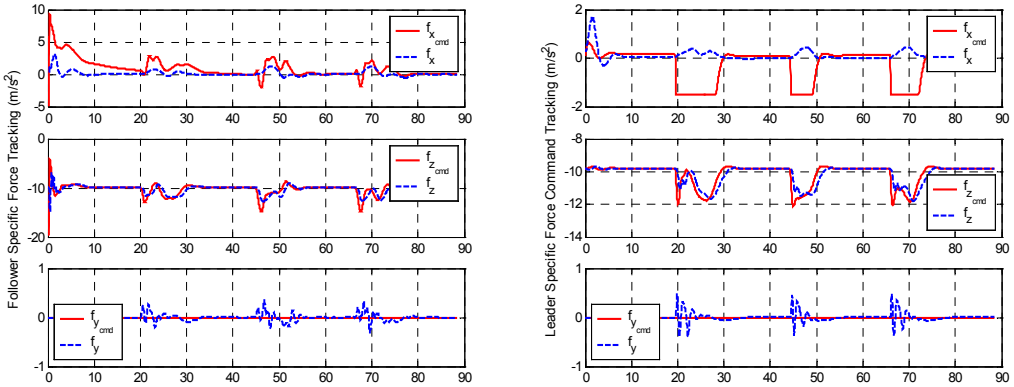

Figure 17. Specific Force Tracking (left = Follower, right = Leader), in m/s^2 (Case 2)

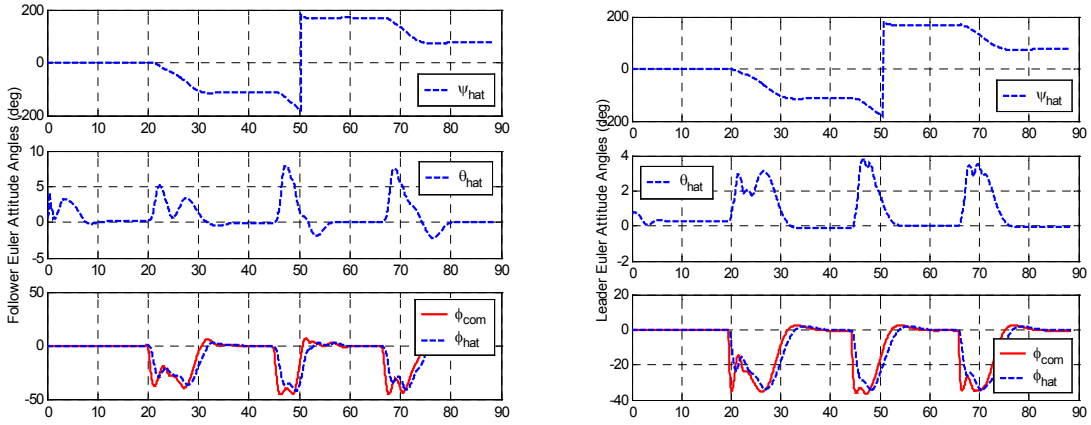

Figure 18. Euler Attitude Angles (left = Follower, right = Leader), in deg (Case 2)

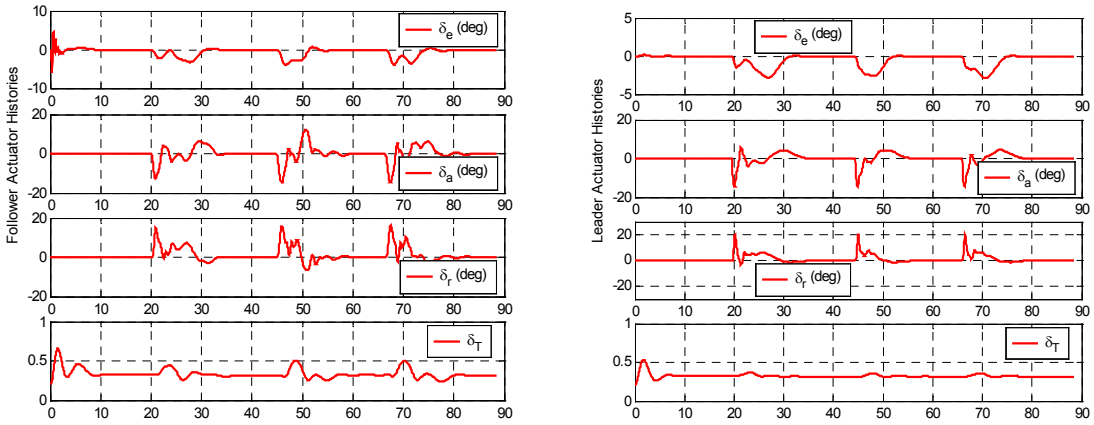

Figure 19. Actuator Histories (left = Follower, right = Leader) (Case 2)

Case 3: Waypoints at the corner of slanted box in 3D inertial space

Figure 20 shows the 3D trajectory of leader and follower for the last maneuver. This maneuver differs from the preceding one in that after passing through the first waypoint the leader turns and climbs to 50 meters, does another turn while holding altitude, and finally returns to the starting point by turning and descending. The length of the box is also smaller indicating that the turns in this case are more severe than in case 2.

Figure 21 shows the range tracking performance. The plot shows the tracking is degraded compared to the tracking in case 1 (figure 11) and case 2 (figure 16). The maximum overshoot has been increased to almost 3 meters, showing the effect of the leader maneuvers on the tracking performance. Figure 22 shows the specific force command tracking histories. Figure 23 shows the Euler attitude angles histories. Figure 24 shows the actuator histories. The plots show acceptable tracking performance and reasonable actuator behavior.

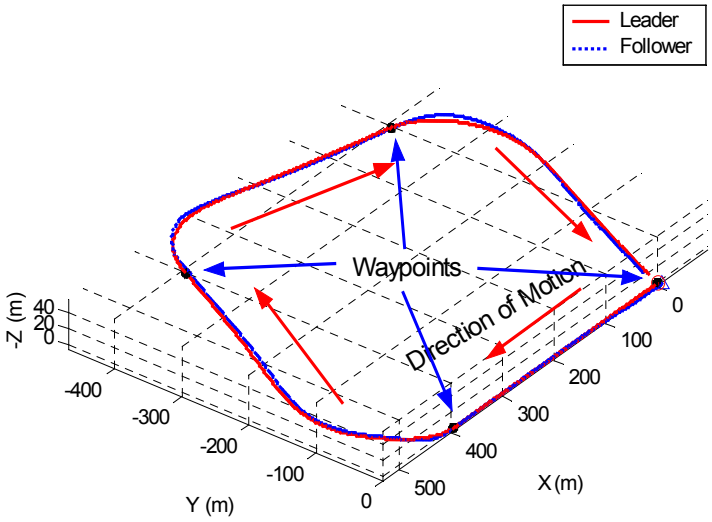

Figure 20. 3D Leader and Follower Trajectory, in meters (Case 3)

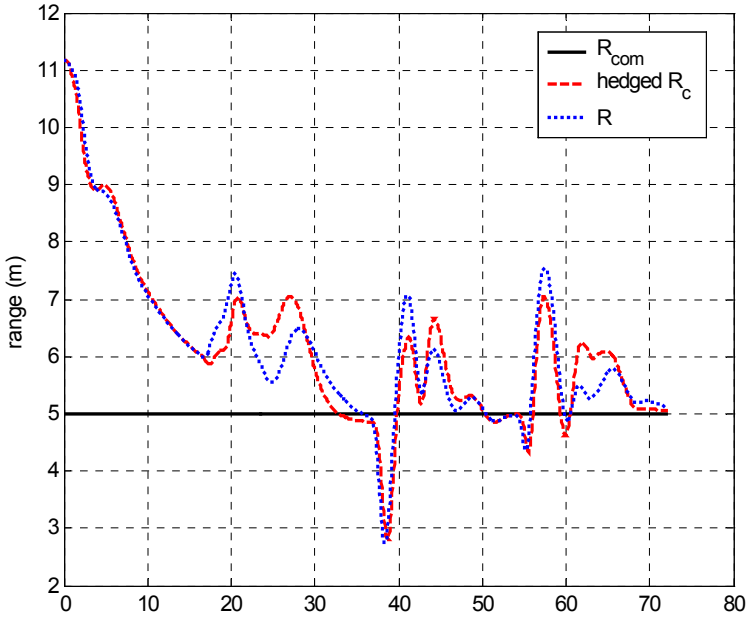

Figure 21. Range Tracking Performance, in meters (Case 3)

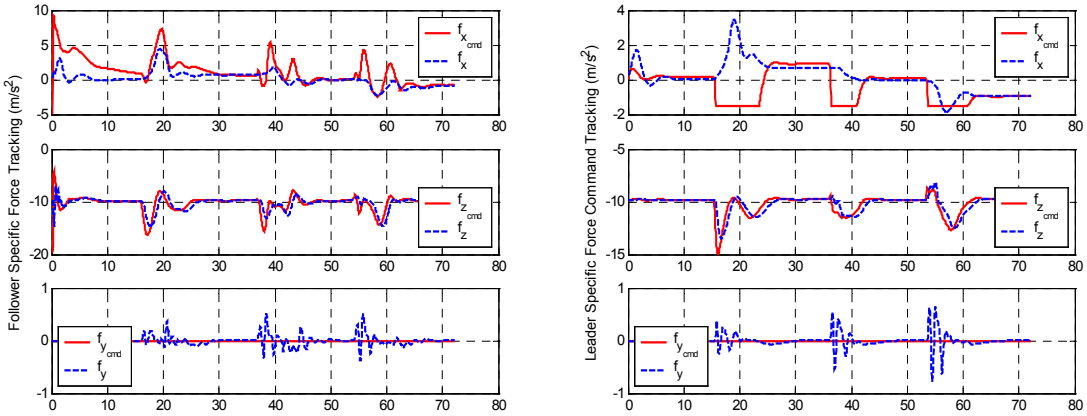

Figure 22. Specific Force Tracking (left = Follower, right = Leader), in m/s^2 (Case 3)

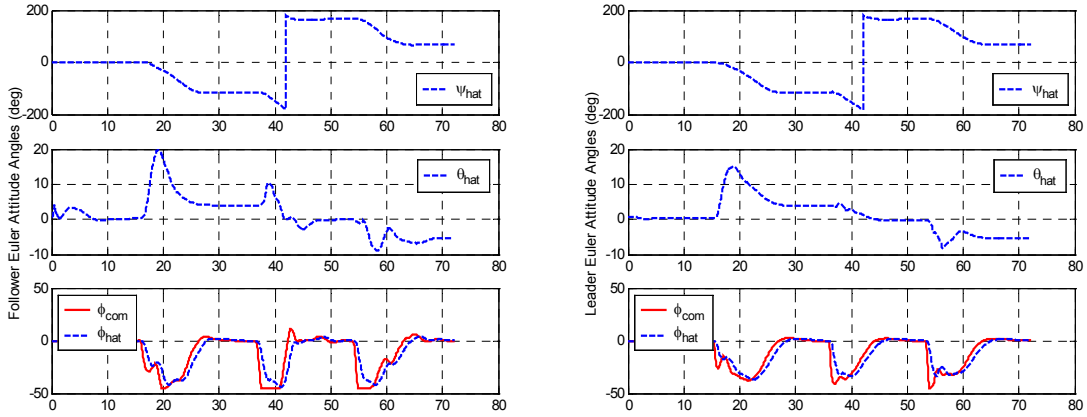

Figure 23. Euler Attitude Angles (left = Follower, right = Leader), in deg (Case 3)

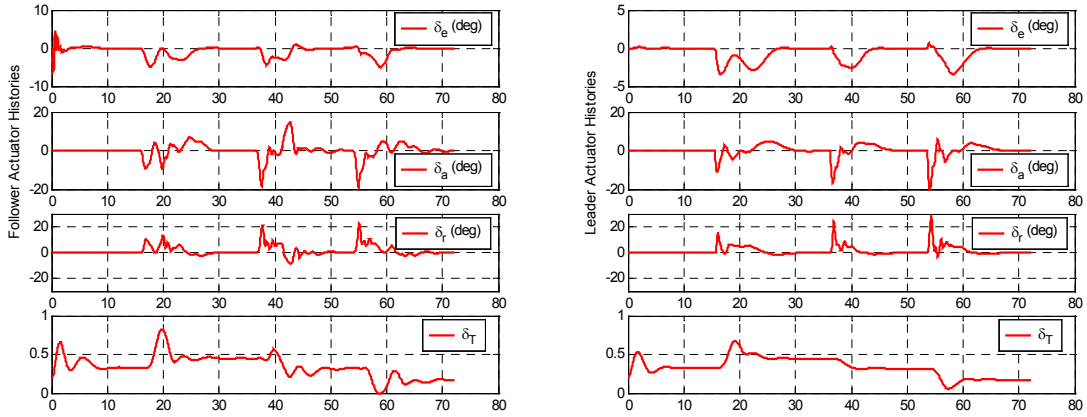

Figure 24. Actuator Histories (left = Follower, right = Leader) (Case 3)

VII. Conclusions

The paper presents an adaptive guidance and control law algorithm for implementation on a pair of Unmanned Aerial Vehicles (UAVs) in a 6 DOF leader-follower formation flight simulation. The follower aircraft guidance law is designed on the assumption that the follower aircraft is equipped with an onboard camera for passive sensing of the relative distance and orientation to the leader aircraft. One conclusion of this paper is that the adaptive velocity based guidance law developed on the assumption of point-mass aircraft models and first-order autopilot modeling[1], always results in a tail-chase situation for the follower aircraft and is inappropriate for application in a 6 DOF setting. With this velocity based guidance law, the range tracking performance is highly degraded in the presence of leader aircraft maneuvers. The adaptive acceleration based guidance law presented in this paper makes the range tracking more robust to leader maneuvers, and the adaptive autopilot design reduces the dependence of the overall controller design on the aircraft aerodynamic data. Future research will be directed towards integrating the guidance and controller design with the vision components of the flight control system, and flight testing.

Acknowledgments

This research has been sponsored under AFOSR contract F4960-01-1-0024 and under NRTC contract NCC 2-945. The authors would particularly like to thank Kaiser Michael Kent of Eglin Air Force Base for providing the nonlinear 6-DOF simulation model, nominal autopilot design and for various technical interactions.

References

[1] R. Sattigeri, A. J. Calise, and J. H. Evers, "An adaptive vision-based approach to decentralized formation control," *AIAA Guidance, Navigation, and Control Conference*, Providence, RI, August 2004.

[2] E. Lavretsky, "F/A –18 Autonomous Formation Flight Control System Design," *AIAA Guidance, Navigation, and Control Conference*, Monterrey, California, August 2002.

[3] R. Ray, B. Cobleigh, M. Vachon and C. St. John, "Flight Test Techniques used to Evaluate Performance Benefits During Formation Flight," *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Monterrey, California, August 2002.

[4] B. Senor, G. Campa, Y. Gu, M. Napolitano, L. Rowe and M. Perhinschi, "Formation Flight Test Results for UAV Research Aircraft Models," *AIAA 1st Intelligent Systems Technical Conference*, Chicago, Illinois, September 2004.

[5] L. Polloni, R. Mati, M. Innocenti, G. Campa and M. Napolitano, "A Synthetic Environment for the Simulation of Vision-based Formation Flight, " *AIAA Modeling and Simulation Technologies Conference*, Austin, Texas, August 2003.

[6] A. Betser, P. Vela, and A. Tannenbaum, "Automatic Tracking of Flying Vehicles Using Geodesic snakes and Kalman filtering," *IEEE Conference on Decision and Control*, Vol. 2, pp 1649-1654, December 2004.

[7] J. Ha, C. Alvino, G. Pryor, M. Niethammer, E. Johnson, and A. Tannenbaum, "Active Contours and Optical Flow for Automatic Tracking of Flying Vehicles," *American Control Conference*, Vol. 4, pp 3441-3446, 2004.

[8] E. Johnson, A. Calise, R. Sattigeri, Y. Watanabe, and V. Madyastha, "Approaches to Vision-based Formation Control," *IEEE Conference on Decision and Control*, Vol. 2, pp 1643-1648, December 2004.

[9] V. Madyastha, and A.J. Calise, "An Adaptive Filtering Approach to Target Tracking," Accepted for publication in *American Control Conference*, 2005.

[10] N. Hovakimyan, A.J. Calise and N. Kim, "Adaptive Output Feedback Control of a Class of Multi-Input Multi-Output Systems using Neural Networks," *International Journal of Control*, October 2004, vol. 77, no. 15, pp 1318-1329.

[11] N. Kim, "Improved Methods in Neural Network based Adaptive Output Feedback Control, with applications to Flight Control," Ph.D. dissertation, School of Aerospace Engineering, Georgia Institute of Technology, GA, November 2003.

[12] 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.

[13] G.F. Franklin, J.D. Powell, and A. Emami-Naeini, "*Feedback Control of Dynamic Systems*", Addison-Wesley Publishing Compnay, 1994, Third Edition, pp. 196-198.

[14] A.J. Calise, M. Sharma and E. J. Corban, "Adaptive Autopilot Design for Guided Munitions," *Journal of Guidance, Control and Dynamics*, Vol. 23, No. 5, Sept-Oct 2000.

[15] J. Pomet and L. Praly, "Adaptive Nonlinear Regulation: Estimation from the Lyapunov equation", *IEEE Transactions on Automatic Control*, Vol. 37, No. 6, 1992, pp. 729-740.

[16] A.J. Calise, S.Lee, and M. Sharma, "Development of a Reconfigurable Flight Control Law for Tailless Aircraft," *Journal of Guidance, Control and Dynamics*, Vol. 24, No. 5, pp. 896-902, 2001.

[17] M. B. McFarland and A.J. Calise, "Multilayer Neural Networks and Adaptive Nonlinear Control of Agile Anti-Air Missiles," *AIAA Guidance, Navigation, and Control Conference*, 97-3540, August 1997.